PERFORMANCE METRICS FOR INTELLIGENT SYSTEMS WORKSHOP

Courtyard Gaithersburg Washingtonian Center, Gaithersburg, Maryland USA
August 28 - 30, 2007

Table of Contents

Technical Sessions

TUE-AM1 Mobile Robot Performance Evaluation I

TUE-AM2 Special Session I: Autonomy Levels for Unmanned Systems

TUE-PM1 Mobile Robot Performance Evaluation II

TUE-PM2 Special Session II: Human Robot Interface Issues

WED-AM2 Autonomy Vs Intelligence

WED-AM2 Panel Discussion I

WED-PM1 Human Machine Interaction

WED-PM2 Special Session III: Space/Aerial Robotics

THU-AM1 Performance Assessment of Algorithms

THU-AM2 Special Session IV: Smart Assembly Systems

THU-PM Panel Discussion II

FOREWORD

The 2007 Performance Metrics for Intelligent Systems (PerMIS) Workshop was held at the Courtyard Gaithersburg Washingtonian Center from August 28–30. This seventh installment of PerMIS started in 2000 targeted at defining measures and methodologies of evaluating performance of intelligent systems, and focused on applications of performance measures to practical problems in commercial, industrial, homeland security, and military applications. The cardinal theme of this year's workshop was *the interplay between autonomy and intelligence*, i.e. how does autonomy influence intelligence and vice versa.

Defining and measuring aspects of a system:

- The level of autonomy
- Human-robot interaction
- Collaboration

Evaluating components within intelligent system

- Sensing and perception
- Knowledge representation, world models, ontologies
- Planning and control
- Learning and adaption
- Reasoning

Infrastructural support for performance evaluation

- Testbeds and competitions for intercomparisons
- Instrumentation and other measurement tools
- Simulation and modeling support

Technology readiness measures for intelligent systems

Applied performance measures in various domains, e.g.,

- Intelligent transportation systems
- Emergency response robots (search and rescue, bomb disposal)
- Homeland security systems
- De-mining robots
- Defense robotics
- Hazardous environments (e.g., nuclear remediation)
- Industrial and manufacturing systems
- Space/Aerial robotics
- Medical Robotics & Assistive devices

This year's exciting program consisted of four plenary addresses, one featured presentation, four special sessions, and two panel discussions. In addition to these, there were five general technical sessions. All of these presentations addressed, in one way or another, performance metrics, evaluation, and analysis of intelligent systems in diverse domains ranging from space robotics to manufacturing, from mobile robotic systems to human machine interaction, to name a few.

PerMIS’07 was sponsored by NIST with technical co-sponsorship of the IEEE Washington Section Robotics and Automation Society Chapter and in-cooperation with the Association for Computing Machinery (ACM) Special Interest Group on Artificial Intelligence (SIGART). We also acknowledge the financial support of the IEEE Washington Section.

There were several firsts to this year’s workshop. The proceedings of PerMIS’07 are being indexed by INSPEC, Ei Compendex, ACM’s Digital Library, and are released, as in previous years, as a NIST Special Publication. These indexing services will enable the presented work to reach a wider audience for increased references and citations. Springer Publishers hosted a booth on the last two days of the workshop during which time some of the displayed books were raffled off. We thank Springer for their participation and hope that this is the beginning of many years of their support.

We would like to thank all members of the PerMIS′07 Program Committee, and the reviewers for contributing to the success of the workshop. Most importantly, we thank all authors for their valuable submissions and the attendees for their participation. We sincerely hope that you enjoyed the presentations and ensuing discussions, while forging new relationships and renewing old ones. It was our great pleasure to host all the attendees.

See you next year!

Raj Madhavan	Elena Messina
Program Chair	General Chair

SPONSORS

NIST
National Institute of Standards and Technology

IEEE

acm

PROGRAM COMMITTEE

General Chair:

Elena Messina (Intelligent Systems Division, NIST, USA)

Program Chair:

Raj Madhavan (Oak Ridge National Laboratory/NIST, USA)

R. Bonneau (AFRL USA)

S. Balakirsky (NIST USA)

G. Berg-Cross (EM & I USA)

J. Bornstein (Army Res. Lab. USA)

S. Carpin (UC Merced USA)

J. Evans (USA)

D. Gage (XPM Tech. USA)

J. Gunderson (Gamma Two, Inc. USA)

L. Gunderson (Gamma Two, Inc. USA)

A. Jacoff (NIST USA)

S. Julier (Univ. College London UK)

T. Kalmar-Nagy (Texas A&M USA)

R. Lakaemper (Temple Univ. USA)

L. Latecki (Temple Univ. USA)

M. Lewis (Univ. of Pittsburgh USA)

A. del Pobil (Univ. Jaume-I, Spain)

L. Reeker (NIST USA)

C. Schlenoff (NIST USA)

A. Schultz (Navy Res. Lab. USA)

M. Shneier (NIST USA)

R. Smith (OSD USA)

R. Tilove (General Motors USA)

E. Tunstel (Jet Propulsion Lab. USA)

PLENARY SPEAKER

Prof. Maria Gini

University of Minnesota, USA

Methodology for Experimental Research in Multi-robot Systems with Case Studies

Tue. 08:30

ABSTRACT

Fully repeatable and controllable experiments are essential to enable a precise comparison of multi-robot systems. Using different case studies, we describe a general methodology for conducting experimental activities for multi-robot systems. This is a first step toward the goal of fostering the practice of replicating experiments in order to compare different methods and assess their strengths and weaknesses.

In the first case study, we examine the problem of building a geometrical map of an indoor environment using multiple robots. The map is built by integrating partial maps made of segments without using any odometry information. We show how to improve the repeatability and controllability of the experimental results and how to compare different mapping systems.

We then present a case study of auction-based methods for the allocation of tasks to a group of robots. The robots operate in a 2D environment for which they each have a map. Tasks are locations in the map that must be visited by one robot. Robots bid to obtain tasks, but unexpected obstacles and other delays may prevent a robot from completing its allocated tasks. We show how to compare our experimental results with other published auction-based methods.

BIOGRAPHY

Maria Gini is a Professor at the Department of Computer Science and Engineering of the University of Minnesota. Before joining the University of Minnesota, she was a Research Associate at the Politecnico of Milan, Italy, and a Visiting Research Associate at Stanford University. Her work has included motion planning for robot arms, navigation of mobile robots around moving obstacles, unsupervised learning of complex behaviors, coordinated behaviors among multiple robots, and autonomous economic agents. She has coauthored over 200 technical papers. She is currently the chair of ACM Special Interest Group on Artificial Intelligence (SIGART), a member of the Association for the Advancement of Artificial Intelligence (AAAI) Executive Council and of the board of the International Foundation of Autonomous Agents and Multi-Agent Systems. She is on the editorial board of numerous journals, including Autonomous Robots, the Journal of Autonomous Agents & Multi-Agent Systems, Electronic Commerce Research and Applications, Integrated Computer-Aided Engineering, and Web Intelligence and Agent Systems.

PLENARY SPEAKER

Dr. Eric Krotkov

Griffin Technologies, USA

Measuring Ground Robot Performance

Tue. 14:00

ABSTRACT

This talk first describes several approaches to measure the performance of ground robots. It is easy enough to measure quantities such as speed and reliability. It is more challenging to define metrics for perception, planning, and autonomy. The talk then presents selected results of applying the approaches to systems developed by several Government programs.

BIOGRAPHY

Dr. Krotkov is the President of Griffin Technologies, a consulting and software firm specializing in robotics and machine perception. Before founding Griffin, he worked in industry as an executive in a medical imaging technology start-up, in government as a program manager at DARPA, and in academia as a faculty member of the Robotics Institute at Carnegie Mellon University. Dr. Krotkov earned his Ph.D. degree in Computer and Information Science in 1987 from the University of Pennsylvania, for pioneering work in active computer vision.

PLENARY SPEAKER

Prof. Illah Nourbakhsh

Carnegie Mellon University, USA

Formalizing Educational Human-Robot Collaboration

Wed. 08:30

ABSTRACT

Designing human-robot collaboration systems is an inherently multidisciplinary endeavor aimed at providing humans with rich, effective and satisfying interactions. Over the past ten years, my laboratory has focused on educational collaboration, wherein the purpose of the interaction is to provide measurable learning for humans through exploration and discovery. We propose that the creation of a successful human-robot collaboration system requires innovation in several areas: robot morphology; robot behavior; social perception; interaction design; human cognitive models and evaluation of educational effectiveness. Our iterative process for collaboration design extends evaluation techniques from the informal learning field together with underlying technical advances in robotics. This talk describes our research methodology, technical contributions and experimental outcomes for three fielded robot systems that push on developing a generalizable, formal approach to educational human-robot collaboration. For the past several months, our group has been laying the groundwork for large-scale dissemination of our technology and curricular instruments.

I will describe the robot “community” we wish to help spawn, and the ingredients that may help to catalyze a broad form of technologically empowered community, including the Telepresence Robot Kit and the Global Connection Project.

BIOGRAPHY

Illah R. Nourbakhsh is an Associate Professor of Robotics and head of the Robotics Masters Program in The Robotics Institute at Carnegie Mellon University. He was on leave for the 2004 calendar year and was at NASA/Ames Research Center serving as Robotics Group lead. He received his Ph.D. in computer science from Stanford University in 1996. He is co-founder of the Toy Robots Initiative at The Robotics Institute, director of the Center for Innovative Robotics and director of the Community Robotics, Education and Technology Empowerment (CREATE) lab. He is also co-PI of the Global Connection Project, home of the Gigapan project. He is also co-PI of the Robot 250 city-wide art+robotics fusion program in Pittsburgh. His current research projects include educational and social robotics and community robotics. His past research has included protein structure prediction under the GENOME project, software reuse, interleaving planning and execution and planning and scheduling algorithms, as well as mobile robot navigation. At the Jet Propulsion Laboratory he was a member of the New Millenium Rapid Prototyping Team for the design of autonomous spacecraft. He is a founder and chief scientist of Blue Pumpkin Software, Inc., which was acquired by Witness Systems, Inc. Illah recently co-authored the MIT Press textbook, Introduction to Autonomous Mobile Robots.

PLENARY SPEAKER

Dr. Alex Zelinsky

CSIRO ICT
Centre,
Australia

Building Autonomous Systems of High Performance, Reliability and Integrity

Thu. 08:30

ABSTRACT

Commercial applications for the everyday deployment of autonomous systems based on robotic and intelligent systems technologies require the highest levels of performance, reliability and integrity. The general public expects intelligent machines to be fully operational 100% of the time. People expect autonomous technologies to operate at higher levels of performance and safety than people themselves exhibit. For example smart car technologies are expected to cause ZERO accidents while human errors kill more 150,000 people on our roads every year! This talk will describe the design principles that have been developed over of the last 10 years through exhaustive trial and error testing to underpin autonomous systems that are suitable for real-world deployment. Currently, it is not yet possible to realise an autonomous system that doesn't fail periodically. Even if the mean rate between failures is days or weeks, a single failure could have catastrophic consequences. The approach we have adopted to address this situation has been to build-in monitoring systems that continually check all key system parameters and variables. If the monitored parameters move outside tightly defined bounds the system will safely shutdown, and alert the human supervisor. The failure conditions are logged and then further testing and debugging is performed. The value and appropriateness of our approach will be shown by a number of real-world studies. We will show that how it is possible to design computer vision systems for human-machine applications can operate with over 99% reliability, in all lighting conditions, for all types of users irrespective of age, race or visual appearance. These systems have been used in automotive and sports applications. We have also show how this approach has been used to design field robotic systems that have deployed in automobile safety systems and 24/7 mining applications.

BIOGRAPHY

Dr. Alex Zelinsky is a well-known scientist, specialising in robotics and computer vision and is widely recognised as an innovator in human-machine interaction. Dr. Zelinsky is currently Group Executive, Information and Communication Sciences and Technology, and Director, CSIRO Information Communication Technology (ICT) Centre. Before joining CSIRO in July 2004, Dr. Zelinsky was CEO of Seeing Machines, a company dedicated to the commercialisation of computer vision systems. Dr. Zelinsky co-founded Seeing Machines in June 2000, the company is now publicly listed on the London Stock Exchange. The technology commercialised by Seeing Machines was developed at the Australian National University where Dr. Zelinsky was Professor and Head of the Department of Systems Engineering (1996 -2000). Prior to joining the Australian National University, Dr. Zelinsky worked as an academic at the University Wollongong (1984-1991) and as a research scientist in the Electrotechnical Laboratory, Japan (1992-1995). Dr. Zelinsky is an active member of the robotics community and has served on the editorial boards of the International Journal of Robotics Research and IEEE Robotics and Automation Magazine, he also founded the Field & Services Robotics conference series. Dr. Zelinsky's contributions have been recognised by awards in Australia and internationally. These include the Australian Engineering Excellence Awards, US R&D magazine Top 100 Award and Technology Pioneer at the World Economic Forum.

FEATURED PRESENTATION

Dr. Vladimir Lumelsky

NASA-Goddard Space Center, USA

Human-Robot Interaction in Physical Proximity: Issues and Prospects

Wed. 14:00

ABSTRACT

After spectacular successes, in 1970s-1980s, in the use of robotics in highly structured environments - e.g. automotive assembly, welding, and painting lines - the penetration of "serious" robots (those large and powerful enough to be harmful) into new applications has slowed down markedly. User manuals of most robot arm manipulators warn that under no circumstance can people enter the workspace of an operating robot. The reason is simple - due to intended use these robots are strong enough to endanger a human, yet their sensing and intelligence is "too dumb" to be trusted for human safety. In the roboticists' parlance, today's robots are not designed to operate in unstructured environments, that is settings not created specifically for the robot's operation. It is not the function the robot is built for that is the problem - it is the robot's interaction with its environment. The problem is lesser with robot rovers but quite pronounced with arm manipulators.

The way to break this barrier is to design robots fully capable of operating in an unstructured environment, in places where things are unpredictable and must be perceived and decided upon on the fly. This is a new terrain - the required hardware and intelligence are to be more complex and sophisticated than what we know today. In this talk we will review related technical and scientific issues.

BIOGRAPHY

Dr. Vladimir Lumelsky is the head of the Laboratory of Robotics for Unstructured Environments at NASA-Goddard Space Center, and is Adjunct Professor of Computer Science at the University of Maryland-College Park. The long-term goal of the laboratory is to develop robots capable of operating in the uncertain and changing settings likely to arise in future NASA missions. This work builds upon Dr. Lumelsky's work on large sensitive robot skin systems prior to joining NASA in 2004, as a professor at Yale University and later at the University of Wisconsin-Madison (where he was The Consolidated Papers Professor of Engineering). Dr. Lumelsky is the author of three books and over 200 professional papers covering the areas of robotics, computational intelligence, human-machine interaction, human spatial reasoning, massive sensor arrays, bio-engineering, control theory, kinematics, pattern recognition, and industrial automation. He has held a variety of positions in both the public and private sectors: he was Program Director at the National Science Foundation, and has led large technical projects, including development of a universal industrial robot controller at General Electric (GE Research Center), and a joint robot skin development effort with Hitachi Corporation. Dr. Lumelsky also has held temporary positions at the Science University of Tokyo (Japan), Weizmann Institute (Israel) and US South Pole Station, Antarctica. He is the founding Editor-in-Chief of the IEEE Sensors Journal, and has served on editorial boards of other professional journals. He has been guest editor of special issues at professional journals; served on the Administrative Committees of IEEE Robotics Society and Sensors Council; chaired technical committees and working groups; and chaired and co-chaired major international conferences, workshops and special sessions. Dr. Lumelsky has served as a technical expert in legal cases, including multinational litigation. He frequently gives talks at US and foreign universities, government groups, think tanks, and in industry. He is a member of several professional societies, and is a Fellow of IEEE.

August

TUESDAY 28

PROGRAM

PERMIS

08:15	Welcome & Overview
08:30	**Plenary Presentation:** **Maria Gini** ***Methodology for Experimental Research in Multi-robot Systems with Case Studies***
09:30	Coffee Break
10:00	**TUE-AM1 Mobile Robot Performance Evaluation I** ***Chairs: C. Schlenoff & M. Childers*** • Evaluation of Navigation of an Autonomous Mobile Robot [N. Muñoz, J. Valencia, N. Londoño] • Assessing the Impact of Bi-directional Information Flow in UGV Operation: A Pilot Study [M. Childers, B. Bodt, S. Hill, R. Dean, W. Dodson, L. Sutton] • A Common Operator Control Unit Color Scheme for Mobile Robots [M. Shneier, R. Bostelman, J. Albus, W. Shackleford, T. Chang, T. Hong] • How DoD's TRA Process Could be Applied to Intelligent Systems Development [D. Sparrow, S. Cazares] • A Brief History of PRIDE [Z. Kootbally, C. Schlenoff, R. Madhavan]
12:30	Lunch on your own
14:00	**Plenary Presentation:** **Eric Krotkov** ***Measuring Ground Robot Performance***
15:00	Coffee Break
15:30	**TUE-PM1 Mobile Robot Performance Evaluation II** ***Chairs: S. Balakirsky & C. Lundberg*** • Assessment of Man-portable Robots for Law Enforcement Agencies [C. Lundberg, H. Christensen] • Performance Metrics and Evaluation of a Path Planner based on Genetic Algorithms [G. Giardini, T. Kalmar-Nagy] • The Evolution of Performance Metrics in the RoboCup Rescue Virtual Robot Competition [S. Balakirsky, C. Scrapper, S. Carpin] • Robot Simulation Physics Validation [C. Pepper, S. Balakirsky, C. Scrapper] • Design and Validation of a Whegs Robot in USARSim [B. Taylor, S. Balakirsky, E. Messina, R. Quinn]
18:30	Reception

August

TUESDAY 28

Note: Please click on the paper title to view it in pdf.

08:15	Welcome & Overview
08:30	**Plenary Presentation:** **Maria Gini** ***Methodology for Experimental Research in Multi-robot Systems with Case Studies***
09:30	Coffee Break
10:00	**TUE-AM2 Special Session I: Autonomy Levels for Unmanned Systems** ***Organizer: Hui-Min Huang (NIST)*** • Autonomy Levels for Unmanned Systems (ALFUS) Framework: Safety and Application Issues [H-M. Huang] • Evaluation of Autonomy in Recent Ground Vehicles Using the Autonomy Levels for Unmanned Systems (ALFUS) Framework [G. McWilliams, M. Brown, R. Lamm, C. Guerra, P. Avery, K. Kozak, B. Surampudi] • A Methodology for Testing Unmanned Vehicle Behavior and Autonomy [D. Gertman, C. McFarland, T. Klein, A. Gertman, D. Bruemmer] • Standardizing Measurements of Autonomy in the Artificially Intelligent [A. Hudson, L. Reeker]
12:30	Lunch on your own
14:00	**Plenary Presentation:** **Eric Krotkov** ***Measuring Ground Robot Performance***
15:00	Coffee Break
15:30	**TUE-PM2 Special Session II: Human Robot Interface Issues** ***Organizers: Salvatore Schipani & Brian Antonishek (NIST)*** • Maze Hypothesis Development in Assessing Robot Performance During Teleoperation [S. Schipani, E. Messina] • Human System Performance Metrics for Evaluation of Mixed-Initiative Heterogeneous Autonomous Systems [L. Billman, M. Steinberg] • Concepts of Operations for Robot-Assisted Emergency Response and Implications for Human-Robot Interaction [J. Scholtz, B. Antonishek, B. Stanton, C. Schlenoff] • Multimodal Displays to Enhance Human Robot Interaction On-the-Move* [E. Haas, C. Stachowiak]
18:30	Reception

*A multi-modal information system will be demonstrated.

August

WEDNESDAY 29

PROGRAM

PERMIS

Note: Please click on the paper title to view it in pdf.

08:15	**Overview**
08:30	**Plenary Presentation:** **Illah Nourbakhsh** ***Formalizing Educational Human-Robot Collaboration***
09:30	**Coffee Break**
10:00	**WED-AM1 Autonomy Vs Intelligence** ***Chairs: J. Gunderson & J. Evans*** • Autonomy (What's it Good for?) [J. Gunderson, L. Gunderson] • Definitions and Measures of Intelligence in Deep Blue and the Army XUV [J. Evans] • Automotive Turing Test [S. Kalik, D. Prokhorov] • Autonomous Robots with Both Body and Behavior Self-Knowledge [B. Gordon] • A Cognitive-based Agent Architecture for Autonomous Situation Analysis [G. Berg-Cross, W-T. Fu, A. Kwon]
12:30	**Lunch on your own**
14:00	**Featured Presentation:** **Vladimir Lumelsky** ***Human-Robot Interaction in Physical Proximity: Issues and Prospects***
15:00	**Coffee Break**
15:30	**WED-PM1 Human Machine Interaction** ***Chairs: N. Dagalakis & A. Steinfeld*** • Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop [A. Steinfeld, S. Bennett, K. Cunningham, M. Lahut, P-A. Quinones, D. Wexler, D. Siewiorek, J. Hayes, P. Cohen, J. Fitzgerald, O. Hansson, M. Pool, M. Drummond] • Survey Measures for Evaluation of Cognitive Assistants [A. Steinfeld, P-A. Quinones, J. Zimmerman, S. Bennett, D. Siewiorek] • Development of Tools for Measuring the Performance of Computer Assisted Orthopaedic Hip Surgery Systems [N. Dagalakis, Y. Kim, D. Sawyer, C. Shakarji] • Haptic Feedback System for Robot-Assisted Surgery [J. Desai, G. Tholey, C. Kennedy]
19:30	**Banquet**

PERMIS

PROGRAM

August

WEDNESDAY 29

Note: Please click on the paper title to view it in pdf.

08:15	Overview
08:30	**Plenary Presentation:** **Illah Nourbakhsh** ***Formalizing Educational Human-Robot Collaboration***
09:30	Coffee Break
10:00	**WED-AM2 Panel Discussion I: Can the Development of Intelligent Robots be Benchmarked? Concepts and Issues from Epigenetic Robotics** ***(Moderator: Gary Berg-Cross, EM & I)*** • Douglas Blank, Bryn Mawr College • James Marshall, Sarah Lawrence College • Lisa Meeden, Swarthmore College • Charles Kemp, Georgia Tech. • Chad Jenkins, Brown University
12:30	Lunch on your own
14:00	**Featured Presentation:** **Vladimir Lumelsky** ***Human-Robot Interaction in Physical Proximity: Issues and Prospects***
15:00	Coffee Break
15:30	**WED-PM2 Special Session III: Space/Aerial Robotics** ***Organizer: Edward Tunstel (JPL)*** • Prototype Rover Field Testing and Planetary Surface Operations [E. Tunstel] • Planning to Fail - Reliability as a Design Parameter for Planetary Rover Missions [S. Stancliff, J. Dolan, A. Trebi-Ollennu] • A Decision Space Compression Approach for Model Based Parallel Computing Processes [R. Bonneau, G. Ramseyer] • Physically-Proximal Human-Robot Collaboration for Air and Space Applications [E. Atkins]
19:30	Banquet (Adam Jacoff, NIST)

August THURSDAY 30

Note: Please click on the paper title to view it in pdf.

08:15	Overview
08:30	**Plenary Presentation:** **Alex Zelinsky** ***Building Autonomous Systems of High Performance, Reliability and Integrity***
09:30	Coffee Break
10:00	**THU-AM1 Performance Assessment of Algorithms** ***Chairs: R. Lakaemper & S. Spetka*** • Analyzing the Performance of Distributed Algorithms [R. Lass, E. Sultanik, W. Regli] • An Agent Structure for Evaluating MAS Performance [C. Dimou, A. Symeonidis, P. Mitkas] • Information Management for High Performance Autonomous Intelligent Systems [S. Spetka, S. Tucker, G. Ramseyer, R. Linderman] • Efficient Monte Carlo Computation of Fisher Information Matrix using Prior Information [S. Das, J. Spall, R. Ghanem] • Performance of 6D LuM and FFS SLAM -- An Example for Comparison using Grid and Pose Based Evaluation Methods [R. Lakaemper, A. Nuchter, N. Adluru, L. Latecki]
12:30	Lunch on your own
14:00	**THU-PM Panel Discussion II: (Re-)Establishing or Increasing Collaborative Links Between Artificial Intelligence and Intelligent Systems** ***(Moderator: Brent Gordon, NASA-Goddard)*** • James Albus, Senior Fellow, Intelligent Systems Division, NIST • Ella Atkins, Associate Professor, University of Michigan • Henrik Christensen, Director, Center for Robotics and Intelligent Machines, Georgia Tech. • Larry Reeker, Computer Scientist, Information Technology Laboratory, NIST
15:30	Coffee Break
16:00	Adjourn

August THURSDAY 30

Note: Please click on the paper title to view it in pdf.

08:15	Overview
08:30	**Plenary Presentation:** **Alex Zelinsky** ***Building Autonomous Systems of High Performance, Reliability and Integrity***
09:30	Coffee Break
10:00	**THU-AM2 Special Session IV: Smart Assembly Systems** ***Organizers: Robert Tilove (GM) & John Slotwinski (NIST)*** • Smart Assembly: Industry Needs and Challenges [J. Slotwinski, R. Tilove] • Science based Information Metrology for Engineering Informatics [S. Rachuri] • Evaluating Manufacturing Control Language Standards: An Implementer's View [T. Kramer] • Interoperability Testing for Shop-Floor Inspection [F. Proctor, W. Rippey, J. Horst, J. Falco, T. Kramer] • A Virtual Environment-Based Training Systems for Mechanical Assembly Operations [M. Schwartz, S. Gupta, D. Anand, R. Kavetsky]
12:30	Lunch on your own
14:00	**THU-PM Panel Discussion II: (Re-)Establishing or Increasing Collaborative Links Between Artificial Intelligence and Intelligent Systems** ***(Moderator: Brent Gordon, NASA-Goddard)*** • James Albus, Senior Fellow, Intelligent Systems Division, NIST • Ella Atkins, Associate Professor, University of Michigan • Henrik Christensen, Director, Center for Robotics and Intelligent Machines, Georgia Tech. • Larry Reeker, Computer Scientist, Information Technology Laboratory, NIST
15:30	Coffee Break
16:00	Adjourn

PERMIS AUTHOR INDEX

ACKNOWLEDGMENTS

These people provided essential support to make this event happen. Their ideas and efforts are very much appreciated.

Website and Proceedings

Debbie Russell (Chair)

Local Arrangements

Jeanenne Salvermoser (Chair)

Jennifer Peyton

Conference and Registration

Kathy Kilmer (Chair)

Teresa Vicente

Mary Lou Norris

Angela Ellis

Finance

Betty Mandel (Chair)

Thank you PerMIS attendees!

Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
100 Bureau Drive, MS-8230
Gaithersburg, MD 20899
http://www.isd.mel.nist.gov/

Evaluation of Navigation of an Autonomous Mobile Robot

N. D. Muñoz, J. A. Valencia, N. Londoño
Polytechnic Jaime Isaza Cadavid,
University of Antioquia
Medellín-Colombia
giraa@udea.edu.co

Abstract— In this paper, the navigation of an autonomous mobile robot is evaluated; Several metrics are described; These metrics, collectively, provide an indication of the quality of the navigation, useful for comparing and analyzing control architectures of mobile robots. Two control architectures are simulated and compared in an autonomous navigation mission.

Keywords**:** *Performance Metrics, Mobile Robots, Control Architectures.*

I. Introduction

An autonomous mobile robot has to combine mission execution with fast reaction to unexpected situations. To overcome this problem, various types of control architectures for mobile robot have been designed, treating to improve performance of the navigation system of a mobile robot for the execution of the mission.

Despite the wide variety of studies and research on robot navigation systems, quality metrics are not often examined, which makes it difficult to make an objective comparison of performance [11]; in general, use of quality metrics is limited to measuring and analyzing the length of the path or the time needed by the robot to complete the task. Additionally, the lack of consensus on how to define or measure these systems impedes rigor and prevents evaluation of progress in this field and compare its different capabilities [5].

However, by applying navigation comparison metrics of a mobile robot, such as the trajectory (path) length, collision risk and smoothness of trajectory, using a protocol, that is in a systematic and ordered way, experimental works on mobile robots navigation control algorithms can be systematized, and this will help researchers to decide which architecture should be implemented in the vehicle.

This paper presents the methodology used for the evaluation of the experiment. First, various performance metrics used in the navigation of mobile robots are described, then the protocol to be followed in the evaluation of the experiment is defined, and finally the obtained results are presented with the aid of a simulation software.

II. Quality indexes on trajectories

There are various metrics that can be used to evaluate the performance of a navigation system, but none of them is able to indicate the quality of the whole system. Therefore it is necessary to use a combination of different indexes that quantify different aspects of the system. Having a good range of performance measurements is useful for: Optimizing algorithm parameters, testing navigation performance within a variety of work environments, making a quantitative comparison between algorithms, supporting algorithm development and helping with decisions about the adjustments required for a variety of aspects involved in system performance [3].

In general terms, navigation performance metrics can be classified in the following order of importance: Security in the trajectory indexes or proximity to obstacles, metrics that consider the trajectory towards the goal and metrics that evaluate the smoothness of the trajectory.

A. Security metrics

These metrics express the relationship between the security with which the robot travels through a trajectory, taking into account the distance between the vehicle and the obstacles in its path [2].

Security Metric-1 (SM1): Mean distance between vehicle and the obstacles through the entire mission measured by all of the sensors; the maximum value will be produced in an environment free of obstacles. If the deviation of the index from its maximum value is low, it means that the route taken took fewer obstacles.

Security Metric-2 (SM2): Minimum means distance to the obstacles. This is taken from the average of the lowest value of the n sensors. This index gives an idea of the risk taken through the entire mission, in terms of the proximity to an obstacle. In an environment free of obstacles SM1 = SM2 is satisfied.

Minimum Distance (Min): Minimum distance between any sensor and any obstacle through the entire trajectory. This index measures the maximum risk taken throughout the entire mission.

B. Dimensional metrics

The trajectory towards the goal is considered in its spatial and temporal dimensions. In general, it is assumed that an optimal trajectory towards the goal is, whenever possible, a line with minimum length and zero curvature between the initial point (x_i,y_i) and the finishing point (x_n,y_n), covered in the minimum time.

Length of the Trajectory Covered (P_L) is the length of the entire trajectory covered by the vehicle from the initial point to the goal. For a trajectory in the x-y plane, composed of n points, and assuming the initial point as (x_1, f(x_1)) and the goal as (x_n, f(x_n)), P_L can be calculated as:

$$P_L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (f(x_{i+1}) - f(x_i))^2} \tag{1}$$

Where (x_i, f (x_i)), i = 1, 2. . . n are the n points of the trajectory in Cartesian coordinates [6].

The length of a trajectory given by y = f(x), in the x-y plane between the points (a, f(a)) and (b, f(b)), can also be calculated as (Selekwa,2004)

$$P_{Laprox} \cong \int_a^b \sqrt{1 + (f'(x_i))^2}\, dx \tag{2}$$

Mean distance to the goal (Mgd): This metric can be applied to robots capable of following reference trajectories. An important aspect when determining the quality of the navigation system of a robot is the ability to follow a trajectory that aims to reach a goal; so, to evaluate the quality of the execution of the trajectory, the mean distance between the vehicle and goal is analyzed. The difference becomes more significant if the distance covered is shorter [9]. The mean distance to the goal is defined by the square of the proximity to the goal distance l_n, integrated across the length of the trajectory and normalized by the total number of points n:

$$l_n = \min\left(\forall n\left(\sqrt{(x_i - x_n)^2 + (f(x_i) - f(x_n))^2}\right)\right) \tag{3}$$

$$Mgd = \frac{\int_0^l l_n^2\, ds}{n} \tag{4}$$

Control Periods (LeM): It is the amount of control periods. This metric relates to the number of decisions taken by the planner to reach the goal, if the robot moves with lineal and constant speed (v). This gives an idea of the time needed to complete the mission [2].

C. Smoothness metrics

The smoothness of a trajectory shows the consistency between the decision-action relationship taken by the navigation system, and also, the ability to anticipate and to respond to events with sufficient speed [9]. The smoothness in the way a trajectory is generated is a measure of the energy and time requirements for the movement; a smooth trajectory allows translates into energy and time savings [4]. Additionally a smooth trajectory is also beneficial to the mechanical structure of the vehicle.

Bending Energy (B_E): This is a function of the curvature, k, used to evaluate the smoothness of the robot's movement. For curves in the x-y plane, the curvature, k, at any point (x_i,f(x_i)) across a trajectory is given by:

$$k(x_i, f(x_i)) = \frac{f''(x_i)}{(1 + (f'(x_i))^2)^{\frac{3}{2}}} \tag{5}$$

The bending energy can be understood as the energy needed to bend a rod to the desired shape [1]. B_E can be calculated as the sum of the squares of the curvature at each point of the line k(x_i,y_i), along the length of the line L. So, the bending energy of the trajectory of a robot is given by:

$$B_E = \frac{1}{n} \sum_{i=1}^{n} k^2(x_i, f(x_i)) \tag{6}$$

Where k(xi, yi) is the curvature at each point of the trajectory of the robot and *n* is the number of points in the trajectory.

The value of B_E is an average and does not show with sufficient clarity that some trajectories are longer than others. Therefore, TB_E can be used instead; this metric takes into account the smoothness and length of the trajectory simultaneously.

TB_E is defined by

$$TB_E = \int_a^b k^2(x)\, dx \tag{7}$$

And numerically,

$$TB_E = \sum_{i=1}^{n} k^2(x_i, f(x_i)) \tag{8}$$

The straighter the trajectory, the lower B_E and TB_E values will be, which is desirable since the energy requirement is increased according to the increase in the curvature of the trajectory.

Smoothness of Curvature (Smoo) is defined by the square of the change in the curvature k of the trajectory of a vehicle with respect to the time, integrating along the length of the trajectory and normalized by the total time t [9].

$$Smoo = \frac{\int_0^l \left(\frac{dk}{dt}\right)^2 ds}{t} \tag{9}$$

III. Definition of the evaluation protocol for the experiments

The control architectures under analysis provide basic capabilities for the mobile robot, such as the ability to evade obstacles and to generate a trajectory towards a goal.

The control architecture algorithms were simulated according to the characteristics of the mobile robot platform Giraa_02 used in the laboratory, which has a cylindrical structure of 30cm diameter and approximately 20cm height; figure 1. It has 8 infrared sensors distributed equally around the robot's circumference, and these have a range of 26.5cm and a 15 degree detection cone; the vehicle has a differential locomotion system, and its position is provided by an optical encoder and a magnetic compass [8].

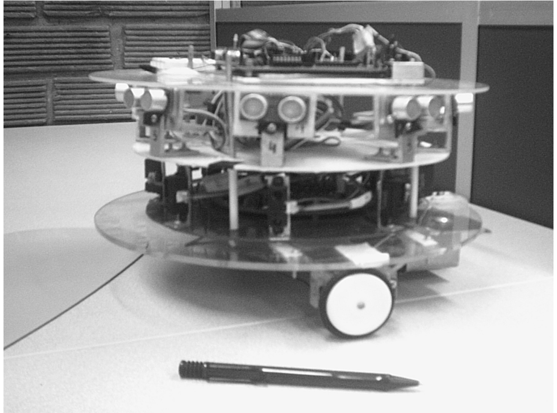


Fig. 1: Mobile Robot Giraa_02

Data acquisition in the mobile robot, which occurs during each control period, consists of the current position of the robot and its orientation (x_i,y_i,θ_i). The eight (8) proximity sensors are also read, the maximum reading being 26.5 cm, so that, if the robot spends n control periods reaching the goal, there is an array of n x 11, and n sampling points per 11 pieces of data (3 coordinates and 8 sensors).

Taking into account that the objective is to execute a navigation mission from a starting point to a final point (navigation mission towards a goal), an order of importance can be established for evaluating the navigation characteristics, as follows:

1. The mean distance between the vehicle and the obstacles during the trajectory
2. The distance covered by the vehicle between the starting point and the goal
3. The time needed to complete the mission
4. The smoothness of the trajectory

The first point considers the security of the trajectory and measures the risk taken by the robot in its movement towards the goal. The second and third points measure aspects related to the planning of the trajectory, and the fourth point considers the quality of the trajectory according to the energy and time required for the movement.

These characteristics can be analyzed using the following set of performance metrics:

1. SM1, SM2 and Min are proposed for evaluating security.
2. PL and LeM are proposed for evaluating the trajectory
3. TB_E is proposed for evaluating the smoothness of the trajectory.

For general purposes, only one metric is required for each one of the 3 categories described in section 2, but the use of various metrics helps to improve the analysis. In our case, the indexes were selected according to the abilities of the mobile robot GIRAA_02, considering the information provided by its data acquisition system; the readings from all the sensors are available, for each point of the path, allowing the calculation of SM1, SM2, and Min. The Mgd index does not apply in this navigation mission since it applies when a trajectory is followed; TB_E is proposed because it analyses the smoothness and length of the path. Also, this metric is numerically simpler and more precise, making it easier to calculate than the other metrics.

IV. Development of the test and results

A. Control Architecture 1

This is a reactive architecture based on a potential field method, which produces two different behaviors: first, attraction to the goal, and second, repulsion of the obstacles. The planning of the movement consists in the proper combination of both behaviors in such a way that the robot reaches the goal without collisions. This combination is achieved using a vector sum [7].

B. Control Architecture 2

This control architecture is based on behaviors denominated AFREB "adaptive fusion of reactive behaviors" [12]. By using a neural net, an appropriate combination of the behaviors can be achieved, so that the system is able to realize more complex tasks, such as navigation towards a goal, while evading obstacles in its path.

TABLE I

ROBOT PERFORMANCE RESULTS

Performance Index	Control Architecture 1	Control Architecture 2
SM1 [cm]	26.1630	25.6276
SM2 [cm]	18.3750	17.3750
Min [cm]	11	7
PL [cm]	562.7810	581.9479
LeM	283	292
TB_E	0.8535	0.0846

Maximum SM1= 26.5cm

C. Simulations

The first stage of the work focused on evaluating the metrics using conventional simulators that permit the validation of the effectiveness and the limitations of the algorithms. SRM simulator was used on the Matlab platform. A 6m x 4m flat, structured environment with static obstacles was created for the execution of a navigation mission between two points (towards a goal);The starting position was 50,50 and the position of the goal was 500,300 (scenario 1); The paths generated by the architectures are shown in figures 2 and 3. Table 1 summarizes the results obtained from the simulation using both control architectures according to the quality metrics described.

In an obstacle-free path, maximum SM1 would be 26.5cm for the Giraa_02, and a similar result occurred with SM2 and Min.

D. Analysis of results

In scenario 1, Architecture 1 uses less control periods, and consequently takes less time to complete the mission, and covers a safer and shorter path. Architecture 2 covers a smoother path, figures 4 and 5 show a smaller change in the orientation during each control period, with consequent energy saving and less structural stress on the robot.

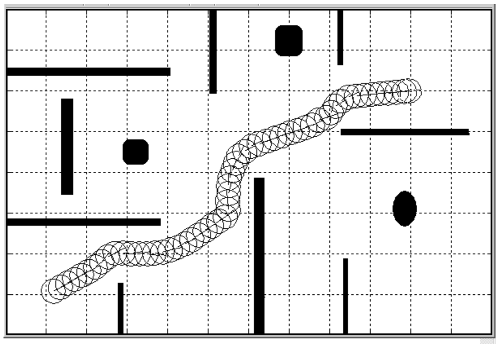


Fig. 2: Path generated by architecture 1

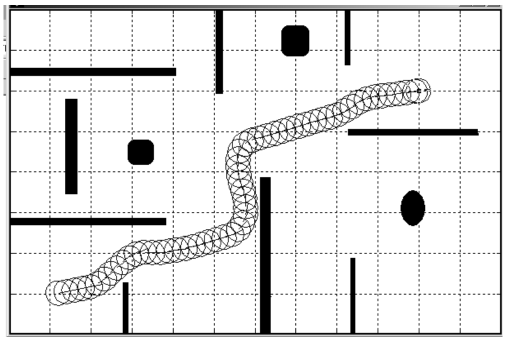


Fig.3: Path generated by architecture 2

From table 1 it can be deduced that the difference between both architectures in the trajectory and time taken is only 3.3% and 3.1% respectively. The robot equipped with architecture 2 passed a minimum 7 cm from any obstacle, which is acceptable for a 30cm diameter robot; also, it showed approximately 65% less bending energy than architecture 1. For these reasons, architecture 2 is considered the best choice.

In the other scenarios (figures 6), Architecture 1 tends to generate safer trajectories, because the robot normally transits through zones that are farther from the obstacles. This is because the closer the robot is to the obstacles, the higher the repulsion potential. Although Architecture 2 is governed by the same repulsion principle, the command that finally guides the robot is a combination of 5 different behaviors, and this means that the role of the repulsion potential is less important, and collisions are less likely than with Architecture 1. The Main difference in all the simulations is that Architecture 2 generates smoother trajectories than Architecture 1. Table II summarizes the results obtained in all scenarios.

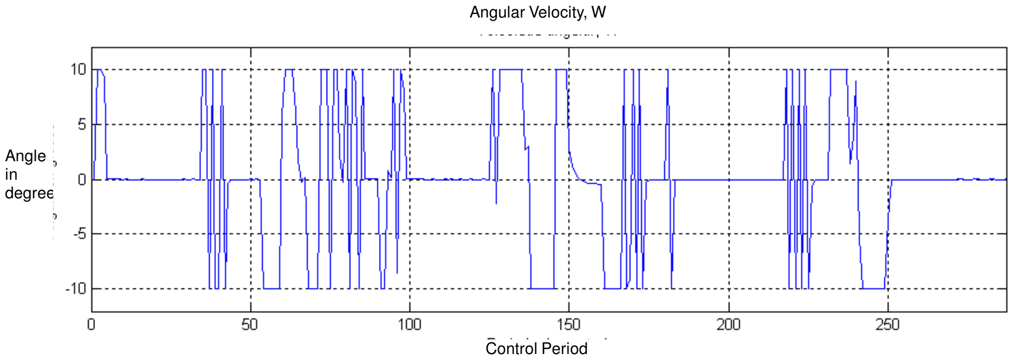


Fig. 4 Smoothness of the trajectory, change in the robot heading each control period, generated by architecture 1

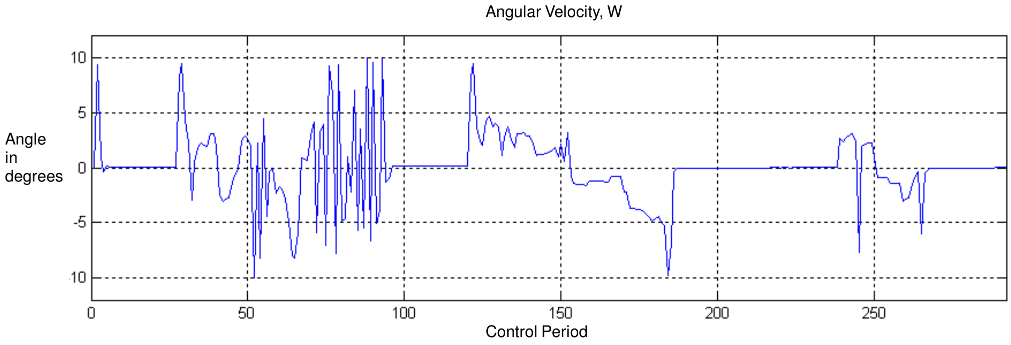


Fig. 5 Smoothness of the trajectory, change in the robot heading each control period, generated by architecture 2

TABLE II
ROBOT PERFORMANCE RESULTS (ALL SCENARIOS)

Metric	SM1 [cm]		SM2 [cm]		Min [cm]		PL [cm]		LeM		TB_E	
Scenario	Arc. 1	Arc. 2	Arc. 1	Arc. 2	Arc. 1	Arc. 2	Arc. 1	Arc. 2	Arc. 1	Arc. 2	Arc. 1	Arc. 2
1	26.1	25.6	18.3	17.3	11	7	562.7	581.9	283	292	0.2463	0.0846
2	25.9	25.7	13.0	14.0	3	7	441.8	429.9	222	216	0.2810	0.0718
3	25.4	23.9	10.0	8.9	3	3	456.7	462.9	234	235	0.5873	0.0120
4	25.0	24.4	13.0	12.4	7	3	395.7	359.9	199	181	0.4007	0.0140
5	25.9	24.9	19.4	16.4	15	3	275.8	259.9	139	131	0.1626	0.0394
6	26.0	25.9	19.7	22.6	7	11	229.9	229.9	116	116	0.1722	0.0469

SM1 maximum = 26.5cm

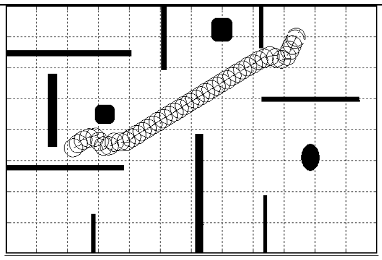
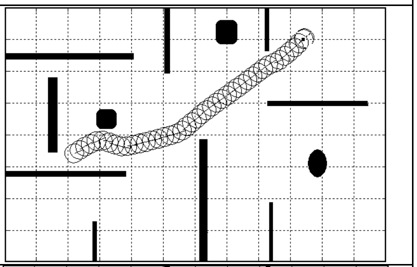
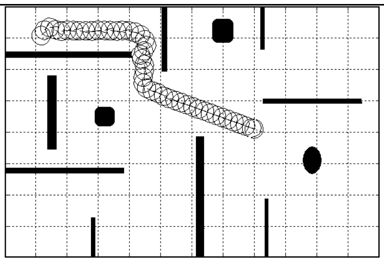
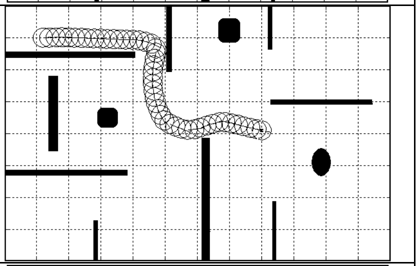
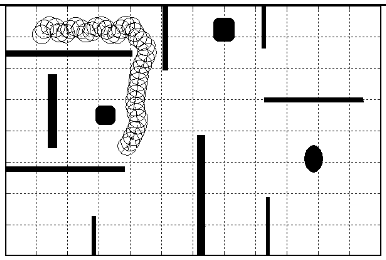
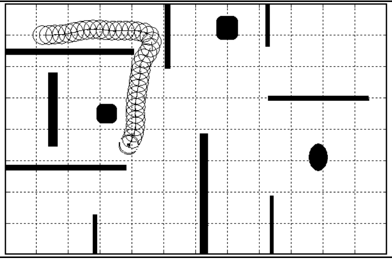
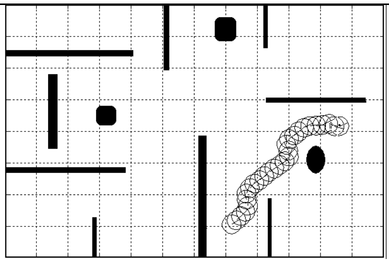
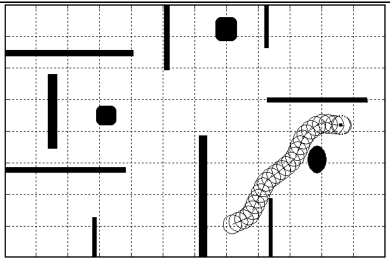
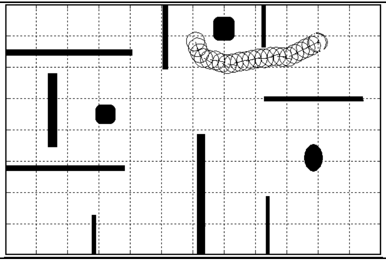
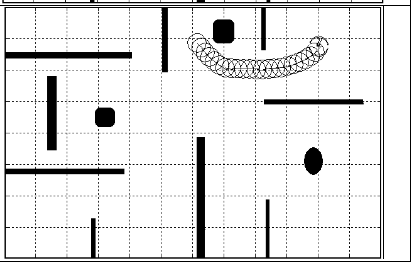
Simulation	Architecture 1	Architecture 2
Scenario 2 start point (100,170) Goal (470,350)		
Scenario 3 start point (50,350) Goal (400,205)		
Scenario 4 start point (50,350) Goal (195,175)		
Scenario 5 start point (350,50) Goal (530,210)		
Scenario 6 start point (300,350) Goal (500,340)		

Fig. 6 Paths generated by the control architectures in other scenarios

The wall-following behaviors CW and CCW in Architecture 2, enable the robot to transit through narrow zones like corridors, while keeping a safe distance from the obstacles and also generating smooth trajectories; this does not happen with Architecture 1, as shown in sceneries 3 and 4. In general terms, Architecture 2 exhibits better performance, and the bending energy index is always lower than in Architecture 1, even in scenarios 2, 4 and 5, generating shorter trajectories and using less time to complete the mission.

V. Conclusion

The results obtained demonstrate the need to establish a procedure that can be used to analyze and compare control architectures using various performance metrics, since, as we have shown, the shortest path or the algorithm that enables the robot to reach the goal most quickly is not necessarily the most appropriate.

One of the most important contributions of the project is the definition of a comparison protocol for control architectures, based on various quality indexes of the trajectories. The protocol can be used as a tool in the analysis of the control algorithms for autonomous navigation missions. Although some of the metrics are intuitive, it was demonstrated that together they are useful for systematizing experimental control algorithms for the navigation of mobile robots. The results show the importance of setting up an ordered and methodical procedure in order to analyze and compare the various performance metrics of the different control architectures. As was seen, the algorithm that produces the shortest or fastest route to the goal is not always the most energy efficient.

The results show better performance for architecture 2 in both simple and complex circumstances. These scenarios demonstrates that architecture 2 performs satisfactorily where architecture 1 is unable to perform, or performs less efficiently. In the simulations, the robot Giraa_02 programmed with architecture 2, was able to navigate satisfactorily while reaching the goal, even through complex and narrow trajectories; generally producing non-oscillatory and smooth movements. This architecture demonstrates that combining behaviors using a neural net is a good option for avoiding conflicts between behaviors. In the same way, the fusion of behaviors through a lineal combination is a simple and efficient method; such a method is necessary given the limited computational capabilities of the robot. For these reasons Architecture 2 was implemented in the Giraa_02 robot in a real environment.

Acknowledgement

This work was partial result of the FPPI06-1-04, FPPI06-1-07, MC02-1-09 and POLI 2061080185 projects, financed by the University of Antioquia and Polytechnic Jaime Isaza Cadavid. The authors wish to thanks to ICARO, GIMEL, Engineering & Software research Groups. http://ingenieria.udea.edu.co/giraa.

References

[1] E. Aguirre and A. Gonzalez, "Fuzzy Behaviors for mobile robot navigation, design, coordination and fusion", *International Journal of Approximate Reasoning*, vol. 25, 2000, pp. 255-289.

[2] J. Álvarez, Planificación del movimiento de vehículos autónomos basada en sensores. Tesis doctoral, Universidad de Oviedo, Oviedo, Spain, 1998, pp. 178.

[3] G. Cielniak, A. Treptow and T. Duckett, "Quantitative Performance Evaluation of a People Tracking System on a Mobile Robot", *Proceedings of the European Conference on Mobile Robots (ECMR)*, Ancona, Italy, 2005.

[4] S. Dongqing, "Aerial robot navigation in cluttered urban environments", PhD Thesis, The Florida State University, Florida, USA, 2006, pp. 87.

[5] J. Evans and E. Messina, "Performance Metrics for Intelligent Systems", *Proceeding of the Performance Metrics for intelligent Systems Workshop,* Gaithersburg, MD, August 14-16 of 2000.

[6] Y. Guo and J. Wang, "A new performance based motion planner for nonholonomic mobile robots", *Proceedings of the 3rd performance metrics for the Intelligent Systems Workshop (PerMIS'03) NIST,* Gaithersburg, MD, September 2003.

[7] J.C Latombe, "Robot Motion Planning", Kluwer Academic Publishers, 4th Edition, Boston, 1996.

[8] N. Muñoz, C. Andrade, N. Londoño. "Diseño y construcción de un robot móvil orientado a la enseñanza e investigación", *Ingenieria & Desarrollo Ed. 9,* 2006.

[9] J. Rosenblatt, "DAMN: Distributed Architecture for Mobile Navigation. PhD. Thesis", Carnegie Mellon University Robotics Institute, Pittsburg, PA, 1997.

[10] M. Selekwa, E. Collins and J. Combey, "Multivalued Verus univalued Reactive Fuzzy Behavior Systems for Navigation Control of Autonomous Ground Vehicles", *Proceedings from the 17th annual Florida Conference on the Recent Advances in Robotics FCRAR2004,* May 20, 2004.

[11] S. Wong, L. Middleton and B. MacDonald, "Performance metrics for robot coverage task", *Proceedings Australasian Conference on Robotics and Automation ACRA,* Auckland, New Zealand, 2002, pp. 7-12.

[12] A. Zalzala and A. Morris, *Neural Networks for Robotic Control,* Ellis Horwood, 1996, pp.278.

Assessing the Impact of Bi-directional Information Flow in UGV Operation: A Pilot Study

Marshal A. Childers, Barry A. Bodt, Ph.D., Susan G. Hill, Ph.D.
U.S. Army Research Laboratory, Aberdeen Proving Ground, MD
mchilders@arl.army.mil, babodt@arl.army.mil, sghill@arl.army.mil

Robert M. Dean, William F. Dodson, Lyle G. Sutton
General Dynamics Robotics Systems, Westminster, MD
rdean@gdrs.com, wdodson@gdrs.com, lsutton@gdrs.com

Abstract**—**In June 2007, the Robotics Program Office of the U.S. Army Research Laboratory and General Dynamics Robotics Systems (GDRS) engaged in an exploratory assessment of how bi-directional information flow impacts Unmanned Ground Vehicle (UGV) operation. The purposes of the pilot study were to frame scenarios, protocol, infrastructure, and metrics for a more formal experiment planned for the fall of 2007 while providing current data feedback for the architecture developers. The study was conducted at Fort Indiantown Gap, PA over two distinct areas of rolling vegetated terrain using the eXperimental Unmanned Vehicle (XUV). In this paper, we will share the preliminary findings of the impact of bi-directional information flow on observed robotic behavior, discuss the associated impact on the operator, and relate lessons learned to the planning of our fall 2007 experiment.

Keywords: *bi-directional information flow, perceptive planning, deliberative planning, pilot study, unmanned ground vehicles*

I. INTRODUCTION

A brief background helps motivate the current study. In FY 2003, the RPO and GDRS conducted, with testing oversight by the National Institute of Standards and Technology, an extensive three-site experiment of an autonomous navigation system (ANS) [1]. The ANS relied on perceptive level planning to achieve a manually pre-determined route of way points in rolling desert, rolling vegetated and urban terrain. The ANS was given a Technology Readiness Level 6 designation by Future Combat Systems in part due to this study. Interim advances in the Soldier Machine Interface (SMI) greatly simplified manual route planning, while perception algorithms and hardware continued to mature. More recent developments in the architecture allow for deliberative planning in a move toward tactically intelligent behaviors.

Higher level deliberative planning draws on the objective of the operation and the global map of *a priori* information (elevation and feature data). Deliberative planning consists of separate layers to independently assess costs for traversing terrain; the current configuration considers costs associated with mobility, time, coverage, exposure, and threat. Those layers are combined using a weighted heuristic into a single planning layer for use by the route planning algorithm. Different weight combinations map into various tactical concepts, which allows the SMI to provide explicit choices to the user such as “prefer roads” or “stealth”; weights can be individually set during experimentation. Deliberative and perceptive level planning are integrated through the field cost interface (FCI) and best information planning (BIP). Local perception provides costs at ~5 Hz rate for local paths finishing along the arc formed by the sensor range. FCI is a feature that provides a bridge between deliberative layer planning and local planning by assigning costs at ~1 Hz rate along the perimeter of the sensor range representing the entry points for continued routes to the objective way point. BIP uses the sensed data flowing up from the perceptive level to update the deliberative planning map. BIP uses the sensed data flowing up from the perceptive level to update the deliberative planning map. Using this updated information may be especially useful with imperfect *a priori* knowledge of the terrain. It is this bi-directional flow of information that is the focus of the study.

II. DATA COLLECTION

The data collection involves a series of informal comparative tests in which a simple route, with a few widely separated way points, is traversed by the robot. The first condition isolates performance of the perceptive layer planning as a baseline. The second condition makes the global map available for revising the route between pre-determined way points in consideration of mobility. Under this condition, BIP is available to assist perceptive level decisions. A third condition again uses the global map for revising the route between way points but in addition exercises the FCI, taking into consideration the current robot location on the map while the run is in progress. A fourth condition allows the global map information in establishing the initial route and enables the mobility deliberative planning layer through the FCI with BIP assisting perceptive level decisions. Subsequent conditions exercise other deliberative planning layers and vary weights in the aforementioned heuristic.

These planning configurations were exercised over two distinct course areas in situations intended to highlight the value added by the deliberative planning layer. Routes were selected in both locations to provide a stiff challenge for perceptive level planning that may benefit from bi-directional information flow. For example, a vegetation-formed cul-de-sac provides such a challenge. Once in the cul-de-sac, the perceptive level is unlikely to be able to determine an exit path. However, when the BIP feature augments the global map, an exit path may appear. Some runs focused on impacting the route based on these layers.

The first phase of data collection was performed at Area B12 at Fort Indiantown Gap, PA and the second phase at Area A1. Area B12 (Figure 1) is consistent with rolling vegetated terrain and is mostly cross country over open fields with high vegetation, but also contains woods, thick brush, large rocks, gravel road, unimproved trails, and mild changes in elevation. The areas wherein a priori terrain feature data was made available for route planning are highlighted by light green shading; for areas not shaded the only a priori data used for planning was elevation data. An initial pre-planned route consisting of three way points incorporated a cul-de-sac located in the vicinity of the second way point. The XUV traversed the route using an onboard Laser Detection and Ranging (LADAR) sensor and corresponding algorithms to detect and avoid encountered obstacles that were unknown a priori. A trail, high brush and trees were encountered along the way to an elevated position. In figure 1, the route begins on the left in a clearing and continues toward an area of trees and brush not present on the global map. After achieving the second way point, the robot turned toward the final way point marked to the far right. The exact path the robot traveled appears in red.

Area A1 is characterized by relatively flat terrain, with open ground being more grassland; trees and brush occur in patches and dense woods and marshy areas are present. In Figure 2, the planned route in yellow begins on a trail in the lower left portion of the figure and proceeds through woods, which are present in the area but were intentionally removed from the global map to encourage interaction with the feature. After achieving the second way point beyond the woods, this path exhibits the planner response to a defined exposure point located further to the right beyond the figure view. The exposure deliberative layer attempts to minimize the line of sight to a position. The path allows the robot to achieve the second way point, but then directs the robot to the tree line, the best option in consideration of exposure, as it proceeds to the third way point. The route sends the robot toward an opening in the tree line, followed by a small meadow, and finally leads it through some tree clusters where the third way point is located on the way to the fourth and final way point. Notice the robot icon and red track on this run has not achieved the final way point in the upper right. This path to a marsh resulted in an emergency stop (e-stop) which was initiated by the safety operator in order to prevent the vehicle from entering the marsh.

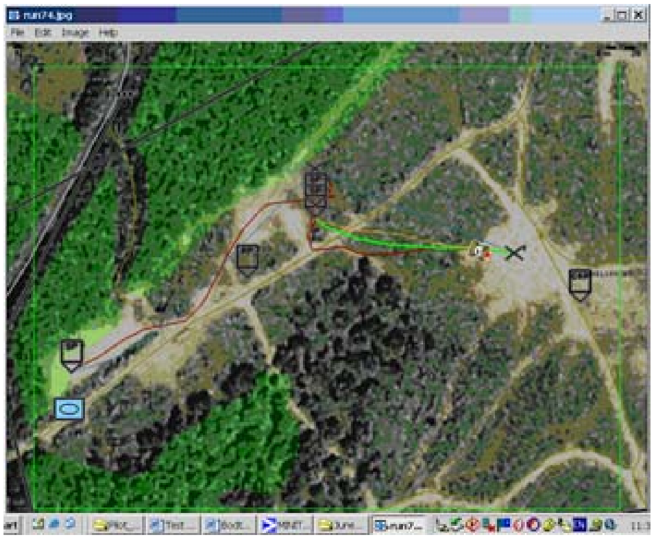


Fig. 1. Area B12

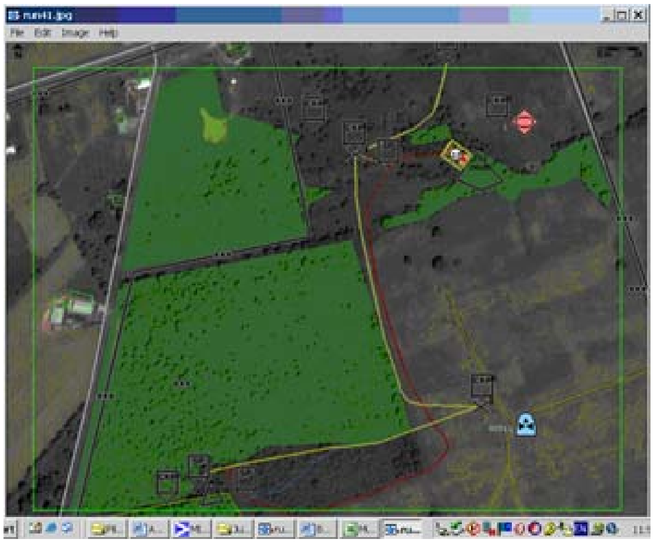


Fig. 2. Area A1

Data collection protocol evolved over the two-week study as performance of technology was observed and as new situations occurred. Initially, rules were established for end of mission, administrative stops, and e-stops, similar to past experimentation. To accommodate BIP, the protocol was refined during the first week at Area B12 to specify when to allow the global map to be updated through BIP. This feature was not automated for the study. Rather, an action on part of the operator to execute a re-plan was required. This re-plan was permitted, when the XUV called for help, usually after three back-ups from the XUV failed to provide a clear path (successive back-ups are part of the ANS and are used to provide better perspective when the robot otherwise does not see a clear path ahead). Re-planning was executed after instructing the XUV to backtrack to the location it occupied after the final, 15m back-up and before the XUV called for help. Another modification was made to this procedure when testing moved to Area A1. There it was determined that when the re-plan was to be executed, the robot should first be repositioned, heading along the re-planned path. Other protocol adjustments responded to a recurrent "off-course" message and allowed aborted runs if the operator was unable to use the cameras for teleoperation.

III. DATA INTERPRETATION

A. Measurement

Measures of performance are elusive. Previous experiments focused on progressing along a pre-determined route safely, as fast as possible, and with minimum operator interventions. Consequently, success or failure, speed, operator intervention frequency and duration were natural metrics for performance. Tactical intelligence, however, provides a far greater challenge due to the qualitative flavor of "how well" the robot has progressed over the route. "How well" must be assessed in consideration of standard tactical considerations that are not crisply defined for this technology and may require trade-off decision making. Measurement of operator workload perspective comes from surveys administered after each run and live observations made during each run, augmented by video record. The operator must oversee the progress of the robot during execution of the selected route (intervening as necessary) and will likely strive for an understanding of the indirectly observed planning decisions made along the way; this presents new cognitive challenges. Quantitative measures for robot behavior would normally include the number and duration of operator interventions, the frequency for required re-planning using BIP, time to complete the route, and exposure time (when the exposure layer is activated); time aspects are not addressed here. In addition, we use observations made by data collectors and plot the differences between pre-planned routes and the alternative routes developed during each run.

B. Descriptive Measures

Simple descriptive statistics appear in the following tables. We recognize they are at best rough and indirect measures of performance but are worth reporting for completeness. Table 1 reports the outcome of each run performed in Area B12. The baseline autonomous mobility (AM) perception runs both resulted in e-stops. BIP alone resulted in the normal end of mission "halt" message to the operator. FCI alone resulted in one normal halt and one end of mission in which the robot traveled far away from the intended final way point. With both BIP and FCI, three runs resulted in two halts and another end of mission where the robot was off course.

TABLE 1

RUN OUTCOME FREQUENCY BY EXPERIMENTAL CONDITION (AREA B12)

	Outcome			
Condition	Halt	Off Course	E-Stop	All
AM	0	0	2	2
BIP	2	0	0	2
FCI	1	1	0	2
BIP+FCI	2	1	0	3
All	5	2	2	9

Table 2 reports the outcomes for conditions run in Area A1, focusing on the mobility benefits of BIP and the FCI. Three runs were aborted due to teleoperation camera failure. The FCI weights were varied, with the larger values yielding more control to the perceptive layer (i.e. FCI=2 means that the weighting of the perceptive layer costs was twice that of the deliberative layer). Results are mixed; FCI alone results in five halts in six attempts, whereas the combined BIP and FCI result in only one halt in seven attempts, one abort and five e-stops. Most e-stops occurred when the XUV tried to cross the marsh, but some were called due to excessive wander of the robot.

TABLE 2

RUN OUTCOME FREQUENCY BY EXPERIMENTAL CONDITION (AREA A1)

	Outcome			
Condition	Halt	Abort	E-Stop	All
AM	2	1	2	5
BIP	0	1	1	2
FCI=1	2	0	0	2
FCI=2	1	0	1	2
FCI=4	2	0	0	2
BIP+FCI=1	0	1	2	3
BIP+FCI=2	1	0	1	2
BIP+FCI=4	0	0	2	2
All	8	3	9	20

Table 3 shows the remaining conditions that were run and corresponding results. Toward the end of the second week, these runs were attempted to explore the impact of additional layers being considered in the deliberative planning. Mobility and FCI weights were also adjusted based on observation from earlier in the week. The weights for the Mobility, Time, and Exposure layers had a possible value of 0 to 1. Five of eight runs resulted in a normal halt. Two runs resulted in an e-stop and one in an abort.

TABLE 3

RUN OUTCOME FREQUENCY AND DELIBERATIVE LAYER WEIGHTS (AREA A1)

	Outcome			
Condition	Abort	E-Stop	Halt	All
BIP+Mob=0.5	1	0	0	1
BIP+Mob=0.5+Exp=1	0	0	1	1
BIP+Mob=0.5+FCI=2	0	1	0	1
BIP+Mob=0.5+Exp=1+ FCI=2	0	0	1	1
BIP+Time=1	0	0	1	1
BIP+Time=1+FCI=4	0	0	1	1
Mob=0.5+FCI=2	0	0	1	1
Mob=0.5+FCI=1+ Exp=1	0	1	0	1
All	1	2	5	8

Table 4 lists the various measures collected during the runs at Area B12. The FCI condition results in fewer back-ups, because the effect of the FCI routed the XUV, unintentionally, away from the cul-de-sac. Teleoperation repositioning was an important measure at Area A1, because it was used after a re-plan that was based on BIP. Repositioning was performed to orient the robot along the re-planned path. A similar summary for Area A1 was produced but is not presented.

TABLE 4

EVENT FREQUENCIES (AREA B12)

	Conditions			
	AM	BIP	FCI	BIP+FCI
Runs	2	2	2	3
Teleop_Obstacle	1	0	1	0
Teleop_Reposition	0	0	0	0
Back-up 5m	9	7	3	6
Back-up 10m	3	4	2	1
Back-up 15m	0	2	2	1
Back-up Total	12	13	7	8
Oper_Max BUs	0	2	1	0
Oper_Off Course	1	1	3	5
Resume_Only	0	1	4	5
Backtrack_15m	0	2	0	0
BU_Stuck	1	0	0	0

C. Path Analysis

Plots of the route traveled were made for each of the runs in the study. The plot, together with summary statistics and narratives collected during the run allow us to interpret events along the run. Figure 3 shows the outcome for a run in Area B12. The yellow line represents the original route plan to visit a second way point prior to traveling to the end way point at the helicopter pad. The global map feature data included for route planning is dated by several years and inconsistent with current vegetation. The red line indicates the path of the robot. The XUV traveled into the high brush near the second waypoint. Several backups were executed by the ANS in the vegetated cul-de-sac. A first re-plan based on BIP appears in blue. Subsequent attempts by the robot still failed to find a path. The operator teleoperated the XUV away from the trees at the end of the cul-de-sac and executed a new re-plan (orange) based on the updated terrain feature data in the global map. This route led the XUV successfully to the goal. Figure 1 in Section II provides a second example of the impact of BIP in which the run required three re-plans (blue, orange, and green) but no teleoperations to successfully reach the objective.

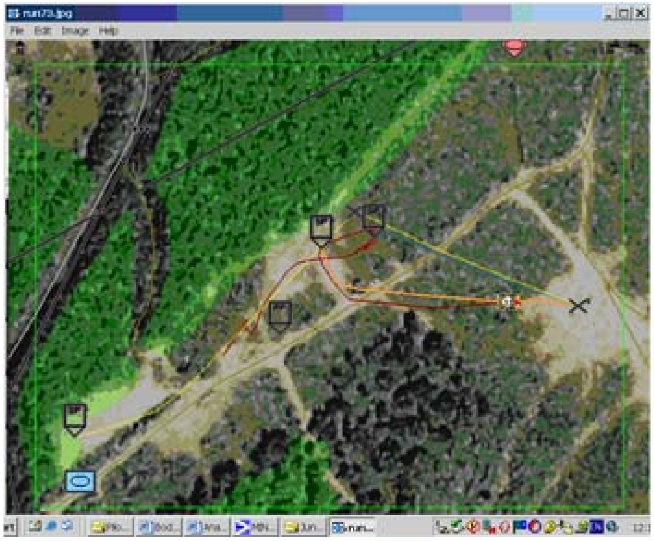


Fig. 3. Area B12 (Mobility =1, BIP)

Area A1 produced several interesting examples. Figure 4 shows one run in which the mobility and exposure deliberative layers were turned on along with BIP and the FCI. The positions from which to limit exposure were to the Southeast (North is up) of the operator control unit (OCU) position, denoted by the blue symbol in the lower right hand corner of the image, and to the Southeast of the final waypoint. The yellow line represents the original plan, passing through an area with dense woods that were intentionally not included on the global map. An early off-course message resulted in the first re-plan (blue); little change in path occurred. Both show the interest in achieving the second way point before retreating to the tree line in consideration of exposure. Although a definitive reason for the XUV traveling wide of that second re-plan before turning to the second way-point is not possible, a plausible reason may be that the deliberative planning layer was attempting to use the tree line along the South edge (not visible on the figure) or subtle changes in elevation to reduce the silhouette of the robot. Past the second way point, another off-course was issued because of the distance between the actual and planned routes. A re-plan (orange) provided a path to the last two way points. After passing through a gap in the trees and progressing through a meadow, the robot traveled to a location, which often produced an off-course message. At that point a final re-plan (green) was provided to guide the XUV to the final way point.

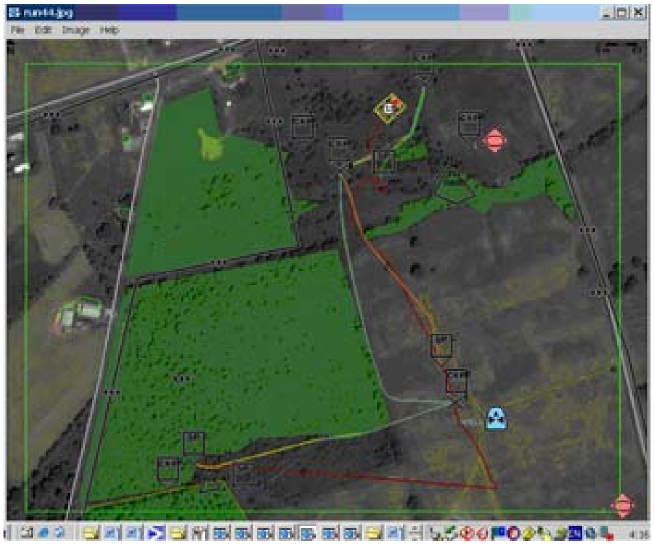


Fig. 4. Area A1 (BIP+Mobility=0.5+FCI=2+Exposure=1)

Figure 5 shows a situation with the deliberative layers for mobility and exposure turned on but only BIP operating during the run. This run also shows three re-plans. During the run, the robot does not appear to wander to the extent apparent in the run depicted in Figure 4.

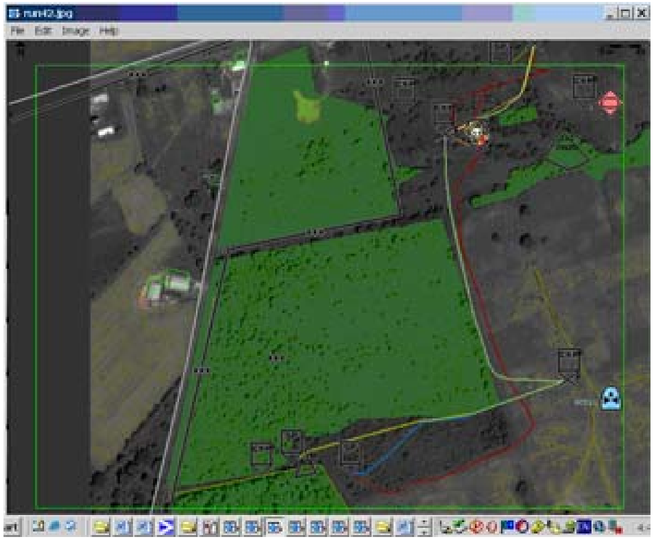


Fig. 5. Area A1 (BIP+Mobility=0.5+Exposure=1)

Not all runs were successful, clearly since we report a total of 11 e-stops in this area. Figure 6 illustrates one such run where the XUV appears lost after 6 re-plan attempts. An e-stop was called for safety reasons when the robot came too close to the OCU position. Figure 7 shows a typical path leading to an e-stop due to the XUV proceeding down a cul-de-sac to an impassable marsh. Actually, this particular run had to be aborted due to inclement weather.

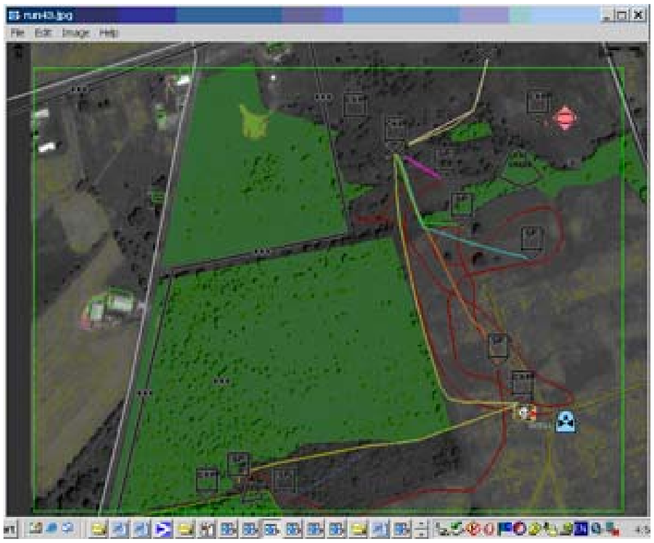


Fig. 6. Area A1 (BIP+Mobility=0.5+FCI=2)

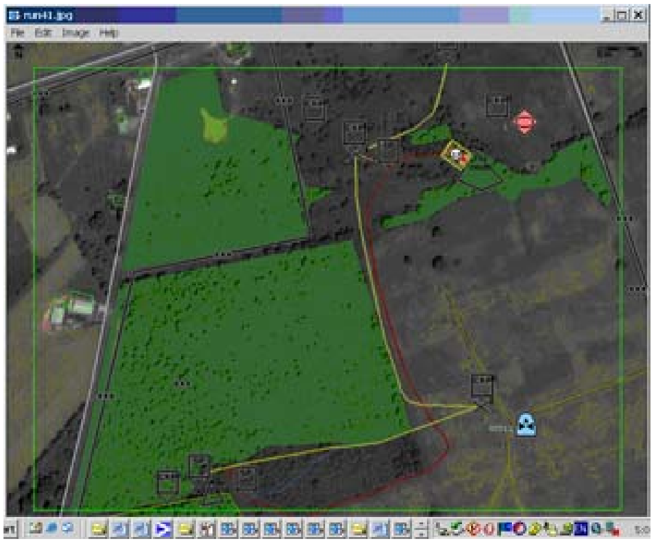


Fig. 7. Area A1 (BIP+Mobility=0.5)

Additionally, two real time displays (not shown) aid interpretation. One shows how the terrain around the robot is being assessed by the dynamic planner in terms of safe progression and another illustrates how the global map is being updated with local terrain features that the robot encounters during route traversal.

D. Operator Workload

Although the pilot study focused primarily on the integration of the bi-directional flow of information into UGV operation, the potential impacts on operator performance were considered as well. Three GDRS software developers (co-authors of this paper) acted as operators of the XUV during all the pilot study runs. Certainly these operators were not intended as representative of Soldier users; however, it was decided that useful operator feedback on workload and situation awareness (SA), as well as methods to measure the workload and SA, could be obtained from the pilot test operators during this early integration assessment.

Operator tasks included set-up for each run, execution of initial route plan, monitoring of XUV status, and intervention where required. There were some required teleoperation interventions. There were also re-plans during runs that implemented BIP. In these cases, the recalculation of routes was automated using the bi-directional flow of data, however, the call for, and execution of, the new plan were operator tasks (the objective is for this to be an automated process in the future). The operators were asked to make ratings of their overall workload (on a scale from 0-10), adapted from the Overall Workload Scale in [2] and [3]. Ratings (on scales of 1-7) of situation awareness (SA) were obtained for the three areas of 1) ability to perceive information, 2) ability to understand information, and 3) ability to predict what would happen. These situation awareness questions were drawn from the definition of SA in [4]. A final question on expectations ("Did the XUV do what you expected...?") was asked, also. In addition to the subjective ratings, video recordings of the OCU, over the shoulder of the operator, were also obtained.

Ratings from 26 trials were collected across all operators. In general, workload ratings were relatively low (mean= 3 (out of 10)) and situation awareness ratings were relatively high (mean=6 (out of 7)). When asked if the XUV behaved as expected, operators responded "yes" for twenty of the 26 trials. It should be noted, however, that some of these "yes" responses had qualifiers attached. For example, five of the "yes" responses also said something like "except for this one part of the run which was unexpected." Interestingly, one operator said that the XUV behaved as expected because "I had different expectations based on what it did for the last [similar] runs." Based on the ratings and additional responses, then, it seems as if the robot behavior at times was unexpected, puzzling the operators, and some expectations were changed based on observed behaviors. This question of expectations and changing expectations needs to be explored further for its implications on information and decision support for operators.

The issues that are highlighted here include the ability of the operator to perceive and understand information relevant to intelligent behavior by the robot, and then know what will happen in the future (as shown by prediction and expectation). What does the operator need to know? How involved does the operator need to be (or want to be) in planning decisions made automatically by the robot; when is permission to execute and proceed needed? Issues of trust in automation and complacency arise. Workload associated with these tasks, for single and multiple robots, as well as all other operational tasks being performed, are important to consider.

IV. CONCLUSION

As a result of the June 2007 exercise, the developers learned a great deal about the technology performance, necessitating changes for later releases. The Army sponsors recognized many issues to be addressed in further testing. With regard to changes, the FCI could benefit from an investigation of the balance of deliberative planning layers and perceptive planning. It was suspected that weighting played a role in some of the unexpected behaviors observed. A desirable change would enable, at the end of the mission, the field cost planner to yield to the original planner so that the end way point can be more consistently achieved. Some runs, especially in Area B12, resulted in end of missions being called when the XUV was far removed from its destination. Further, improved tracking is sought to keep synchronized the robot and the OCU. It is suspected that the run illustrated in Figure 6 was probably due to the instructions coming to the robot at an inappropriate time for where the robot actually was at that time. Improved handling of off-course messages is also being worked; during the experiment, this message resulted in "resume mission" or a "re-plan" and execute.

Several improvements related to the BIP are also ongoing. The range of local sensing is being increased from 20 m to 60 m. This should increase the potential path options seen by BIP. A planned improvement is to increase the scan sweep upon robot back-up and when the robot slows, the latter taking advantage of the opportunity not to compete with processing that supports the ANS. During the June 2007 exercise, we observed instances when the XUV missed opportune paths in very close proximity, most likely because the scan was not wide enough to see them. When the opportunity called for using BIP to suggest a re-plan for execution, current protocol requires the operator to reorient the robot to the new route; this process is being automated. Efficiencies in sensed data logging are also being pursued.

With regard to the design of the study, several issues were recognized. The test protocol must be sufficiently robust to handle the new situations created by enhanced robot behaviors. Data consistency depends on a tight, strictly adhered to protocol. Communications continue to present a challenge; to work this problem more time is warranted prior to the experiment to evaluate base station locations. Scripts will be developed to automatically set configurations for each run. In the present study, this work was done anew each time, sometimes at both the OCU and at the XUV, taking more time and introducing more opportunity for set-up error.

The terrain and mission context of runs must also be revisited. In the present study, we relied on one classic case, the cul-de-sac, to exercise the new technology. Other interesting cases must be determined to highlight "problem solving" over an array of challenges. Further, the terrain must be expanded and more varied for the coming experiment. Elevation data, for example, changed very little in either area, minimizing the technology's ability to leverage it in consideration of, for example, an exposure layer. And by restricting the length of the run in the present exercise (run lengths for Area B12 and Area A1 were less than 1 kilometer), we limited the FCI in leveraging mobility and time layers as well. An expanded area is available for testing in the fall. Finally, to provide a richer environment for workload assessment and to experiment with a maturing Reconnaissance Surveillance, and Target Acquisition (RSTA) capability, RSTA mission elements should be rolled into the subsequent experiment.

The glaring issue remains of how to measure the success or failure of this intelligent system. Our approach is merely to identify elemental challenges for the system to overcome and then to determine whether or not they were overcome. But the question of exactly how to determine success and to what degree remains elusive. In a large scale study, changes in the frequency of e-stops, teleoperation, etc. would be revealing. Elements of time to complete the mission and the duration required to overcome a course obstacle would also serve as a basis for comparison. A decision tree developed post hoc to be evaluated using subjective utilities from a Soldier scout has been considered. The authors welcome input on measurement that could be helpful in subsequent evaluations.

REFERENCES

[1] R. Camden, B. Bodt, S. Schipani, J. Bornstein, T. Runyon, F. French, C. Shoemaker, A. Jacoff, A. Lytle, "Autonomous Mobility Technology Assessment Final Report," ARL-TR-3471, 2005.

[2] M. Vidulich and P. Tsang, "Absolute Magnitude Estimation and Relative Judgment Approaches to Subjective Workload Assessment," in Proceedings of the Human Factors Society 31st Annual Meeting, 1987, pp. 1057-1061.

[3] S. Hill, H. Iavecchia, J. Byers, A. Bittner, A. Zaklad, and R. Christ, "Comparison of Four Subjective Workload Rating Scales," *Human Factors*, 34(4), pp. 429-439, 1992.

[4] M. Endsley, "Design and Evaluation for Situation Awareness Enhancement," in Proceedings of the Human Factors society 32nd Annual Meeting, 1988, pp. 97-101.

[5] M. Endsley, "Towards a Theory of Situational Awareness in Dynamic Systems," *Human Factors* 37(1), pp. 32-64, 1995.

A Common Operator Control Unit Color Scheme for Mobile Robots

M. Shneier, R. Bostelman, J. S. Albus, W. Shackleford, T. Chang, T. Hong
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD

Abstract

The Intelligent Systems Division at the National Institute of Standards and Technology (NIST) has participated in the Defense Advanced Research Project Agency (DARPA) Learning Applied to Ground Robots (LAGR) project for the past 2 ½ years. In Phase 2 of the LAGR program, NIST was asked to provide a common operator control unit (OCU) color scheme for all LAGR teams to use. The color scheme simplifies the task of LAGR's evaluation team by providing a straightforward way to compare the performance of each of the teams using the different OCUs. During Phase 1, LAGR performers applied their own standards to the OCU color scheme and DARPA and other performers had a very difficult experience evaluating what the robot was computing based on stereo image, instrumented bumper, and inertial data.

NIST developed the color scheme based on real-world conventions and on the desire to accommodate as much of the teams' existing color schemes as possible. For example, typically red lights mean stop and green lights mean go for automobiles. This scheme was adopted by coloring obstacles red and traversable ground green in the new common color scheme. Red, green, blue (RGB) colors were produced for a variety of necessary parameters including: unknown regions, lethals, bumper hits, road/path, planned and traversed paths, goal, and waypoints. Also, vehicle modes were expressed such as: Normal Control, Aggressive, Backing, Stopped, and Manual modes.

The paper discusses the color scheme for ground robots developed for the LAGR Program.

I Introduction

The Operator Control Unit (OCU) for a mobile robot needs to display a lot of complex information about the state and planned actions of the vehicle. This includes displays from the robot's sensors, maps of what it knows about the world around it, traces of the path it has already traveled and predictions of the path it is planning to take, and information about obstacles, clear ground, and unseen regions. The information needs to be easy to understand even by people who have no understanding of the way the control system of the robot works, and should enable them to halt the vehicle only if it is about to take an action that will cause damage.

In order to display all the information in an understandable way, it is necessary to use color to represent the different types of region. There are no existing conventions on what colors to use and each OCU developer must decide how best to assign the colors. This is fine if the color scheme chosen is indeed easy to understand and if the people using the OCU are only dealing with a single robot. However, there is a growing need for the ability to control more than one robot, especially in the areas of urban search and rescue and bomb disposal robots, where a damaged or destroyed robot needs to be replaced as soon as possible.

There has been at least one effort to create a multi-robot OCU[1], called MOCU. The goal is to create an OCU that can control multiple vehicles of different types and from different manufacturers from a single OCU. The OCU is based on a core set of capabilities enhanced by modules that provide specific capabilities and communication protocols needed by each vehicle. The capabilities are defined at run time through configuration files. An impressive set of vehicles can be monitored simultaneously using MOCU. Unfortunately for the purposes of this paper, no standard color scheme is described for MOCU, so it was not possible to benefit from this approach.

Another paper does describe the color scheme used in its multi-vehicle OCU [2]. They describe the following assignments of colors, some of which are similar to the choices discussed below. The display a two-dimensional scrollable map of the search area marked with icons to indicate the relative positions of the robots, obstacles, target munitions and terrain features. Icons for robots change color to indicate the current status of the robot: green indicates operation within normal limits and red indicates a fault or a lack of progress. In the work described in this paper, multiple robot modes are described with different colors. Robot icons also indicate the heading of the robot and its position relative to domain objects and other robots. An obstacle detected by the robot or manually entered by the operator is displayed as a gray box. Areas that are indicated clear by the robot's IR sensors can optionally be colored blue on the operator's display, to distinguish those areas from unswept areas. If the robot detects steep parts of the terrain, the OCU can mark those areas yellow or orange, depending on the sensed inclination. As the robot moves, the clear areas under the munitions detector are shown in green, similar to what is used in our OCU. When a robot detects unexploded ordinance, its location is marked with a red circle on the OCU display. In the OCU described here, critical obstacles are also displayed in red.

The OCU developed in this work was oriented towards a somewhat different purpose. As part of the DARPA LAGR program [3], a large number of teams were tasked with developing learning algorithms to try to improve their performance in a series of field trial, held once a month. After the first phase of the program, NIST was asked to develop a common color scheme for the OCUs used by each team so that the evaluation team could reduce the need to learn the meaning of colors used by each team.

II The DARPA LAGR Program

The DARPA LAGR program [4] aims to develop algorithms that will enable a robotic vehicle to travel through complex terrain without having to rely on hand-tuned algorithms that only apply in limited environments. The goal is to enable the control system of the vehicle to learn which areas are traversable and how to avoid areas that are impassable or that limit the mobility of the vehicle. To accomplish this goal, the program provided small robotic vehicles to each of the participants (Figure 1). The vehicles are used by the teams to develop software. A separate LAGR Government Team, with an identical vehicle, conducts tests of the software each month.

The vehicle provided by DARPA is a small but very capable robot with substantial on-board processing capacity and a rich set of sensors. The sensors include two pairs of color cameras mounted on a turret on the front of the vehicle, a pair of infra-red range sensors (non-contact bumpers) on the front of the vehicle, and a physical bumper centered on the front wheels of the vehicle. For position sensing, the vehicle has a Global Positioning System (GPS) receiver, wheel encoders, and an inertial navigation system (INS). In addition, there are sensors for motor current, battery level, and temperature. There are four single-board computers on the vehicle, one for low-level vehicle control, one for each of the stereo camera pairs, and one for overall control of the vehicle. All processors use the Linux operating system. The vehicle has an internal Ethernet network connecting the processors, and a wireless Ethernet link to external processors.

The availability of range information from stereo vision enables the robot to navigate largely using the geometry of the scene. Sensor processing is aimed at determining where the vehicle is and what parts of the world around it are traversable. The robot can then plan a path over the traversable region to get to its goal.

When the vehicles were delivered, they came with a baseline control system and a baseline OCU. While some of the teams stuck with the baseline OCU, many developed their own OCU to provide better information for debugging and monitoring the robot. Since there were no standards for how the OCU should look, each of the teams developed dissimilar appearance models. As a result, when the Government evaluation team ran the monthly tests, they had to learn the conventions used by each OCU in order to monitor performance of the associated vehicle controller. This led them to request that NIST develop a common color scheme for Phase II of the program.

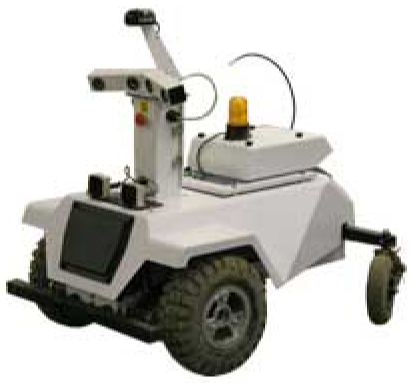


Figure 1. The robot used in the DARPA LAGR program.

III The Color Scheme

There is a large amount of literature on color and how to select color schemes for human-computer interfaces. Reference [5] provides a brief overview and links to more comprehensive resources. Many of these approaches make use of the color wheel and recommend selecting complementary or analogous colors. There are many tools to help in this selection [6]. This approach breaks down, however, when more than a small number of colors have to be selected and when certain colors have an accepted meaning in the application.

The way the color scheme for the LAGR OCUs was developed was to start out with a straw man proposal, which was sent out to the teams and to the DARPA Program Manager. Comments were received from a number of teams, which resulted in a revised scheme. This process was iterated until a scheme acceptable to all was developed. This may not be the best way to assign colors, but the intention was not to find the most pleasing scheme, but one that would be easy to interpret. Note that the color scheme was not intended to change the individual teams' OCU layout or content, only to ensure that the appearance of similar information in different OCUs would be consistent in meaning.

For the OCU, a large amount of information had to be represented simultaneously. Most of the teams displayed similar information, but allowance had to be made for extra features if they were required by even one of the teams. The information to be displayed included maps, which often included a low resolution long-range map and a higher resolution close-up map. These displays contain most of the information the color scheme needs to represent. Also displayed is vehicle mode (normal, aggressive, backing up, etc.). A third set of displays includes one or more images of what the vehicle is currently seeing. While there are often overlays on the images, so far the colors for these have not been standardized.

Maps need to display a wide range of information about the terrain and the planned and traversed paths. The colors assigned to the various features are shown in Figure 2 and 3. The same color scheme is used for both the low resolution and high resolution maps. Most colors are fixed, and refer to a single type of feature, but traversability ranges across a green-to-black spectrum based on how expensive it is to cover the associated terrain. Green means the vehicle can easily drive, while black means it is very difficult to drive. Note that in the Hue, Saturation, Luminance (HSL) color space, this requires changing only the Luminance component of the color, making for an easy mapping from cost to color. These costs, in conjunction with obstacles and bumper hits, are used by the planners to determine the optimal path to the goal. Comparing our scheme for representing steep slopes with that of [2], they have two fixed colors, orange and yellow for different degrees of steepness, while our scheme would assign smoothly-varying colors starting as green at the base of the slope and becoming darker as the slope gets steeper. This, we believe, gives more information about the true nature of the traversability of the slope.

🟩→⬛	Traversability cost ranges from low = green (0,255,0) to high = black (0, 0, 0)
🟦	Unknown regions = blue (0, 0, 255
🟥	Lethal obstacles = red (255, 0, 0)
🟥	Bumper hits = dark red (170, 0, 0)
⬜	Road/path = light gray (215, 215, 215)
🟨	High-level planned path = yellow (255,255,0)
🟧	Low-level planned path = orange (255, 150, 0)
🟪	Traversed path = dashed purple (255, 0, 255)
⬜	Goal direction = white (255, 255, 255)
□	Goal = white square (255, 255, 255)
🟨	Waypoints = yellow squares on high level path (255,255,0)
▽	Camera FOV border = white (255, 255, 255)

Figure 2. The map colors and their interpretations.

The vehicle itself is displayed on the map, together with its field of view (FOV), planned path, and the path traversed so far. The planned path includes waypoints and an indication of the location of the goal. The straight-line path to the goal is also indicated. If the vehicle controller determines that it is traversing a road or path, this is displayed in gray.

The display of the vehicle changes color to indicate the current mode (Figure 4). Modes represent the status of the vehicle. Not all teams make use of all the modes, but simple ones, such as normal autonomous driving, stopped, and backing up are universal.

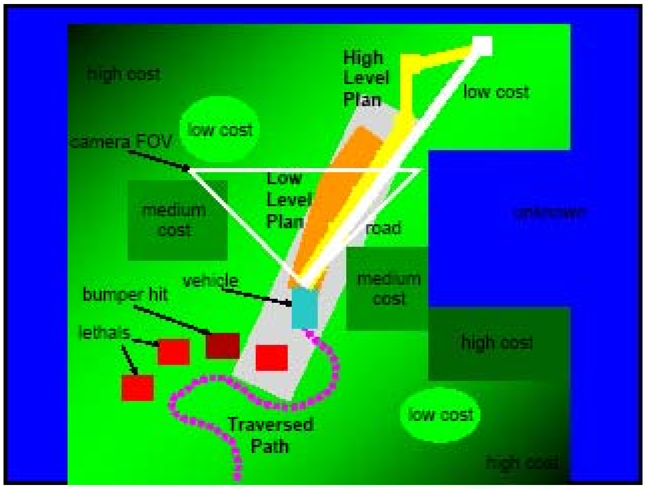


Figure 3. A schematic showing the meaning of the colors on the map.

🟦	Normal Control = aqua (40, 200, 200)
🟪	Aggressive = dark purple (130, 0, 130)
🟪	Backing up = purple (255, 0, 255)
🟫	Stopped = brown (150, 100, 50)
🟦	Manual = light blue (120, 120, 255)

Figure 4. Vehicle modes displayed by coloring the vehicle on the map.

IV Implementation by LAGR Teams

Each of the LAGR teams was required to implement the scheme by the February 2007 test date. The common OCU Color Scheme code was sent to all teams in December, 2006. Implementation was only partially achieved, in that some teams did not have their OCUs ready in time, and some implemented only some aspects of the color scheme. NIST implemented the scheme (Figure 5 and Figure 6), showing both a high-resolution near-range map and a low-resolution, longer range map. In these figures, the vehicle believes that it is traversing a path, so the planned trajectory remains on the path until it ends and the vehicle has to enter unknown terrain to reach the goal.

SRI International originally developed their own color scheme, shown in Figure 7, which included only some of the new scheme features shown in Figure 2. There is no representation of the vehicle, and no waypoints, and the traversable terrain is shown in a single color. While the items represented have not changed, SRI implemented the full color scheme (Figure 8, by permission of SRI) so that the terrain is now shown with variation in traversal cost, obstacles are in red, and unknown regions in blue.

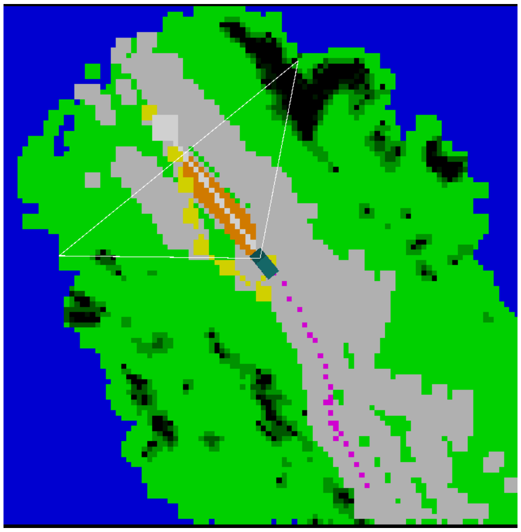


Figure 5. The NIST high resolution, short-range map.

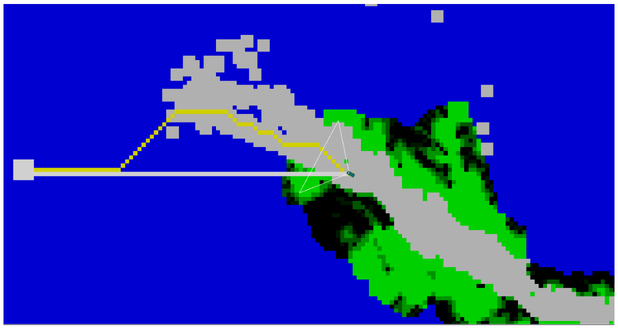


Figure 6. The NIST low resolution, long-range map.

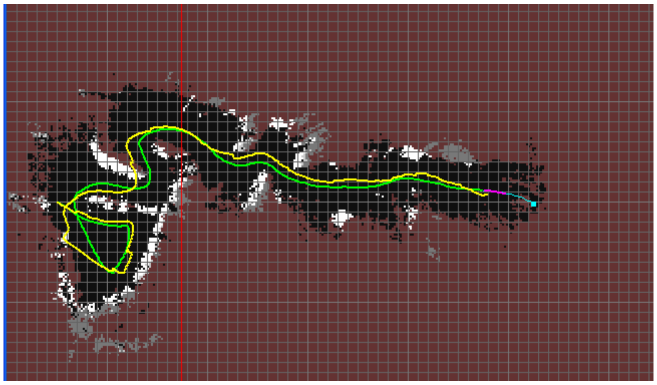
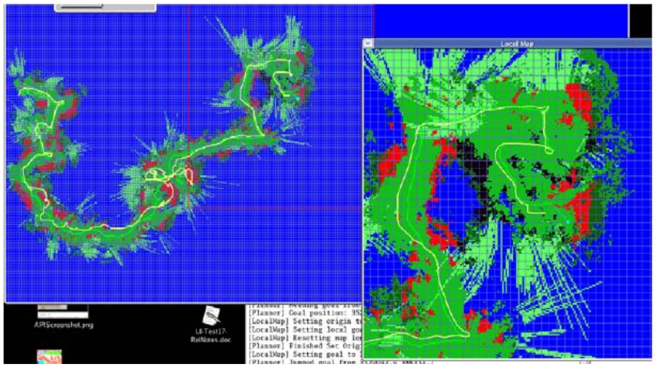


Figure 7. The old SRI OCU color scheme.

Another of the teams, Netscale Technologies, Inc., converted from their old, rather attractive, color scheme shown in Figure 9 for the high-level (low resolution) map and Figure 10 for the high resolution map. Their implementation of the common color scheme is shown in Figure 11 (low resolution) and Figure 12 (high resolution). Note that Netscale made other changes to what is represented in the OCU, so the maps are not directly comparable.

An interesting variant of the color scheme was implemented by the University of Pennsylvania. While it uses similar colors to the common color scheme, it uses pastel versions. For their scheme, the color correspondence to the map cost from low to high is shown in the color bar on the right of Figure 13 (by permission of U. Penn.). Green means safe to drive over, red marks obstacles, blue areas are unknown, and white indicates the planned path. This is in contrast to their old color scheme, shown in Figure 14. That color scheme was based on shades of gray. The higher the map cost, the darker the grayscale, and the lower the map cost, the whiter the color. In this scheme, white meant safe to drive over, dark gray (black) meant an obstacle, midgray was unknown, and blue indicated the planned path. Clearly, the new color scheme is closer to the standard, but it is not fully compliant.



```
[Planner] Goal position: 35
[LocalMap] Setting origin t
[LocalMap] Setting local go
[LocalMap] Resetting map lo
[Planner] Finished Set Orig
[LocalMap] Setting goal to
```

Figure 8. The new SRI OCU using the common color scheme.

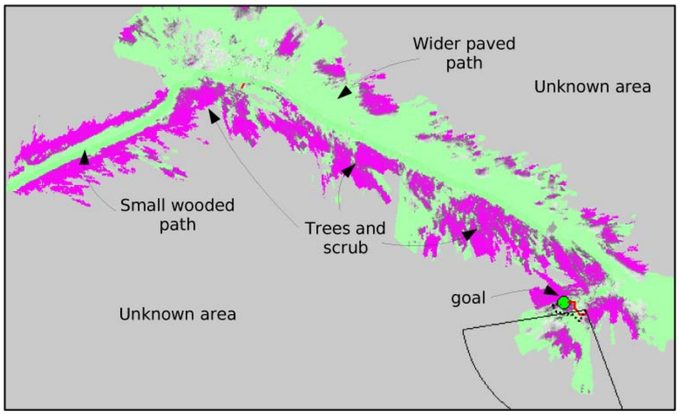


Figure 9. The old Netscale low resolution map color scheme (with permission from Netscale Technologies).

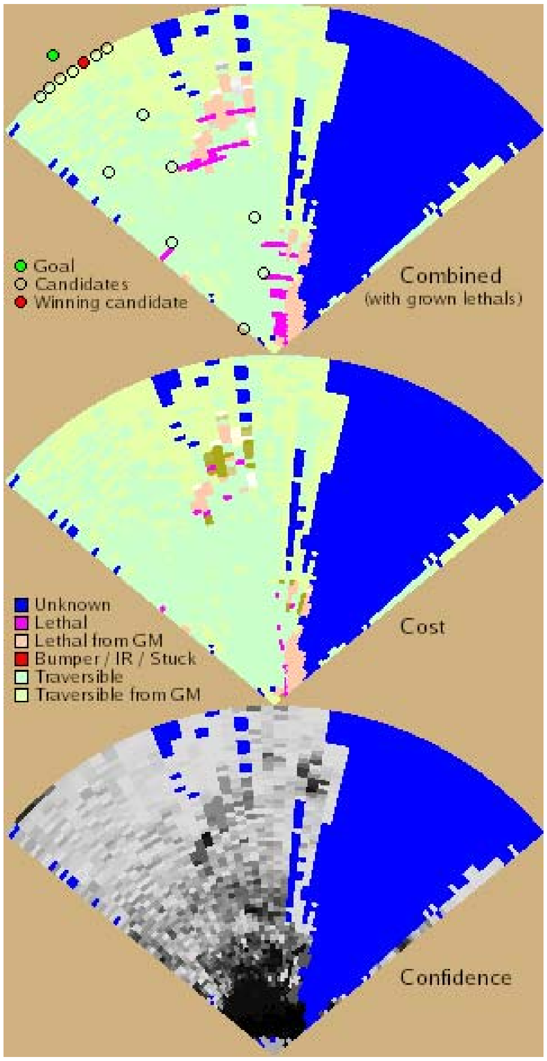


Figure 10. Old Netscale high-resolution maps.

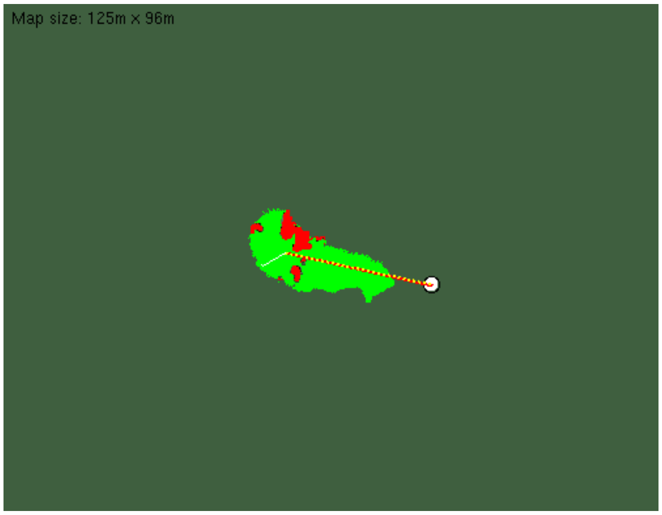


Figure 11. New Color Scheme for Netscale's low resolution maps.

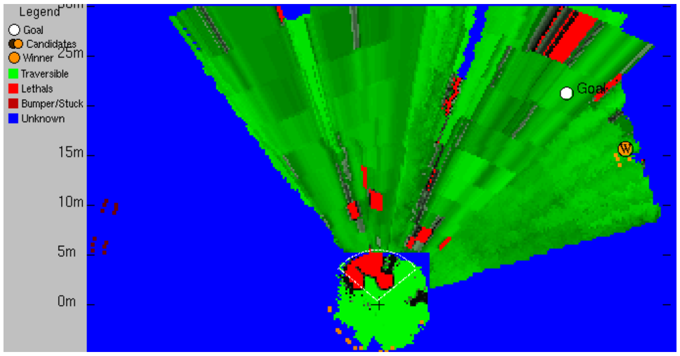


Figure 12. New color scheme applied to Netscale high resolution map.

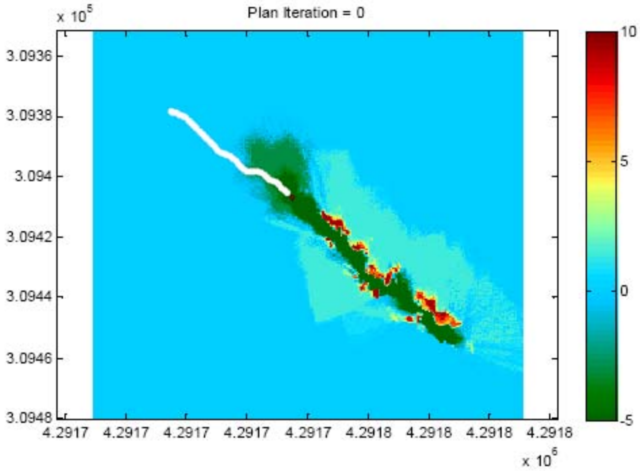


Figure 13. U. Penn's variant of the color scheme.

V Discussion and Conclusions

The color scheme was distributed to the teams in December, 2006. They were expected to try to have it in place by the January, 2007 test, and were mandated to have it in place by the February, 2007 test. While a few of the teams made the deadline, many did not. NIST provided a function to map cost and identification of a pixel or region to color, and now all teams have either adopted the color scheme as is, or, like U. Penn, make use of a variant based on the common color scheme.

Use of the color scheme has had the desired effect. The Government evaluation team has expressed satisfaction in the results. They can more easily understand the OCUs of different teams, although the fact that teams do not have to use a standardized OCU layout makes the benefit smaller than it could have been. There is a lot of resistance in the teams to changing their OCU displays. The displays were developed for debugging and for monitoring the vehicles' progress. Given the very large variation in approaches to the LAGR problems, it would be difficult and very time-consuming to come up with a universal OCU that captured all the possible information to be displayed and also supported the debugging approaches of all the teams.

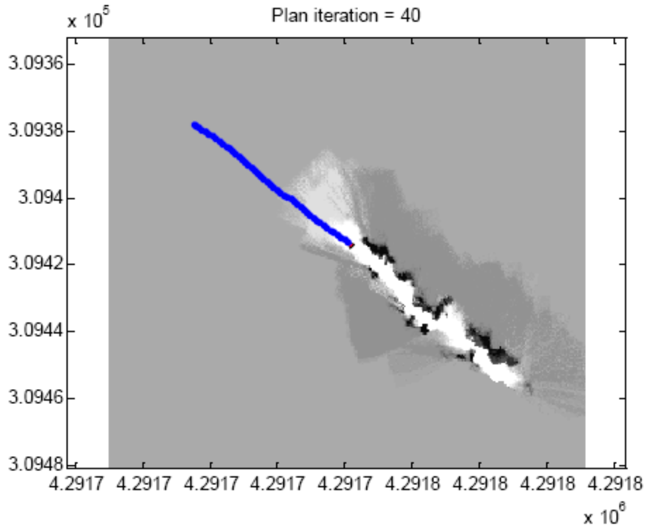


Figure 14. U. Penn's prior color scheme.

It is interesting that there are few, if any, standard color schemes for computer applications. Large vendors, such as Microsoft and Apple, develop and publish in-house styles for their products that cover layout, color, and menus, amongst other things. Typically, colors are chosen from "themes" or groups of compatible colors Thus, all Microsoft Office products have the same look and feel on a single machine. By choosing different themes, different users keep consistency between different tools, but the tools may appear different than those on another machine. This is not a standard, however, although other vendors are encouraged to adopt the color schemes. Other tools for accomplishing similar tasks often use different color schemes to differentiate their products and avoid potential legal issues.

There are many tools to help choose colors that go together [6], but they work primarily for small numbers of colors. For larger numbers, people sometimes use a "natural" color scheme in which a photograph of a natural scene is used to pick the colors. The assumption is that nature is harmonious, so the resulting color scheme will be as well. This approach works well, but doesn't suit our purposes. We have certain colors that by convention in the program have predefined meanings (e.g., red for obstacles). Some of the colors should stand out from the others, so not all the colors should be harmonious. We also had to accommodate strong feelings on the part of different teams about use of certain colors. All this led us to a consistent, understandable color scheme that breaks many of the rules for picking colors.

It was a useful, if somewhat tedious process to develop the common color scheme. Work needs to be done more broadly to develop standard color schemes for different application areas, such as medical images, geographic information systems, and other complex visual displays that require substantial effort to understand.

Bibliography

[1] D. N. Powell, G. Gilbreath, and M. H. Bruch, "Multi-robot operator control unit," SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium ed Orlando, FL: 2006.

[2] P. K. Pook, J. A. Frazier II, T. Ohm, R. Robert, and G. Whittinghill, "A Testbed for Evaluating Robot Team Interaction under Operator Supervision," Proceedings of UXO Forum '98, http://citeseer.ist.psu.edu/267793.html 1998.

[3] L. D. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The DARPA LAGR Program: Goals, Challenges, Methodology, and Phase I Results," *Journal of Field Robotics, Special Issue on Learning in Unstructured Environments*, vol. 23, no. 11/12, pp. 945-973, 2006.

[4] T. Wagner, "Learning Applied to Ground Robots (LAGR)," http://www.darpa.mil/ipto/programs/lagr/index.htm, 2005.

[5] Wikipedia, "Color Theory," http://en.wikipedia.org/wiki/Color_theory, 2007.

[6] P. Lyons and G. Moretti, "Nine Tools for Generating Harmonious Colour Schemes,", The 6th Asian Pacific Conference on Computer Human Interaction (APCHI 2004) ed Rotorua, New Zealand: 2004.

How DoD's TRA Process Could Be Applied To Intelligent Systems Development

D.A. Sparrow
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311, USA
dsparrow@ida.org

S. Cazares
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311, USA
scazares@ida.org

***Abstract*—Over the past several years, the Department of Defense (DoD) has instituted a Technology Readiness Assessment (TRA) process based on NASA's Technology Readiness Levels (TRLs). The motivation was to ensure that technology development was complete and that performance was understood before entering into the System Development and Demonstration (SDD) phase of a program. Such a disciplined approach may aid in Intelligent Systems development. However, NASA's TRLs were derived in a context of hardware systems, and the hardware TRLs needed modification to treat software and software-intensive systems. This paper will examine under what conditions additional modifications might be necessary to treat Intelligent Systems.**

Technology development can only be "complete" in the context of a specific program with known performance requirements. Thus, the TRA's focus on critical technology elements (CTEs)—those technologies used in a new or novel way that are essential to system performance. These CTEs are assessed for their performance in a relevant environment, as determined from a consideration of the system's requirements. For Intelligent Systems, this focus on CTE's and relevant environment may provide a disciplined approach to ensuring technology maturity before system development.

The algorithms that make decisions will often be the distinctive CTEs, unlike the CTEs of hardware systems. However, the major differences between Intelligent and hardware systems are likely to be in the "relevant environment". Intelligent Systems that develop and execute a course of action will, by their nature, present challenges in the definition of the "relevant environment". We will explore the effect of various degrees of "intelligence" on CTEs and the relevant environment in this paper.

I. Introduction

Over the past several years, the Department of Defense (DoD) has incorporated a Technology Readiness Assessment (TRA) process into the formal systems acquisition process. This is a data-driven, metrics-based approach that uses NASA's Technology Readiness Levels (TRLs). The motivation behind instituting this process was to ensure that technology development was complete and that performance was understood before entering into the System Development and Demonstration (SDD) phase of a program. Such a disciplined approach may aid in Intelligent Systems development. However, NASA's TRLs were derived in a context of hardware systems. The hardware TRLs needed modification to treat software and software-intensive systems. This paper examines under what conditions additional modifications might be necessary to treat Intelligent Systems, and even though no modifications were identified, some subtle differences between Intelligent and hardware systems were found.

One difference between Intelligent and hardware systems is in the nature of the "relevant environment". An Intelligent System with enough capacity for adaptation may be able to function in environments outside the design space. Intelligent Systems that develop and execute a course of action will, by their nature, operate in a variety of environments and require attention to a comprehensive set of possible environments early in development. A second major difference is the need for early identification of metrics to properly train an Intelligent System, since Intelligent Systems frequently require training to optimize the values of their many parameters for a given purpose. We find the need for early attention to environments and metrics to be distinguishing features of Intelligent Systems.

Technology development can only be "complete" in the context of a specific program with known performance requirements. Thus, the TRAs evaluate the readiness of Critical Technology Elements (CTEs)—those technologies that are used in a new or novel way and are essential to system performance. These CTEs are assessed for their performance in a relevant environment, as determined from a consideration of the system's requirements. For Intelligent Systems, this focus on CTEs and the metrics needed to assess performance in a relevant environment may provide a disciplined approach to ensuring technology maturity before system development.

II. The Technology Readiness Assessment (TRA)

A. Motivation

The purpose of a TRA is to ensure that programs entering the SDD phase of development are based upon mature technologies. Experience has shown that programs that enter SDD with immature technologies are frequently plagued with additional and, hence, expensive design cycles [1]. Current DoD regulations require a TRA before Milestone B or Milestone C approval (Key Decision Points B and C for space programs). These are the decision points for beginning SDD and Low Rate Initial Production (LRIP), respectively. For ships, a TRA is also required at program initiation [2].

B. The TRA: Critical Technology Elements (CTEs)

A TRA is a metrics-based process that assesses the maturity of the CTEs of a system design. The CTEs must be

- Essential for system performance and
- New or novel technologies or technologies used in a new or novel way.

An example here may be instructive. The Crusader Self-Propelled Howitzer system was envisioned to have a much faster rate of fire and much improved cross-country mobility than those of earlier systems. The faster rate of fire was a capability that depended upon CTEs. This capability required use of an autoloader, which, in turn, required a propellant that could be handled by a machine. Two candidates emerged: (1) A new liquid propellant that could be regeneratively pumped into the combustion chamber (i.e., pumped using the pressure in the combustion chamber rather than an external force) and (2) an established solid propellant, packaged in a combustible case form that could be handled by a machine. The first candidate was clearly a new or novel technology. The second candidate was an established technology used in a new or novel way. Both are examples of CTEs, and the maturity of these CTEs was essential for proceeding forward with the Howitzer development program. In contrast, cross-country mobility was a capability that did not depend upon CTEs. Cross-country mobility for a vehicle of this weight class required tracks, pads, shoes, and the like. However, since there was nothing new or novel in this aspect of the mobility system, none of these components qualified as CTEs.

C. Technology Readiness Levels (TRLs)

The CTEs, once identified, are assessed against a set of predefined metrics as part of the TRA process. The DoD has adopted NASA's TRLs as metrics for assessing the maturity of hardware-based systems:

1. Basic principles observed and reported
2. Technology concept and/or application formulated
3. Analytical and experimental critical function and/or characteristic proof of concept
4. Component and/or breadboard validation in a laboratory environment
5. Component and/or breadboard validation in a relevant environment
6. System/subsystem model or prototype demonstration in a relevant environment
7. System prototype demonstration in an operational environment
8. Actual system completed and qualified through test and demonstration
9. Actual system proven through successful mission operations.

For software systems, some changes in TRLs 4–7 were required, reflecting the differences in subsystems and software modules and the need for end-to-end demonstrations of software systems. The software TRLs used by the DoD are

1. Basic principles observed and reported
2. Technology concept and/or application formulated
3. Analytical and experimental critical function and/or characteristic proof of concept
4. Module and/or subsystem validation in a laboratory environment
5. Module and/or subsystem validation in a relevant environment
6. Module and/or subsystem validation in a relevant end-to-end environment
7. System prototype demonstration in an operational high fidelity environment
8. Actual system completed and qualified through test and demonstration in an operational environment
9. Actual system proven through successful mission-proven operational capabilities.

TRLs 1–3 apply to invention through proof of concept (PoC) and are clearly in the realm of basic science and technology (S&T). Achieving TRL 4 depends upon component or module development, usually with some application in mind. TRLs 5–7 exist in the context of a program with requirements. Requirements are a prerequisite for defining the relevant and operational environments in which the CTEs are envisioned to perform. Requirements are also needed to determine some minimum threshold above which the performance of a technology is deemed acceptable in a relevant or operational environment. Finally, requirements are frequently needed to determine whether the application of a technology is new or novel.

At this stage, in addition to formal requirements, the expert judgment of the technical community is needed for a variety of reasons. First, expertise is needed in identifying CTEs. In addition, expertise is needed in deducing the relevant environment from the operational environment. Although satisfactory performance is usually demonstrated earlier in a relevant environment (at TRL 6) than in an operational environment (at TRL 7), the operational environment is typically defined first, with the definition of a relevant environment derived thereafter. Expert technical judgment is key to this derivation.

Considering satellite systems at this point is instructive. One does not want to have to launch a satellite into the space environment in order to proceed with the design and demonstration of the satellite. The relevant environment for the technologies on the satellite will depend upon what aspect of the environment (e.g., thermal load, the radiation environment of space, or g-forces during launch) is causing stress in the satellite. Any of these stressors and the effects of the stressors upon the satellite can be readily tested and demonstrated in the lab. In general, technical expertise is needed to ensure that while the performance of a system's CTEs is demonstrated in the face of environmental stressors, an expensive, exhaustive demonstration program is not applied to noncritical elements.

Despite the usefulness of the TRA process, an important point to note is that TRAs are confined in scope. They are not risk assessments, design reviews, or a method to address system integration. Yet, a well-performed TRA should result in the use of mature technologies, which, in turn, should reduce risk and enable efficient system integration.

D. Regulatory Aspects

The TRLs were developed and first used by NASA in the 1970s. In the late 1990s and early 2000s, DoD became interested in TRLs as a means of managing the tendency for programs to enter SDD with immature technologies. Use of TRLs as part of a formal TRA was first required in 2003. TRL 6 was the standard for Milestone B (entry into SDD), and TRL 7 was the standard for LRIP. In 2006, a statutory requirement was enacted for certification that "the technology in the program has been demonstrated in a relevant environment" prior to Milestone B. This language is taken from the definition of TRL 6.

The Government Accountability Office (GAO) has also begun to look at Department of Energy (DOE) projects from a technology readiness perspective [3]. DOE has responded by reassessing their own processes in this area. One can reasonably expect increased attention to technology readiness in many areas. For instance, any fielding of Intelligent Systems is likely to trigger regulatory interest. The Federal Aviation Administration (FAA) has already issued rulings on unmanned air vehicles in commercial airspace, even though these unmanned systems have little onboard intelligence or autonomy. As Dr. Zelinsky points out in his abstract "People expect autonomous technologies to operate at higher levels of performance and safety than people themselves exhibit" [4].

E. Summary

The purpose of the DoD's TRA process is to ensure that all new or novel technologies essential for meeting system requirements have been identified and that the performance of these technologies has been demonstrated in the appropriate environment(s). The motivation behind the TRA process is to prevent the extended deadlines and high costs that immature technologies often cause in SDD. As discussed in the next section, applying the TRA process to Intelligent Systems will present some challenges. However, development of Intelligent Systems will also benefit from the TRA process.

III. Intelligent Systems

A. Definitions

To consider what may distinguish Intelligent Systems from other "ordinary" hardware or software systems in the DoD acquisition process, we will use the following working definition: *An Intelligent System is a system that makes complex decisions or recommendations in a complex environment in place of a human.* The system may use data provided externally or obtained by the Intelligent System itself. The decisions *may* result in actions in the physical world.

Many autonomous systems make and execute complex decisions in constrained environments *or* simple decisions in complex environments. Either of these cases can be handled as "ordinary" development. The case of interest is one in which the data stream and output options are complex *and* the field of controlled action is extensive and complex.

Some specific examples may be illustrative at this point. Deep Blue, the chess-playing computer, is a pure example of an autonomous system that makes complex decisions but acts within a constrained environment. There are no consequences to chess playing in the real world that require special attention. Factory robots exhibit complex behaviors but, much like Deep Blue, operate in a constrained environment—including an environment that is constrained even in the case of failure. In another example, the Global Hawk Unmanned Aerial Vehicle (UAV) has some capability to select an airfield and land when certain failures occur. This is an impressive accomplishment but, again, relies on the constrained environment of an airfield landing strip. On the opposite end of the spectrum, the automotive industry has produced several autonomous systems that must operate in the full range of automotive environments. These systems have been successful in large measure because of their simple input data streams and simple basis for decision making. Anti-lock brake (ALB) systems and air bags are two examples.

In contrast to these examples, an example of the case of interest for an Intelligent System would be a "robo-medic": an autonomous robotic vehicle capable of operating in collapsed buildings or mines, locating injured people, performing triage, and perhaps diagnosing or even treating injuries. (Think of St. Bernards dispensing brandy.) A more topical example might be the robot-assisted surgery discussed in the Human Robot Interface and Human Machine Interaction sessions of this conference.[5]

In summary, this paper proposes the following description of what distinguishes an *Intelligent* System from a system that is merely *autonomous*:

- An *autonomous* system is one that makes and executes a decision to achieve a goal without full, direct human control.
- An *Intelligent* System is one that performs autonomously in complex and/or new environments.

B. Proof of Concept (PoC) for Intelligent Systems

As discussed previously, the TRLs initially proposed by NASA were intended for hardware systems. Modification of some TRLs was needed for software systems. Upon initiating the line of research discussed in this paper, we anticipated that further modification to the TRLs would be needed for Intelligent Systems development, particularly those TRLs addressing the concepts of "relevant and operational environments". At this point in our investigation, however, we believe the TRL language is suitable (i.e., without modification) for Intelligent Systems, but we also believe that more and earlier attention must be paid to the concepts of "relevant environment" and "metrics to assess performance". Specifically, in the development of ordinary hardware and software systems, precise definition of performance requirements (including the performance metrics on which they are based and the environments in which they are tested) are not needed until TRL 5. We believe that Intelligent System development requires at least an initial definition of environments and performance metrics at a stage earlier than TRL 5, such as during the PoC stages of TRLs 2 and 3.

In a typical PoC demonstration of an "ordinary" hardware or software system, the definition of environments is often based upon convenience, and little attention is given to size, weight, and power. Furthermore, the performance of the system is often defined in several vague and qualitative ways because one specific, quantitative set of metrics to define performance is often not yet agreed upon. It is only after PoC is demonstrated that performance requirements are defined, and this often does not occur until TRL 5. These requirements define (1) the range and type of environments in which the system must perform, (2) the quantitative metrics used to measure how well the system performs in those environments, and (3) the thresholds placed upon the metrics, showing the minimum performance level the system must meet.

In contrast, during a PoC demonstration of an Intelligent System, the metrics used to define performance and the environments in which the performance is judged must already be defined because an Intelligent System often requires training. As we noted earlier, the purpose of an Intelligent System is to make a decision or recommendation in a manner that is in some fashion better than or as good as a human. To achieve this ability, training is often essential. As is discussed in the following sections, the training requirement of Intelligent Systems necessitates the definition of "relevant environment" and "performance metrics" at an earlier stage in the TRA process than would be necessary for "ordinary" hardware or software systems.

i) Operational and Relevant Environments

In the development of "ordinary" hardware and software systems, environments considered during PoC are often too narrow, which leads to a "point solution" without the growth potential we associate with intelligence. This is problematic for Intelligent System development for two reasons:

1. By definition, an Intelligent System must perform in complex environments. Furthermore, the ability to know these environments in advance may not be possible. Defining an environment that is broad enough to encompass all types of environments in which the Intelligent System may later have to function can be a challenge.
2. An Intelligent System often requires training using a set of data labeled with ground truth information, and consolidating such data into a training set can often be a challenge. To avoid a point solution or other suboptimal outcome, the training data must be drawn from the comprehensive set of environments in which the Intelligent System may later perform. Thus, the environments must be defined before the training data is collated and input to the Intelligent System training module. If a trained Intelligent System is required to demonstrate PoC at TRL 3, then the environments that dictate the characteristics of the training data are required before TRL 3.

ii) Metrics to Assess Performance

During the development of "ordinary" hardware and software systems, metrics to assess system performance (and the thresholds placed upon them) may not be defined until as late as TRLs 5 and 6. Intelligent System development, however, requires earlier attention to performance metrics since these metrics define the quantitative criteria that must be minimized (or maximized) with the learning rule of the training process. Since training of some sort may be required to show PoC at TRL 3, these performance metrics must be defined before TRL 3.

These early defined metrics can, and should, later evolve into some of the more rigorous system requirements put into place at TRLs 5 and 6. At that point, thresholds can be assigned to each of the metrics to show the minimum level of performance above which the system must demonstrate in order for the program to proceed further. Thus, while the technology developer uses the performance metrics at TRLs 2 and 3 to probe "how well" the technology can be envisioned to perform, the system developer uses the evolved metrics (and related thresholds) at TRLs 5 and 6 to determine if the technology performs "good enough".

C. Summary

An Intelligent System makes complex decisions or recommendations in a complex environment in place of a human. With little modification, the TRA process can be applied to the development of an Intelligent System. However, special attention must be paid in TRLs 2 and 3 during the PoC phase of Intelligent System development. Specifically, technology developers must formulate at an initial definition of the "environment" and "performance metrics" at TRLs 2 and 3 rather than waiting until TRLs 5 and 6, as is possible in the development of ordinary hardware and software systems. Intelligent Systems require this early attention to environments and performance metrics because training is often essential for Intelligent System functionality. While the considered environments define the breadth and scope of the labeled data input to the training module of the Intelligent System, the performance metrics define the quantitative measure that is minimized (or maximized) by the learning rule during the training process.

IV. Conclusion

Both complex data and a complex sphere of action are required for an Intelligent System to be substantively different from an ordinary system. For a given mission, one can sometimes collapse either the data or the sphere of action, resulting in a much simpler development. Even in the fully complex case, the current language of the TRL definitions appear to us adequate to ensure sufficient technical maturity as developments pass from the technology to the product phase. However, for these fully complex cases, successful Intelligent Systems development requires early attention to the breadth and unpredictability of the environments. In addition, early identification of performance metrics will be a prerequisite for the training aspects of Intelligent Systems.

REFERENCES

[1] GAO/NSIAD-00-137, *Defense Acquisition: Employing Best Practices Can Shape Better Weapon System Decisions*, April 26, 2000. Available on-line: http://www.gao.gov.new.items/ns00137t.pdf

[2] Department of Defense, Technology Readiness Assessment (TRA) Deskbook, May 2005. Available on-line: http://www.defenselink.mil/ddre/doc/tra_deskbook_2005.pdf

[3] GAO-07-336, *Major Construction Projects Need a Consistent Approach for Assessing Technology Readiness To Help Avoid Cost Increases and Delays*, March 2007. Available on-line: http://www.gao.gov/new.items/d07336.pdf

[4] Dr. Alex Zelinsky, “Building Autonomous Systems of High Performance Reliability and Integrity,” Invited talk, PerMIS 2007.

[5] S. Schipani and E. Messina, “Maze Hypothesis Development in Assessing Robot Performance During Teleoperation,” Tue-PM2 PerMIS 2007, and N. Dagalakis, Y. Kim, D. Sawyer, and C. Shakarji, “Development of Tools for Measuring the Performance of Computer Assisted Orthopaedic Hip Surgery Systems,” Wed PM1 PerMIS 2007.

A Brief History of PRIDE

Z. Kootbally, C. Schlenoff and R. Madhavan
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD, USA
Email: {zeid.kootbally, craig.schlenoff, raj.madhavan}@nist.gov

Abstract — PRIDE (PRediction In Dynamic Environments) is a framework that provides an autonomous vehicle's planning system with information that it needs to perform path planning in the presence of moving objects. The underlying concept is based upon a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. This framework supports the prediction of the future location of moving objects at various levels of resolution, thus providing prediction information at the frequency and level of abstraction necessary for planners at different levels within the hierarchy.

This paper presents the chronology of the development of the PRIDE framework. We describe the different prediction algorithms developed for moving object predictions. We provide details on different work performed specifically for each prediction algorithm and how these algorithms are used together to give better predictions. The chronology also relates the successive simulation packages and testbeds[1] used in each step of the development of the PRIDE framework.

Keywords: *4D/RCS, aggressivity, autonomous vehicles, critical time points, long-term prediction, moving object prediction, PRIDE, short-term prediction, integration methodology.*

I. Introduction

The field of autonomous ground vehicles has made prominent strides during the last decade. Advancements have been made in methods for autonomous navigation of autonomous vehicles in dynamic environments. Funding for research in this area has continued to grow over the past few years, and recent high profile funding opportunities have started to push theoretical research efforts into practical use. Autonomous systems in this context refer to embodied intelligent systems that can operate fairly independently from human supervision. Many believe that the DEMO III Experimental Unmanned Vehicle (XUV) effort represents the state of the art in autonomous off-road driving [17]. This effort seeks to develop and demonstrate new and evolving autonomous vehicle technology, emphasizing perception, navigation, intelligent system architecture, and planning. It should be noted that the DEMO III XUV has only been tested in highly static environments. It has not been tested in on-road driving situations, which include pedestrians and oncoming traffic. There have also been experiments performed with autonomous vehicles during on-road navigation. Perhaps the most successful has been that of Dickmanns [2] as part of the European Prometheus project in which the autonomous vehicle performed a trip from Munich to Odense (> 1600 kilometers) at a maximum velocity of 180 km/h. Although the vehicle was able to identify and track other moving vehicles in the environment, it could only make basic predictions of where those vehicles were expected to be at points in the near future, considering the vehicle's current velocity and acceleration. The agent architecture AUTODRIVE [19] simulates the generation and execution of a driver's plan to reach a destination safely while taking account of other road users and obeying traffic signs and signals. The selection of appropriate goals is made through a process of "dynamic goal creation" that causes the continual run-time creation and modification of sub-goals.

Most of the work in the literature dealing with drivers' actions and predicted behavior has been performed by psychologists in an attempt to explain drivers' behaviors and to identify the reason for certain dysfunctions [1], [3], [7]. Our research interest bears upon a level of situation awareness of how other vehicles in the environment are expected to behave considering the situation in which they find themselves. When humans drive, they often have expectations of how each object in the environment is expected to move according to the situation they find themselves in. When a vehicle is approaching an object that is stopped in the road, we expect it to slow down behind the object or try to pass it. When we see a vehicle with its blinker on, we expect it to turn or change lanes. When we see a vehicle traveling behind another vehicle at a constant speed, we expect it to continue traveling at that speed. The decisions that we make in our vehicle are largely based on these assumptions about the behavior of other vehicles.

To address this need, we have developed a multi-resolutional, hierarchical framework, called PRIDE (PRediction in Dynamic Environments) that provides an autonomous vehicle's planning system with information that it needs to perform path planning in the presence of moving objects [12], [15]. This framework supports the prediction of the future location of moving objects at various levels of resolution, thus providing prediction information at the frequency and level of abstraction necessary for planners at different levels within the hierarchy.

This paper presents the chronology of the development of the PRIDE framework, starting back in 2003 when the

[1] Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

initial concept called Moving Object Representation, Prediction, and Planning System (MORPPS) was first introduced using a Kalman filter-based prediction approach. In 2004, we started using the AutoSim simulation package to provide higher resolution simulations of moving objects and on-road driving. We also introduced a second set of prediction algorithms that predicted at longer timeframes (seconds into the future as opposed to tenths of seconds). The term PRIDE appeared in 2005 and looked at using the outputs of the two prediction approaches to strengthen/weaken the results of the other. PRIDE was also applied to simulate realistic traffic patterns during on-road driving by using the longer-term prediction algorithms to control individual vehicles on a crowded roadway. More recently, in 2006 and 2007, work has been performed to determine the future time horizons when the different prediction algorithms give the best results. We also started incorporating driver aggressivity into the longer-term algorithms, and determined how the perceived aggressivity of a driver in the environment affected the future position of the vehicle they were driving. During this same time, we ported the PRIDE algorithms over to the Mobility Open Architecture Simulation and Tools (MOAST) and the Urban Search and Rescue Simulation (USARSim) framework [16], which provided a higher-fidelity simulation platform with a physic-based engine.

This paper is organized as follows: Section II presents the initial concept called Moving Object Representation, Prediction, and Planning System (MORPPS), which explored logic-based motion prediction while using different prediction algorithms for different environments. Section III provides an overview of the PRIDE framework. Section IV gives details on the short-term prediction approach along with the description of LAser Detection And Ranging (LADAR) noise models. Section V describes the second prediction approach, the long-term, cost-based, probabilistic moving object prediction algorithms. Section VI provides information on different works performed on the integration of the long-term and short-term predicted estimates. Section VII discusses the role of aggressivity in PRIDE and describes how it is addressed using MOAST and the USARSim simulation environment. Section VIII concludes the paper and gives an overview on future work.

II. The Days of MORPPS

The initial moving object framework called MORPPS (Moving Object Representation, Prediction, and Planning System) [14] was developed in 2003. This framework provides a mechanism to apply appropriate prediction algorithms and representational approaches in order to fully capture the information needed to navigate in the presence of moving objects.

A. Logic-Based Motion Predictions in Constrained Environments

The framework explores logic-based prediction algorithms for use in constrained environments. The purpose of these algorithms is to predict the probability that an object will occupy a given location in space at a given time by taking into account: a) the constraints that are placed on the object's motion and b) the influencing factors that would cause it to take a given action over another at specific times.

In the case of on-road driving, vehicles must stay on the road and as such, the road network provides the constraints dictating the bounds in which a vehicle may travel. A database structure [4] has been developed to capture detailed information about the road network, which includes information about the curvature of lanes, road interconnectivity, signage and traffic control, lane marking, etc.

The rule-based prediction approach requires that one discretizes the possible actions that a moving object may take. In the case of a vehicle driving on-road, we limit the actions of the vehicle to be: remain at a constant velocity in the current lane, slowly accelerate in the current lane, rapidly accelerate in the current lane, slowly decelerate in the current lane, rapidly decelerate in the current lane, change to a lane on the left, change to a lane on the right, turn to a lane on the left (at an intersection), turn to a lane on the right (at an intersection), make a U-Turn (at an intersection).

B. Constraints on Motion and Influencing Factors

Different factors can affect the probabilities associated with the possible actions that a vehicle may take while driving on-road. There are two classes of factors that we must consider. The first are factors that limit the possibilities of where the vehicle is able to reach. In other words, by considering these factors, we can eliminate certain portions on the maps that are not reachable by the vehicle. We call these *constraints on motion*. An example of a constraint on motion is *a priori* road network information, where the road network limits the possible locations that the vehicle can possibly attain.
The second are factors that influence which of the possible actions the vehicle is likely to perform out of those that are available to it. We call these *influencing factors*. An example of an influencing factor can be the weather and the environmental conditions. Weather and environmental conditions affect the visibility and slickness of the road surfaces. As the weather and environmental conditions worsen, the probability often increases that the vehicle's velocity will decrease.

III. The PRIDE Framework

Many efforts on the framework led to the second generation of MORPPS called PRIDE (PRediction In Dynamic Environments) that was conceived in 2004. From this time, we consider PRIDE as a multi-resolutional, hierarchical framework that provides an autonomous vehicle's planning system with information required to perform path planning in the presence of moving objects. This framework supports the prediction of the future location of moving objects at various levels of resolution, thus providing prediction information at the frequency and level of abstraction necessary for planners at different levels within the hierarchy. To understand the way that PRIDE was developed and the functionality that it is intended to provide, it is important to understand the 4D/RCS architecture [5] on which it was based. 4D refers to the four dimensions (three dimensions of space and one dimension of time), and RCS stands for Real-time Control Systems. The 4D/RCS architecture provides a reference model for unmanned vehicles on how their software components should be identified and organized. It defines ways of interacting to ensure that high-level objectives can be met. To achieve this, the 4D/RCS reference model provides well defined and highly coordinated sensory processing, world modeling, knowledge management, cost/benefit analysis, behavior generation, and messaging functions, as well as the associated interfaces.

The 4D/RCS conceptual framework spans the entire range of operations that affect intelligent vehicles, from those that take place over time periods of milliseconds and distances of millimeters to those that take place over time periods of months and distances of thousands of kilometers. The 4D/RCS model is intended to allow for the representation of activities that range from detailed dynamic analysis of a single actuator in a single vehicle subsystem to the combined activity of planning and control for hundreds of vehicles and human beings in full dimensional operations covering an entire theater of battle. In order to span the wide range of activities included within the conceptual framework, 4D/RCS adopts a multilevel hierarchical architecture with different range and resolution in time and space at each level, as shown for a military environment in Figure 1 [5] and described below.

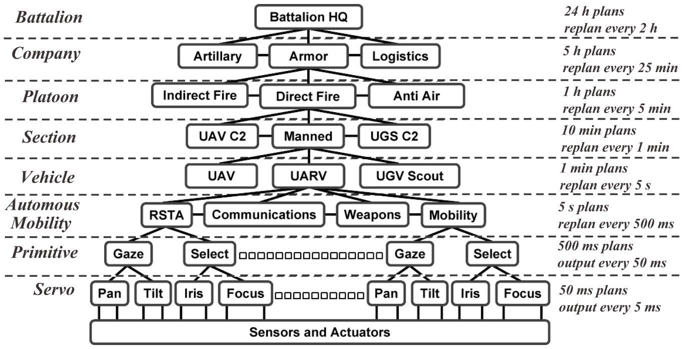


Fig. 1. A high level block diagram of a typical 4D/RCS reference model architecture.

At the Servo level, commands to actuator groups are decomposed into control signals to individual actuators. Outputs to actuators are generated every 5 milliseconds (ms). Plans that look ahead 50 ms are regenerated for each actuator every 5 ms. Plans of individual actuators are synchronized so that coordinated motion can be achieved for multiple actuators within an actuator group. At the Primitive level, multiple actuator groups are coordinated and dynamical interactions between actuator groups are taken into account. Plans look ahead 500 ms and are recomputed every 50 ms. At the Autonomous Mobility level, all the components within an entire subsystem are coordinated, and planning takes into consideration issues such as obstacle avoidance and gaze control. Plans look ahead 5 seconds (s) and replanning occurs every 500 ms. At the Vehicle level, all the subsystems within an entire vehicle are coordinated to generate tactical behaviors. Plans look ahead 1 minute (min) and replanning occurs every 5 s. At the Section level, multiple vehicles are coordinated to generate joint tactical behaviors. Plans look ahead 10 min and replanning occurs about every minute. At the Platoon level, multiple sections containing a total of 10 or more vehicles of different types are coordinated to generate platoon tactics. Plans look ahead an hour (h) and replanning occurs every 5 min. At the Company level, multiple platoons containing a total of 40 or more vehicles of different types are coordinated to generate company tactics. Plans look ahead 5 h and replanning occurs every 25 min. At the Battalion level, multiple companies containing a total of 160 or more vehicles of different types are coordinated to generate battalion tactics. Plans look ahead 24 h and replanning occurs at least every 2 h.

The PRIDE framework was developed to provide moving object predictions to planners running at any level of the 4D/RCS hierarchy at an appropriate scale and resolution. The underlying concept of the PRIDE framework is based on a multi-resolutional, hierarchical approach that incorporates multiple prediction algorithms into a single, unifying framework. At the higher levels of the framework (Vehicle level and above, as shown in Figure 1), moving object prediction needs to occur at a much lower frequency and a greater level of inaccuracy is tolerable. At these levels, moving objects are identified as far as the sensors can detect, and a determination is made as to which objects should be classified as "objects of interest". Once objects of interest are identified, we use the long-term prediction approach presented in section V to predict where those objects will be at various time steps into the future. At the lower levels (Autonomous Mobility level and below, as shown in Figure 1), we utilize estimation theoretic short-term predictions using sensor data as described in section IV to predict the future location of moving objects with an associated confidence measure.

IV. Implementing the Short-term Prediction Algorithm

Details on the development of a combined probabilistic object classification and estimation theoretic framework to predict the future location of moving objects, along with an associated uncertainty measure can be found in [11]. The framework proposed adopts a more generalized view of moving object representation and prediction in concurrently integrating multiple knowledge representation approaches from disparate sources to completely model the information necessary for dynamic planning.

A. The OneSAF Testbed (OTBSAF)

In this approach, the prediction algorithms are tested using the OneSAF (OTBSAF) testbed as the virtual sensor. OTBSAF is a simulation package used for integrating, testing and user feedback of technology developments into the OneSAF Objective System. It provides operational environments useful for identifying, developing, prototyping, demonstrating, and testing of enabling technologies and entity behaviors. As a simulated environment, OTBSAF is able to represent moving objects. By querying OTBSAF, we can retrieve an object's location and velocity at the current time. To validate the testbed and prediction algorithms, we are initially using this retrieved data to serve as our processed sensor data.

B. LADAR Noise Model

In this work, the LADAR sensor is the primary source of sensor data. The data retrieved from OTBSAF is perfect sensor data. In other words, when we ask for the location or dimensions of the object, we are presented with the exact location and the exact dimensions without any associated uncertainty. Although convenient, this does not represent the information that we expect to get from sensors on the actual vehicle. To compensate for this, we have introduced a noise model into the data retrieved from OTBSAF [11].

C. Prediction of Moving Objects

An Extended Kalman Filter (EKF) is employed to predict (estimate) the position and velocity of the moving object at a future time instant. Kalman's prediction theory allows the computation of the best estimate of a future system state by using the most recent estimates of system state along with the system dynamic model. With appropriate interpretation, covariance analysis inherent in the Kalman filtering techniques serves as a confidence measure indicative of the uncertainty in the predicted system states. The EKF thus provides a convenient measure of prediction accuracy through the covariance matrix. The EKF employs a nonlinear model derived from equations based on the kinematics of the moving objects (vehicles) to be predicted.

The EKF is a well established recursive state estimation technique where estimates the states of a nonlinear system are obtained by *linearization* of the nonlinear state and observation equations. Within the PRIDE framework, short-term prediction of objects moving at variable speeds and at given look-ahead time instants (every one-tenth of a second) are predicted using the EKF. It should be noted here that, in contrast to the long-term predictions, the estimation-theoretic short-term prediction algorithm does not incorporate *a priori* knowledge such as road networks and traffic signage and assumes uninfluenced constant trajectory. More information on the short-term prediction algorithm can be found in [10].

V. Implementing the Long-term Prediction Algorithm

The long-term (LT) situation-based probabilistic prediction approach was implemented in AutoSim in 2004 [12]. Autosim was developed by the Advanced Technology Research Corporation and was used to provide higher resolution visualizations of moving objects and on-road driving. AutoSim is a high-fidelity visualization tool which models details about road networks, including individual lanes, lane markings, intersections, legal intersection traversibility, etc. Using this package, we have simulated typical traffic situations (e.g., multiple cars negotiating around obstacles in the roadway, bi-directional opposing traffic, etc.) and have predicted the future location of individual vehicles on the roadway based upon the prediction of where other vehicles are expected to be.

The LT prediction approach is used to predict the future location of moving objects for longer time horizons. Figure 2 graphically shows the overall process flow.

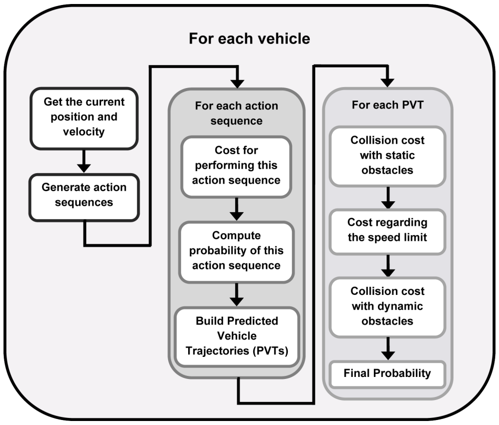

For each vehicle
Get the current position and velocity
Generate action sequences
For each action sequence
Cost for performing this action sequence
Compute probability of this action sequence
Build Predicted Vehicle Trajectories (PVTs)
For each PVT
Collision cost with static obstacles
Cost regarding the speed limit
Collision cost with dynamic obstacles
Final Probability

Fig. 2. The situation-based probabilistic (long-term) prediction process.

The output of this loop is a list of locations with associated probabilities showing where a vehicle is expected to be at specific times in the future. Using these probabilities, we can create traffic patterns in one of two ways:

- Control the vehicle to move to the location with the highest probability. For example, if the vehicle has a 40 % chance of being at location A, a 30 % chance of being at location B, a 20 % chance of being at location C, and a 10 % chance of being at location D, the vehicle will always be commanded to move to location A.
- Control the vehicle to move to a location whose likelihood is proportional to the probability that it is expected to be there. One approach would be to use a random number generator. In this way, a vehicle's movement would be closely tied to the probabilities coming out of the moving object predictor, as opposed to always moving to the location with the highest probability.

Independent of the approach used to control the vehicles, the output of these algorithms result in realistic traffic patterns involving one to many vehicles that can be used as a basis to evaluate the performance of autonomous vehicle within simulated on-road driving scenarios.

A. Possible Vehicle Actions

The process of predicting several time steps into the future consists of a series of continuous actions which constitute a driving procedure. Each action is accomplished in one time step, thus, for a time of prediction n, n actions will be completed. The long-term prediction algorithms use different types of actions. The first type of actions consists of a set of speed profiles: Quick Acceleration (QA), Slow Acceleration (SA), Keep the same Speed (KS), Quick Deceleration (QD), Slow Deceleration (SD). The second type of actions concerns the changing of lanes: a vehicle has the possibilities of staying in its lane (SL), changing to the right lane (CR), changing to the left lane (CL). The last type of action pertains to intersections, a vehicle has the possibility to turn left, to turn right or to go straight through an intersection.

At this step, for each vehicle on the road, the algorithm computes all possible sequences of actions, regarding the current velocity and location. Some actions may not be possible due to the vehicle's current velocity (for example, a vehicle moving slowly cannot change lanes in one second during a deceleration). In this case, those actions are not considered. Each sequence of actions is generated in a realistic way using rules. Presently, a single rule is applied to all of the possible action sequences to generate the most realistic ones. To evaluate these rules, we associate a value to each 'acceleration profile': 2 for QA, 1 for SA, 0 for KS, -1 for SD, and -2 for QD. The rule states that a vehicle can only switch from an action to another action if their values differ at most by one. An example of action sequences and their associated validity is shown in Table I.

TABLE I

EXAMPLE OF VALID AND INVALID SEQUENCES OF ACTIONS.

Actions				*Validity*	*Description*
SD	SD	SD	SD	Valid	
QD	QD	QA	QA	Invalid	QD to QA illegal

B. Cost Model

The sequences of actions are deemed finite, and the probabilistic LT prediction algorithms use an underlying cost model that simulates the danger that a driver would incur by performing an action or occupying a state [15]. These costs are being used by multiple efforts within the program that this effort is a part of. Thus, there is value of building the probabilities directly from these costs to allow for synergy with other efforts. These costs can be separated in two different categories:

1) The cost representing the vehicle's actions: This cost represents the penalties for performing an action as a function of the amount of attention needed. For example, the changing lane action needs more concentration than going straight in the same lane, thus the cost for changing lane is greater.
2) The cost representing the vehicle's state on the road: The proximity to other static and dynamic objects on the road is assigned to a cost of collision with these objects. Examples of static objects on the road are road blocks, debris, etc. Examples of dynamic objects on the road are other vehicles. The costs associated with static or moving objects is proportional to the danger and imminence of collision. For example, a road block at one kilometer ahead is less dangerous than another vehicle passing at three meters ahead.

Examples of costs are shown in Table II.

TABLE II

EXAMPLE OF ACTIONS WITH THEIR CORRESPONDING COSTS.

Action	*Cost*
Quick Acceleration (QA)	5
Quick Deceleration (QD)	5
Changing lane (CL, CR)	20
Opposite direction	500
Collision (CO)	1000
Being under the speed limit (US)	5
Being over the speed limit (OS)	5

C. Predicted Vehicle Trajectory

Costs of collision between vehicles are computed using Predicted Vehicle Trajectories (PVTs) which represent the possible movements of vehicle throughout the time period of prediction being analyzed. A PVT is a vector whose origin represents the current position of the vehicle ($x_{IP}, y_{IP}, t_{IP} = 0$) at $time = 0$ and its extremity represents the predicted position ($x_{PP}, y_{PP}, t_{PP} = t_{pred}$) where t_{pred} is the predetermined time in the future for the prediction process. Also contained within the PVT is the action-cost and action-probability information.

A collision is detected when PVTs cross each other, the location and time of the collision is determined using a parametrization of each PVT. This information can be obtained by using a parametrization of each PVT as represented in the following equations.

$$\begin{cases} x_1(t_1) = x_{PP_1} t_1 + x_{IP_1}(1 - t_1) \\ y_1(t_1) = y_{PP_1} t_1 + y_{IP_1}(1 - t_1); \; t_1 \in [0, 1] \end{cases} \tag{1}$$

$$\begin{cases} x_1(t_2) = x_{PP_2} t_2 + x_{IP_2}(1 - t_2) \\ y_1(t_2) = y_{PP_2} t_2 + y_{IP_2}(1 - t_2); \; t_2 \in [0, 1] \end{cases} \tag{2}$$

where t_1 and t_2 are the parameters for each PVT. Equations (1) and (2) create a linear system where t_1 and t_2 can be solved using Cramer's rule:

$$t_1 = \frac{\begin{vmatrix} x_{IP_2} - x_{IP_1} & x_{IP_2} - x_{PP_2} \\ y_{IP_2} - y_{IP_1} & y_{IP_2} - y_{PP_2} \end{vmatrix}}{\begin{vmatrix} x_{PP_1} - x_{IP_1} & x_{IP_2} - x_{PP_2} \\ y_{PP_1} - y_{IP_1} & y_{IP_2} - y_{PP_2} \end{vmatrix}}$$

$$t_2 = \frac{\begin{vmatrix} x_{PP_1} - x_{IP_1} & x_{IP_2} - x_{IP_1} \\ y_{PP_1} - y_{IP_1} & y_{IP_2} - y_{IP_1} \end{vmatrix}}{\begin{vmatrix} x_{PP_1} - x_{IP_1} & x_{IP_2} - x_{PP_2} \\ y_{PP_1} - y_{IP_1} & y_{IP_2} - y_{PP_2} \end{vmatrix}}$$

The two vehicles will cross each other at two different times, (t_1, t_{pred}) for the first vehicle, (t_2, t_{pred}) for the second vehicle. For a small difference between the two times, the collision is probable or certain. Conversely, for a large difference, the collision is improbable. Thus if the PVTs cross and the difference of time is less than a predetermined time (τ), we use Equation (3) to determine the collision cost:

$$Collision\ Cost = CO\left(\tau - \left(t_{pred}|t_1 - t_2|\right)\right) \qquad (3)$$

where CO is the predetermined maximum cost than can occur when colliding with a specific object (Table II) and τ is the predetermined time difference in which a cost for collision will be incurred.

D. From Cost to Probability

As discussed previously, the PRIDE algorithms compute n realistic sequences of actions with an associated cost. Based on this cost, we can determine the probability that the vehicle will perform that sequence of actions in the following way. The first step is to create a ratio of the cost for performing a given sequence of actions to the sum of all of the costs for performing n sequences of actions:

$$ratio_i = \frac{\displaystyle\sum_{j=1}^{n} cost_j}{cost_i}, \forall\ i \in [1, n]$$

We then normalize the ratio of each sequence of actions by dividing it by the sum of all of the ratios, as shown in Equation (4):

$$proba_i = \frac{ratio_i}{\displaystyle\sum_{j=1}^{n} ratio_j}, \forall\ i \in [1, n] \qquad (4)$$

Equation (4) computes the normalized probability of a given sequence of actions occurring as compared to all sequences of actions that are possible at that time.

VI. Integration of the Long-term and Short-term Predictions

One key component of the PRIDE framework is the ability to integrate the predictions from the two algorithms described in Sections IV and V. With this integration, we are able to increase or decrease the confidence of the results of each of these prediction algorithms based upon how well the predictions align. The methodology used to integrate the long-term prediction estimates with those provided by the short-term prediction algorithm is detailed in [15].

A. Significance of Critical Time Points

We define critical time points as those that lie between time periods when both ST and LT provide useful estimates. This is important as it provides opportunities for leveraging the predictions when both prediction algorithms provide valuable estimates during these times. To facilitate discussion, we define t_{bp} as the *break-off point* beyond which the ST estimates are of little value.

When an exteroceptive sensor observation becomes available, the innovation and the innovation covariance (which is a 2×2 matrix as we are considering x_v and y_v), are checked to determine if the EKF updates are to be performed with that observation. The following two conditions are checked to determine if the observation falls within 2σ (95 %) bounds:

$$\left|\left(\frac{\nu(1)}{\sqrt{\mathbf{S}(1,1)}}\right)\right| <\ 2.0 \qquad \text{and} \qquad \left|\left(\frac{\nu(2)}{\sqrt{\mathbf{S}(2,2)}}\right)\right| <\ 2.0$$

If the above conditions are not satisfied, the ST estimates will no longer be bounded (the covariances of the position estimates grow without bounds) and accordingly their consistency cannot be guaranteed. The time instant at which this occurs is termed the break-off point, t_{bp}.

B. Experimental Results

The integration of the predictions from the two algorithms has been performed in different ways through important efforts.

In 2005, work was performed to apply the integration methodology on a straight line with obstacle avoidance [15]. The resulting prediction estimates showed that while the ST predictions provide accurate position estimates within a shorter time horizon, the quality of the predictions degrade considerably as the time horizons get longer. Conversely, the LT prediction algorithms specifically address this shortcoming by providing realistic estimates at longer time horizons that are amenable for autonomous on-road driving. The probabilistic scaling methodology was used to integrate the two prediction algorithms more tightly, such that the results of the ST prediction can help to validate those of the LT prediction and vice-versa.

In 2006, a new way to apply the integration methodology was implemented [10]. To analyze the performance of the prediction algorithms and to determine the window in which both the ST and LT algorithms provide reasonable results, we let the vehicle traverse the track until the first break-off point occurs. As mentioned earlier, the break-point occurs when the ST estimates are no longer consistent. The integration methodology is used on the ST and LT estimates belonging to the time period $[0 - t_{bp}]$ by varying the speed of the vehicle and the time of prediction.

During the same year, in our last effort using the integration methodology, we have tested the performance of the ST and the LT prediction algorithms with several data sets of varying data rates, speeds and prediction intervals on a closed-track [8]. The results have consistently demonstrated that ST estimates are superior to LT estimates in the time period $[t_0 - 0.25t_{bp}]$ and the LT estimates are to be preferred in the time period immediately after t_{bp} until $2t_{bp}$ especially when no external corrections are available for ST prediction. Subsequently, $[0.25t_{bp} - t_{bp}]$ is the most desired time period for the integration of the ST and LT estimates. We compare the results of the integration methodology performed in the two mentioned time periods along a closed-track. We use the last LT estimates from the previous integration to find the next break-off point, and we repeat the same process until the last break-off point of the track.

VII. Driver Aggressivity

The addition of aggressivities is the latest enhancement to the PRIDE framework. The term aggressivity in this context refers to the following description [18]:

> *A driving behaviour is aggressive if it is deliberate, likely to increase the risk of collision and is motivated by impatience, annoyance, hostility and/or an attempt to save time.*

The aggressivity feature was developed after the integration of the PRIDE framework with the Open Architecture Simulation and Tools (MOAST) and the Urban Search and Rescue Simulation (USARSim) simulation environment [16]. This effort provides predictions incorporating the physics, kinematics and dynamics of vehicles involved in traffic scenarios.

A. Mobility Open Architecture Simulation and Tools (MOAST)

MOAST is a framework that provides a baseline infrastructure for the development, testing, and analysis of autonomous systems that is guided by three principles: 1) Creation of a multi-agent simulation environment and tool set that enables developers to focus their efforts on their area of expertise, 2) Creation of a baseline control system which can be used for the performance evaluation of the new algorithms and subsystems, and 3) Creation of a mechanism that provides a smooth gradient to migrate a system from a purely virtual world to an entirely real implementation.

MOAST implements a control technique which decomposes the control problem into a hierarchy of controllers with each echelon (or level) of control, adding more capabilities to the system. Module-to-module communications in MOAST is accomplished through the Neutral Message Language (NML) [6], based on a message buffer model.

B. Urban Search And Rescue Simulation (USARSim)

USARSim is a high-fidelity physics-based simulation system that provides the embodiment and environment for the development and testing of autonomous systems. This is an open source simulation environment that is based on Epic Games Unreal Tournament 2004. Originally developed to study human robotic interactions in multi-agent environment in an Urban Search And Rescue (USAR) environment [9], USARSim is expanding its capabilities to provide realistic simulation environments to assist in the development and testing of cognitive systems, autonomous nautical vessels, and autonomous road driving vehicles.

USARSim utilizes the Karma Physics engine and high-quality 3D rendering facilities of the Unreal game engine to create a realistic simulation environment that provides the embodiment of a robotic system.

C. System architecture of the MOAST/USARSim and PRIDE Frameworks

The embedded client-server architecture (Figure 3) of the Unreal game engine enables USARSim to provide individualized control over multiple robotic systems through discrete socket interfaces. The interfaces provide a generalized representation language that enables the user to query and control the robots' subsystems. All the communications between the clients (Unreal client and the Controller) and the server are performed through the network. The Unreal Server includes the Unreal Engine, Gamebots to bridge the Unreal Engine with outside applications, the maps and the models (robot models, victims, etc). MOAST first connects to the Unreal Server, then it sends commands to USARSim to spawn a robot. At this step, MOAST listens to the sensor data and sends commands to control the robot.

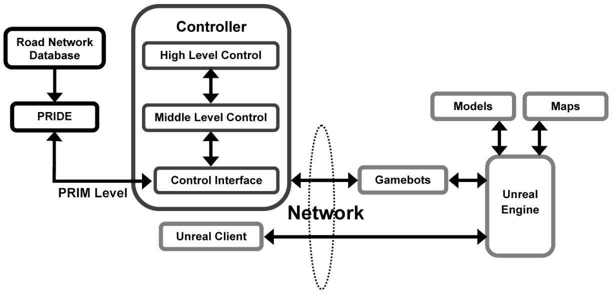


Fig. 3. System architecture of USARSim, MOAST and PRIDE.

As depicted in Figure 3, PRIDE uses a Road Network Database [4] to retrieve the information about road networks for the moving object prediction process. The purpose of the Road Network Database is to provide the data structures necessary to capture all of the information necessary about road networks so that a planner or control system on an autonomous vehicle can plan routes along the roadway at any level of abstraction. The PRIDE framework assumes knowledge of the current position and the velocity of the vehicles on the road to predict their future locations. The PRIDE algorithms retrieve the status (position and velocity) of every vehicle by querying their corresponding navigation channel. At this step, the information from the Road Network Database is used to compute the future positions of the moving objects. The data commands are sent to MOAST through the Primitive level.

D. Modeling Aggressivity within PRIDE

Unlike other approaches that use an underlying static cost model for activities such as path planning, this approach introduces the concept of a dynamic cost model, where the costs are vehicle specific and are a function of what is perceived in the environment. As explained in section V, we associate underlying costs to various actions and states. We then sum the costs that are associated with a specific driving maneuver and use that overall cost to determine the probability that a vehicle will perform that maneuver; the higher the cost to perform the maneuver, the lower the probability that it will occur. However, different drivers have different driving behaviors, and thus have different underlying costs model. One driver may be very conservative, only changing lanes when absolutely necessary, never exceeding the speed limit, etc. On the other hand, another driver may drive very aggressively, weaving in and out of lanes, greatly exceeding the speed limit, and tailgating other drivers. In most cases, one would experience both kinds of drivers on any trip (along with many drivers that fall somewhere in the middle), and a moving object prediction framework needs a mechanism to account for all such circumstances.

When a driver is first encountered, it is extremely rare that one can instantaneously determine the perceived aggressivity of the driver. This information is often determined after observing the driver for a certain amount of time, characterizing their driving behaviors, and assigning an aggressivity. The aggressivity that is assigned greatly impacts PRIDE's predictions as to where that driver will be at times in the future. For example, we would likely assume that a conservative driver will remain in their lanes whenever possible and stay a safe distance behind the vehicle in front of him. An aggressive driver would have a higher probability of changing lanes. We may also find that the aggressivity of the driver may change over times. There are times when one can observe a driver for many seconds at a time. In this case, the driver's aggressivity may change, perhaps they are very aggressively trying to get to a certain lane but become more passive when they get there.

The PRIDE framework addresses all of these driver types and all of the situations mentioned above. Experiments and corresponding results performed on the aggressivity can be found in [13].

VIII. Conclusions and Future Work

The utility of algorithms of predictions has proved to be particularly important with emphasis on complex path planning for autonomous vehicles in dynamic environments. This paper presented the chronology of the development of the PRIDE framework, a hierarchical, multi-resolutional approach for moving object prediction during autonomous on-road driving. We discussed the different concepts used during each step of the development of PRIDE. We described the different prediction algorithms, how they can be used to predict the future location of moving objects. We then showed the features within PRIDE and how they individually make the strength of each algorithm. We also detailed how the short-term and long-term algorithms can be unified to provide better predictions and we gave an overview of different efforts using the integration methodology. We provided an overview of the successive simulation packages used to accomplish more complex traffic situations and used to implement the set of features that constitute this framework today.

Although substantial progress has been made in designing and implementing the PRIDE framework, there is still much to be done. In order to have more complicated traffic situations, we plan on using multiple vehicles in more complex road networks, even though PRIDE is not limited algorithmically to deal with multiple vehicles. In future papers we will tape a real traffic scenario and compare the results to those provided by PRIDE, in this way we can analyze how well PRIDE predicts the future location of the vehicles. PRIDE aims to integrate fuzzy logic for traffic negotiation at intersections and for identification of object of interests. We also plan to upload a release of PRIDE on sourceforge once a stable version is available.

References

[1] A. Champion, S. Espie, and J. Auberlet. Behavioral Road Traffic Simulation with ARCHISM. In *Proceedings of the Summer Computer Simulation Conference*, 2001.

[2] E. Dickmanns. The Development of Machine Vision for Road Vehicles in the Last Decade. In *Proceedings of the International Symposium on Intelligent Vehicles*, pages 644–651, 2002.

[3] S. Espie, F. Saad, and B. Schnetler. Microscopic Traffic Simulation and Driver Behavior Modeling: The ARCHISM Project. In *Proceedings of the Strategic Highway Research Program (SHRP) and Traffic Safety on Two Continents*, 1994.

[4] C. Schlenoff et al. The NIST Road Network Database: Version 1.0. Technical Report NISTIR 7136, National Institute of Standards and Technology, July 2004.

[5] J. Albus et al. 4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems. Technical Report NISTIR 6910, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2002.

[6] V. Gazi, M. Moore, K. Passino, W. Shackleford, and F. Proctor. *The RCS Handbook: Tools for Real-Time Control Systems Software Development*. John Wiley & Sons Inc., New York, NY, June 2001.

[7] S. Hoseini, M. Vaziri, and Y. Shafahi. Combination of Car Following and Lane Changing Models as a Drivers' Optimization Process. *Applications of Advanced Technologies in Transportation Engineering*, 0-7844-0730-4:601–605, 2004.

[8] Z. Kootbally, R. Madhavan, and C. Schlenoff. Prediction in Dynamic Environments via Identification of Critical Time Points. In *Proceedings of the IEEE Workshop on Situation Management (SIMA), Military Communications Conference*, Washington DC, October 2006.

[9] M. Lewis, K. Sycara, and I. Nourbakhsh. Developing a Testbed for Studying Human-Robot Interaction in Urban Search and Rescue. In *Proceedings of the International Conference on Human Computer Interaction (HCI)*, pages 270–274, Crete, Greece, June 22–27 2003.

[10] R. Madhavan, Z. Kootbally, and C. Schlenoff. Prediction in Dynamic Environments for Autonomous On-Road Driving. In *Proceedings of the IEEE International Conference on Control, Automation, Robotics and Computer Vision (ICARCV)*, pages 1690–1695, December 05–08 2006.

[11] R. Madhavan and C. Schlenoff. Moving object prediction for off-road autonomous navigation. In *Proceedings of the SPIE Aerosense Conference*, volume 5083, pages 134–145, Orlando, FL, 2003.

[12] C. Schlenoff, J. Ajot, and R. Madhavan. PRIDE: A Framework for Performance Evaluation of Intelligent Vehicles in Dynamic, On-Road Environments. In *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, 2004.

[13] C. Schlenoff, Z. Kootbally, and R. Madhavan. Driver Aggressivity Analysis within the Prediction In Dynamic Environments (PRIDE) Framework. In *Proceedings of the 2007 SPIE Defense and Security Symposium*, volume 6561, May 2007.

[14] C. Schlenoff, R. Madhavan, and S. Balakirsky. An Approach to Predicting the Location of Moving Objects During On-Road Navigation. In *Proceedings of the Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments: World modeling, planning, learning, an communicating at the International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 71–79, 2003.

[15] C. Schlenoff, R. Madhavan, and Z. Kootbally. PRIDE: A Hierarchical, Integrated Prediction Framework for Autonomous On-Road Driving. In *Proceedings of the 2006 International Conference on Robotic Applications (ICRA)*, pages 2348–2353, 2006.

[16] C. Scrapper, S. Balakirsky, and E. Messina. MOAST and USARSim: A Combined Framework for the Development and Testing of Autonomous Systems. In *Proceedings of the 2006 SPIE Defense and Security Symposium*, volume 6230, Orlando, FL, April 17–21 2006.

[17] C. Shoemaker and J. Bornstein. Overview of the Demo III UGV Program. In *Proceedings of the SPIE Robotic and Semi-Robotic Ground Vehicle Technology Conference*, pages 202–211, 1999.

[18] L. Tasca. A Review of the Literature on Aggressive Driving Research. In *Aggressive Driving Conference*, 1996.

[19] S. Wood. *Planning in a Rapidly Changing Environment*. Phd thesis, School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK, 1990.

Autonomy Levels for Unmanned Systems (ALFUS) Framework: Safety and Application Issues

Hui-Min Huang
National Institute of Standards and Technology
Gaithersburg, MD 20899, U.S.A.
hui-min.huang@nist.gov

Abstract--The Autonomy Levels for Unmanned Systems (ALFUS) framework is generic and applicable to multiple unmanned system (UMS) domains. The key component of the Framework is metrics along the three established axes or aspects. This paper attempts to examine how the metrics might be applied to selected domains that include homeland security, manufacturing, and defense. In particular, the paper attempts to lay out how the critical UMS concerns, including requirements specification, performance measures, safety, and risks might be established from the Framework.

Key words: *autonomy, contextual autonomous capability (CAC), environment, human independence (HI), human robot interaction (HRI), metrics, mission, task, unmanned system (UMS)*

I. INTRODUCTION

The ALFUS Ad Hoc Working Group has been developing the ALFUS Framework aiming at providing standard terms, definitions, metrics, and tools to facilitate UMS lifecycle practices. Participants from various Government organizations and their contractors, including U.S. Departments of Commerce, Defense, Energy, and Transportation and from industry have been volunteering their efforts. The current results include a terms and definitions document [1], which has begun to be adopted by or referenced in various documents [2]. The Framework document is expected to be published soon.

ALFUS is an ongoing effort. As such, this paper highlights some key accomplishments of ALFUS, discusses current issues, as well as points out future directions.

II. FRAMEWORK

ALFUS is highlighted with a three-aspect model, as shown in Figure 1. The aspects of mission complexity (MC), environmental complexity (EC), and human independence (HI) characterize the autonomy of UMSs. The objective for a UMS autonomous operation is to achieve the missions as assigned by its human operator(s) through the designed human-robot interface (HRI) or assigned by another system that the UMS interacts with. Each of the aspects is further elaborated with a set of metrics, as described in the earlier papers, including [3, 4, 5].

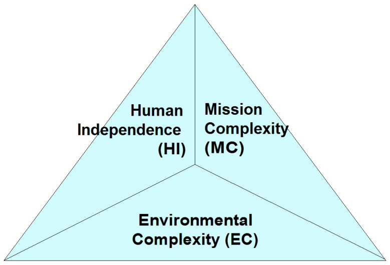


Figure 1: The Three Aspects for ALFUS

A Potential Benefits

Autonomy offers many benefits to human life. The ALFUS framework helps characterizing the autonomy. This characterization process would, in turn, help the design and evaluation of the UMS.

(1) *Enhance safety:* Human safety is the utmost concern in the modern society. However, there are tasks not suited for humans, particularly, those that must be performed in environments that may be

- dangerous—where heavy machinery may be running, a building may be collapsing, or chemical, biological, radioactive, nuclear, and explosive material might exist
- extreme—where it may be too hot, too cold, or too tight
- hostile—where enemy may be firing.

UMSs are suited for these tasks. The ALFUS Framework employs sets of definitions to facilitate communication of the issues and sets of metrics to facilitate the analysis of the issues. For example, in a dangerous environment, certain types of HRI may be needed at certain portions of the mission. The difficulty of the task may not exceed certain levels. These are just some examples for ALFUS application.

(2) *Enhance outcome:* By enhancing outcome, we mean achieving:

- mission/task/order goals
- accuracy and repeatability, in time and space
- savings in time, space, and material

For example, it has been well recognized that those tasks that are repetitive and boring to humans and those beyond human physical abilities could be easily achieved by UMSs. Also, appropriately equipped UMSs enhance the outcome. A UMS with high sensing and perception capability has a better chance of achieving a task requiring high precision.

We attempt to explore applying ALFUS from these perspectives.

III. KEY CONCEPTS IN ALFUS

A key definition in ALFUS is **Autonomy**:

> *"A UMS's own ability of integrated sensing, perceiving, analyzing, communicating, planning, decision-making, and acting, to achieve its goals as assigned by its human operator(s) through designed human-robot interface (HRI)"*

The autonomy is based on the UMS's internal capability of performing all the identified autonomy enabling functions in an integrated manner. This integrated function set forms a complete control cycle. The autonomy is further elaborated into the second key concept in ALFUS, which is called **Contextual Autonomous Capability (CAC)**:

> *"An unmanned system's contextual autonomous capability is characterized by the missions that the system is capable of performing, the environments within which the missions are performed, and human independence that can be allowed in the performance of the missions.*
>
> *Each of the aspects, or axes, namely, mission complexity, environmental complexity, and human independence is further attributed with a set of metrics to facilitate the specification, analysis, evaluation, and measurement of the contextual autonomous capability of particular UMSs"*

This CAC model facilitates the characterization of UMSs from the perspectives of requirements, capability, and levels of difficulty, complexity, or sophistication. The model also provides ways to characterize UMS's autonomous operating modes. The three axes can also be applied independently to assess the levels of mission complexity, environmental complexity, and autonomy for a UMS.

The HI axis is also referred to as the axis for level of autonomy (LOA) [6].

As defined, the CAC model encompasses multiple layers of abstraction. The following are the two essential layers:

- The Metric Model for ALFUS: UMS is characterized with defined sets of metric, including the percentage of a mission that is planned and executed by the UMS onboard processors, the levels of task decomposition, the solution ratio in the physical environment, etc.
- The Executive Model for ALFUS: a UMS is characterized with the three aspects or axes, namely, MC, EC, and HI. These axes are summaries of the individual metrics. Particularly, the weighted averages of metric scores form the axis scores. The HI scores correspond to levels of autonomy, similarly for the levels of MC and EC.

Additional layers of abstraction are allowed. For example, the human interaction time metric along the HI axis might be further decomposed to actuation time, monitoring time, sensory data acquisition time, etc. Earlier concepts even involve another, even higher layer, single CAC score that is a weighted average of the three axis scores. The CAC index is a combination of the metric scores of the three axes and the result can come from many combinations of the three axes. Figure 2 provides an illustration. However, participants are feeling that this might be an oversimplified index. Further investigation of this issue is planned.

The higher layers facilitate requirements specification and communication purposes, whereas the lower levels facilitate implementation and testing and evaluation.

In the research community, the term autonomy level may be used in different contexts. Bruemmer, D.J., et al., in [7], uses the term dynamic autonomy. Barynov and Hexmoor used terms including preference autonomy, choice autonomy, and decision autonomy [8]. In practices, autonomy levels are often used to indicate only the degrees of human independence. They are all consistent with and can be facilitated by the ALFUS CAC model.

The CAC index, including the autonomy level may be used in a nominal sense while the specific level values are dynamic or are adjusting, to the extent of the system design, along the course of mission execution depending on the changes of the environmental and operating conditions.

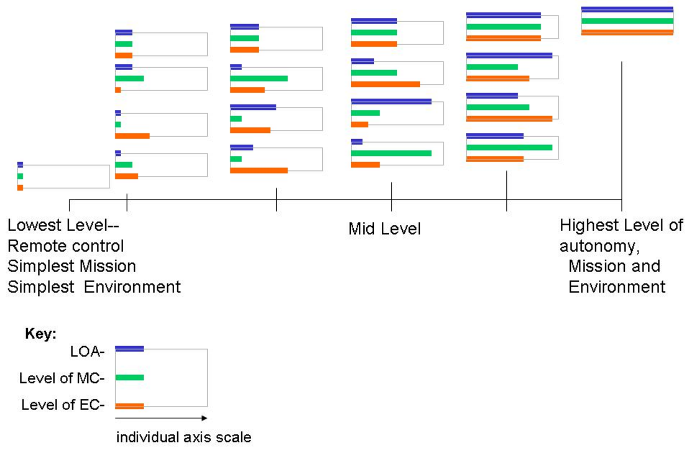


Figure 2: Illustrative Combinations of CAC

IV. ALFUS MODELS FOR UMS SAFETY, RISK AND MISSION SUCCESS

We postulate the following new, key concepts for the purpose of expanding the applicability of ALFUS for UMS.

$$(L(CACR - L(CAC)) \propto L(risk)$$

Where
CACR: CAC Requirements
L: level;
$\propto$ *: proportional to or positively corresponds to*

Note that,

$L(risk) <= 0$

indicates minimal or no risk.

This leads to the following:

$$L(safety) \propto (1 - |L(risk)|)$$

where:
$|\ |$*: normalization*

These, themselves, lead to the following observations:

Level of safety may be contributed by the following factors:

a. insufficient capability in any of the root autonomous capabilities [9]
b. insufficient scores in any of the metrics/axes.

In addition, safety may be considered as a subset of complexity, either mission or environmental.

V. SIMULATION TO FACILITATE AND EVALUATE THE BENEFITS OF AUTONOMY

It is well understood that the application of simulation can save UMS development costs. We investigated how ALFUS could facilitate the UMS simulation.

A Non-Physical Entities

To explore the benefits of UMS simulation, we need to define a set of new ALFUS terms and concepts. Although ALFUS stresses physical UMSs to distinguish itself from the general information technology (IT) world, there are situations when ALFUS is applied to non-physical entities, such as the following:

- Logical UMS (LUMS) are those inherently non-physical entities that interact with UMSs, such as a computer control and management software system like a flexible manufacturing system when it is treated as an independent entity. In a hierarchical control system, a high level control node that coordinates low-levels UMSs may be a LUMS.

- Virtual UMS (VUMS) or Soft UMS (SUMS) refers to UMSs in simulation.

LUMS, VUMS, and SMUS interact with UMSs using established communication channels.

B ALFUS to Facilitate Simulation

A UMS must be specified before development, so does a UMS simulator. There might be two approaches to the development of the simulator, one that is geared toward the specific UMS, and the other toward a generic simulation environment.

For a specific simulator, the ALFUS-based UMS specification is used as design criteria for the simulator. The VUMS should be developed to be able to perform at the level of CAC as the to-be-developed UMS.

In a generic UMS simulation environment, the simulated operating environment might be adjusted, per the design, to the desired difficulty level. For example, the friction of the roads, the slopes of the hills, the density of the traffic, etc., could all be designed as adjustable. The autonomous capabilities of the simulated UMSs can be measured and characterized. These are all benefits facilitated by ALFUS.

The mission could be scripted to the desired level of complexity, as well. Attributes such as the number of VUMSs in a team and the commanding structure, the sensory capabilities, the accuracy of the goals, etc., could be designed as adjustable to reflect the desired level of MC.

Similarly, an ALFUS enabling HRI in a simulator could be designed such that the levels of human interaction time could be adjusted, the types of interactions that the VUMS could initiate could be pre-set, the HRI displays could be adjusted to simulate different levels of stress that might be caused, etc.

VI. APPLYING ALFUS TO MANUFACTURING DOMAINS

A Rationale

Automation is a key to manufacturing efficiency and safety. The challenge is that a manufacturing process could be very complex and dynamic. It could involve operators in a semi-automated facility. It receives work orders for different products and different quantities. It may need to generate various kinds of reports that contain different kinds of information for different purposes such as production control, quality control, or maintenance analysis. The process may also need to adjust its schedules to accommodate storage or shipping constraints. Therefore, a framework for performance measure and capability characterization like ALFUS should be beneficial. The following lists some features that ALFUS could apply:

a. A flexible manufacturing system (FMS), in its entirety, could be considered a UMS. If required, the highest level system software could be considered a LUMS.
b. Equivalent missions/tasks include production orders and inspection orders at a high level, machine a part and inspect a part at a middle level, or drill a hole of X diameter and or inspect the hole at a low level. Correspondingly, the manufacturing system capability could be characterized as number of parts per day.
c. Autonomous capabilities could help the many factors that a manufacturing process may encounter, such as raw material composition/sizes/weights, equipment breakdown, etc. The variation in the raw material could cause adjustments for the equipment, including its settings, workload, and process flow. It should be beneficial that this could be done in a human-machine coordinated way.
d. It is desirable that a manufacturing process's performance be measured. ALFUS might serve this purpose.
e. The EC needs to be characterized for such conditions as a new operator could inadvertently interfere in a work volume, a part might fall off a UMS carrying the lot along the route, or a machine breaks down. In other words, a manufacturing environment could be highly unstructured.
f. The low level machining instructions correspond to the low level skills as identified in the military UMS domain. Skills have different levels of difficulty, so are the machining instructions. For example, for inspecting holes, tolerances make differences in terms of difficulty.

B Toward ALFUS Measures and Indices

Autonomous capability related measures, derived from ALFUS, could help characterizing a manufacturing process in the following possible ways:

a. Highly autonomous manufacturing UMS might correspond to higher initial equipment cost but lower overall lifecycle cost as well as higher capability for complex "missions," i.e., products.

   $$\left|\textit{initial cost}\right| \propto L(CAC))$$

   $$\left|\textit{lifecycle cost}\right| \propto (1 - L(CAC))$$

b. Lower complexity products might mean that they are suitable for mass production on low CAC manufacturing UMSs.

C Examine a Safety Model for a Industrial Process

The following is a multiple layers safety model for a manufacturing process plant, listed from narrow to broad scopes or from the low to high levels, i.e., item #a is the lowest level and item #h the highest. The higher level safety design activates when the low level design fails [10]:

a. to design the equipment, turning, milling, drilling, forging, die-casting, rolling, etc., and process plant to be inherently safe
b. process control to be designed with safety functions
c. procedure for and activation of alarms and operator intervention
d. safety shut down and interlock of affected entities
e. response mechanisms for fire and gas
f. containment system for the hazard
g. plant emergency response evacuation system
h. community emergency response evacuation system.

Each step contains an independent safety design, yet they are integrated to produce a coordinated safety operation.

We observe that system configuration is expanded and system complexity increases from the lower to higher levels. As a result,

a. Safety related missions or tasks become more complex.
b. The operating environments become broader and involve more entities. They may tend to be more dynamic and unstructured.
c. Higher levels of CAC provides for higher capability for safety.

VII. ADDITIONAL DOMAINS

A. Defense

UMSs are well suited for military types of operations. UMSs can replace soldiers in harm's way and can get themselves in extreme operational and environmental conditions. War fighting, surveillance, medical assistance in the field, logistic support, etc., are just a few of the fruitful areas for UMS deployment. These are also rich issues warranting the application of ALFUS.

B. Search and Rescue (SAR)

One of the major concerns in SAR would be the environment. Would it be accessible? Would it be safe for responders to approach? How is an environment or an environmental condition be described and conveyed to the decision maker so that the following issues, among additional, others, can be revolved: an appropriately composed and equipped Emergency Response Team [2] be dispatched, at a certain Point of Arrival, whether and what kind of Incident Support Team might be needed, and, possibly, under the command of a certain Federal Coordinating Officer.

EC levels might be used to identify particular environments used for robotic certification.

Efforts have also begun at NIST to establish the performance metrics for SAR robots by developing test arenas with various, adjustable levels of difficulty [11, 12].

C. Border Security

Variety in terrain and lengths in distance are among the challenges of securing the National borders. For the portions of the border that is difficult to traverse, ALFUS could be applied to characterize the levels of complexity, which could facilitate deploying UMSs with appropriate CACs. For the busy crossing ports, UMSs could help the safety related tasks such as baggage checking and identity verification. ALFUS could help characterizing the task complexity specifying HRI requirements.

D. Bomb Disposal

The ultimate concern for bomb disposal would be safety of human. Therefore, this is the type of task for UMS. ALFUS could help analyze the complexity of such an operation. The results could help optimize human assistance, including a safe operating condition and environment for the involved operator.

E. Standard Mission/Task Ratings

Skill ratings for human tasks are used [13, 14, 15]. It would be interesting to explore similar ratings for UMS. For a particular domain, a collection of tasks or a collection of typical scenarios that involves the combination of task, environment, and HRI can be rated for CAC. The information would be maintained in a database. When a situation arises that calls for the deployment of a UMS, the situation could be analyzed and the matching tasks or scenarios could be identified. The information could be used to efficiently deploy a capable UMS to handle the situation.

VIII. SUMMARY

Key concepts for the ALFUS Framework are introduced, with particular focus on the safety issue. A selected set of domains are analyzed for the applicability of ALFUS, including manufacturing, military, and homeland security. We discovered that ALFUS CAC should be helpful for indicating a robot's ability to conduct certain missions. We also discovered that each application domain may be unique that warrants expansion of the existent metric sets in ALFUS. Safety concerns also warrants expansion of the existent metric sets.

ACKNOWLEDGEMENT

ALFUS participants contribute to the Framework contents through workshop participation. Funding sources for this effort, throughout various stages of the Framework development efforts, include Department of Homeland Security, Science and Technology Directorate, Army Research Laboratory, and NIST.

REFERENCES

[1] *Autonomy Levels for Unmanned Systems Framework, Volume I: Terminology, Version 1.1*, Huang, H. Ed., NIST Special Publication 1011, National Institute of Standards and Technology, Gaithersburg, MD, September 2004

[2] ASTM International Standards E2521-07a and F 2541-06; http://astm.org

[3] Huang, H., et al., "Characterizing Unmanned System Autonomy: Contextual Autonomous Capability and Level of Autonomy Analyses," Proceedings of the SPIE Defense and Security Symposium 2007, Orlando, Florida, March 2007

[4] Huang, H., "The Autonomy Levels for Unmanned Systems (ALFUS) Framework--Interim Results," Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, Gaithersburg, MD, August 2006

[5] Huang, H., et al., "Autonomy Measures for Robots," Proceedings of the 2004 ASME International Mechanical Engineering Congress & Exposition, Anaheim, California, November 2004

[6] Ragon, M. and Jones, J., Decomposition of Mobility & Allocation of Functions to Autonomy Levels, Future Combat System Program Lead System Integrator Slide Presentation for ALFUS #14 and #15 Workshops, September 2006 and January 2007

[7] Bruemmer, D., et al., "Shared understanding for collaborative control," *IEEE Transactions on Systems, Man and Cybernetics, Part A* **35**(4), pp. 494-504, July 2005

[8] Barynov, S. and Hexmoor, H., "Quantifying Relative Autonomy in Multiagent Interaction," Book Chapter in Agent

Autonomy, Hexmoor, H., et al., Ed., Kluwer Academic Publishers, The Netherlands, 2003

[9] *Autonomy Levels for Unmanned Systems Framework, Volume I: Terminology, Version 1.1*, Huang, H. Ed., NIST Special Publication 1011, National Institute of Standards and Technology, Gaithersburg, MD, September 2004

[10] *Process/Industrial Instruments and Control Handbook*, McGraw-Hill Handbooks Series, McGraw-Hill Professional Publishing, NY, NY, 1999

[11] Jacoff, A., Weiss, B, Messina, E., "Evolution of a Performance Metric for Urban Search and Rescue Robots (2003)*,"* Proceedings of the 2003 Performance Metrics for Intelligent Systems (PerMIS) Workshop, Gaithersburg, MD, August 16 - 18, 2003

[12] Messina, E. and Jacoff, A., *"*Performance Standards for Urban Search and Rescue Robots*,"* Proceedings of the SPIE Defense and Security Symposium, Orlando, FL, April 17-21, 2006

[13] DTIC Accession Number AD0645054: Comparison Of Merited Grade And Skill Level Ratings Of Airman Jobs, http://stinet.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=AD0645054, Defense Technology Information Center, Fort Belvoir, VA

[14] http://www.trainingfinder.org/competencies/list_levels.htm

[15] http://www.us-army-info.com/pages/mos/skills.html

Evaluation of Autonomy in Recent Ground Vehicles Using the Autonomy Levels for Unmanned Systems (ALFUS) Framework

George T. McWilliams, Michael A. Brown, Ryan D. Lamm, Christopher J. Guerra,
Paul A. Avery, Kristopher C. Kozak, Bapiraju Surampudi
Southwest Research Institute
6220 Culebra Road
San Antonio, TX, USA
george.mcwilliams@swri.org

Abstract— Over the last few decades, the field of unmanned systems (UMSs) has begun to emerge into a variety of markets. The military has already deployed unmanned air, sea, and ground vehicles. Universities and other research institutions have developed semi-autonomous passenger vehicles that have driven on highways in the U.S. and abroad. The National Aeronautics and Space Administrations (NASA) has developed unmanned rovers that have been navigating the planet Mars for several years. Even the transit and commercial freight market has developed programs for unmanned vehicles research to help solve complex logistics issues.

In order to compare the capabilities of unmanned systems in such a wide variety of markets, the Autonomy Levels for Unmanned Systems (ALFUS) framework has been established in a series of workshops. While this framework is still under some development, it can be used in its current state to compare unmanned systems. In this paper, we highlight some of the major accomplishments made in the field of ground vehicle autonomy in particular. We then map the capabilities of these ground vehicles to the ALFUS framework and summarize the resulting trends that occur from this mapping.

***Keywords*:** *autonomy, unmanned systems, environment, human independence, mission*

I. Introduction

Over the last few decades, the field of unmanned systems (UMSs) has grown into a hot research topic, which has led to the deployment of a variety of UMS types in several different markets. In the U.S., the Department of Defense (DoD) has been the main catalyst in this research since a Congressional mandate requires that one-third of the nation’s combat vehicles are to be unmanned by 2015. The DoD has also benefited the most from this research by already deploying unmanned air, sea, and ground vehicles to military operations.

The area of ground vehicle autonomy has been a particularly interesting research field due to the fact that ground vehicles are used in so many different markets. The military uses unmanned ground vehicles (UGVs) for dangerous operations such as explosive ordnance disposal (EOD). Universities and other research institutions have developed semi-autonomous passenger vehicles that have driven on highways in the U.S. and abroad. The National Aeronautics and Space Administrations (NASA) has developed unmanned rovers that have been navigating the planet Mars for several years. Even the transit and commercial freight market has developed programs for unmanned vehicles research to help solve complex logistics issues.

Because of the complexity involved in comparing such unmanned and autonomous systems, the Autonomy Levels for Unmanned Systems (ALFUS) framework has been developed through a series of workshops. The levels of autonomy are decomposed into categories of mission complexity (MC), environmental difficulty (ED), and human independence (HI) [1]. The ALFUS framework includes terms and definitions, a detailed model and summary model for autonomy levels, and the guidelines and processes needed to apply the generic framework [2]. Although the ALFUS framework is still under development during the writing of this paper (Summer 2007), the current state of the framework can be used to compare the autonomy levels in the ground vehicle markets discussed above.

In this paper, we highlight some of the major accomplishments made in the field of ground vehicle autonomy by selecting and evaluating individual vehicles and programs. We then map the capabilities of these ground vehicles to the ALFUS framework and summarize the resulting trends that occur from this mapping.

II. Evaluation Process

A. Vehicle Selection

Because of the amount of research done in ground vehicle autonomy and the number of vehicles developed, it is challenging to choose which vehicles to highlight. For example, the military already has numerous UGVs, such as iRobot's *Packbot*, Remotec's *Andros*, and the Army Future Combat Systems' MULE, as well as several others. However, there have been a number of demonstration vehicles and programs that have captured the state-of-the-art at their respective times in history. These vehicles and programs were developed by a variety of researchers for a variety of different markets. This paper intends to divide these vehicles and programs into categories based on their use and their market. These categories will be discussed in general in the following sections. Then at least one case study of a vehicle or program will be described and that vehicle or program will be evaluated using the ALFUS Summary Model. The categories used in this paper include On-Road Passenger Vehicles, Vehicles used for Transit and Freight, Extraterrestrial Rovers, Off-Road Military Vehicles, and DARPA Grand Challenge 2005 Vehicles. The DARPA Grand Challenge vehicles could have been placed under the Off-Road Military Vehicles category, but it is given its own category since so many vehicles competed in the event and since it is the most recent program to take place. It should be noted that not all of the vehicles selected are UGVs by its strictest definition since some of them transport humans. However, they are all unpiloted when they are in "autonomous mode", and their autonomous capabilities can still be evaluated through the ALFUS framework.

B. Metrics and Tools

A spreadsheet-based software tool has been developed to automatically compute the autonomy levels based on the weights and the metric scores that users input [3]. However, since the tool has been developed, the ALFUS framework has continued to evolve and the metrics have continued to change. The method of evaluation used in this paper is based on the most recent tables of metrics. The MC, ED, and HI of these vehicles and programs under evaluation are matched up against these metrics and placed in the appropriate bin without using a tool to analyze the lowest level subtasks of these categories.

C. Disclaimer

The capabilities mapped to the ALFUS framework for the following vehicles and programs was subjective and was provided for illustrative purposes only. It is not the intent of the authors to conduct a detailed analysis of the systems discussed in the case studies and therefore there may be variability or inaccuracies in the analysis of the autonomy levels presented.

III. On-Road Passenger Vehicles

A. Analysis of Autonomy Levels in On-Road Passenger Vehicles

Many movies have been made that feature fully autonomous On-Road Passenger Vehicles and illustrate the dreams of a technological society that can move people at high speeds in vehicles that drive themselves. Researchers have taken on this challenge by attempting to solve parts of the problem over time. This section will analyze the application of the ALFUS autonomy levels for On-Road Passenger Vehicles and attempt to apply this framework to two vehicles that accomplished autonomous steering capabilities: the ARGO Autonomous vehicle and the NavLab 5 "No Hands Across America" vehicle.

1) *Mission Complexity:* The mission complexity associated with On-Road Passenger Vehicle autonomy is relatively simple: Move people from point A to point B. Mission complexity can increase if route optimization is desired or cooperative maneuvers with other vehicles (e.g. a fleet) are required. So far in On-Road Passenger Vehicles, high mission complexity has not been realized.

2) *Environment Difficulty:* The environmental difficulty of the mission associated with On-Road Passenger Vehicles can vary significantly based upon the ultimate application. In this case, the environment can range from closed-track, or protected lane navigation, to negotiation of complex urban roads with significant potential for interference by other vehicles and pedestrians.

3) *Human Independence:* Like the previous two perspectives, the HI metric has wide variability ranging from partial human control such as human throttle control to fully autonomous passenger vehicles.

While not yet realized through full implementation, the ALFUS framework can be used in the evaluation of UGVs used for On-Road Passenger Vehicles. The following describe two systems and an analysis as to their autonomy level determination.

B. Case Study: NavLab – "No Hands Across America"

The Carnegie Mellon University (CMU) NavLab 5 vehicle was a 1990 Pontiac Trans Sport that achieved 98.2% autonomous driving on a 3000 mile tour from Pittsburgh, PA to San Diego, CA called "No Hands Across America" [4]. The vehicle used the Rapidly Adapting Lateral Position Handler (RALPH) computer program. RALPH uses video images to determine the location of the road ahead and the appropriate steering direction to keep the vehicle on the road. The researchers actuated the throttle and brake manually. High level processing was performed on a Sparc LX class portable workstation. Low level steering motor control and safety monitoring was performed using an HC11 microcontroller. A color camera was used along with a differential GPS system to determine the vehicle's position and upcoming trajectory. Table I depicts the NavLab 5 vehicle aligned within the ALFUS framework.

TABLE I

ALFUS SUMMARY MODEL OF NAVLAB 5

	MC	ED	HI
10			
9			
8			
7		X	
6			X
5			
4	X		
3			
2			
1			
0			

C. Case Study: ARGO Autonomous Vehicle

The ARGO Autonomous Vehicle was demonstrated in 1998 on a 2000 km Italian tour and used only a stereoscopic vision system to perform lane-following and obstacle avoidance behaviors [5]. The vision system acquired pairs of 768x288 pixel grey level images at 25 Hz using a PCI Matrox graphics board. A 200 MHz MMX Pentium processor was utilized to process the images and perform autonomous steering capability by controlling an actuator on the steering wheel.

Acoustic and Visual warnings were given to the driver via onboard devices and displays. These warnings alerted the driver of unsafe distances to the leading vehicle or unsafe vehicle positions in the lane. The vehicle finished with an average speed of 90 km/h. 94% of the time the car was in fully autonomous mode, with the longest autonomous stretch being 54 km. Table II depicts the ARGO Autonomous Vehicle aligned within the ALFUS framework.

TABLE II

ALFUS SUMMARY MODEL OF ARGO AUTONOMOUS VEHICLE

	MC	ED	HI
10			
9			
8			X
7		X	
6			
5			
4	X		
3			
2			
1			
0			

IV. TRANSIT AND FREIGHT

A. Analysis of Autonomy Levels in Transit and Freight

Autonomy in transit and commercial freight has long been viewed as the panacea to complex logistics issues associated with the movement of people and goods. Numerous projects have been undertaken to evaluate the effectiveness of the application of autonomy in this domain; however, none have been widely deployed. This section will analyze the application of the ALFUS autonomy levels to transit and freight systems and attempt to apply this framework to two recent programs: the Carnegie Mellon University (CMU) Houston-Metro Automated Bus Project and the CityMobil – CyberCar Project.

1) *Mission Complexity:* The mission complexity associated with unmanned logistics and transit systems can vary depending upon the tasks to be performed. The transit of people or goods via an UGV between two points can be classified as the lowest level of mission complexity next to teleoperation. Should the mission include route optimization, end operations such as loading or unloading of freight, or multi-modal transit, then the autonomy level could increase significantly. Thus far in transit and freight, a high degree of mission complexity has not been realized.

2) *Environment Difficulty:* The environmental difficulty of the mission associated with logistics and transit can also vary significantly based upon the ultimate application. In unmanned systems, this can range from closed-track, or protected lane navigation, to negotiation of complex urban environments with significant potential for interference by other vehicles and pedestrians. Additional constraints for various forms of hazardous material transport can make the environment challenging.

3) *Human Independence:* Like the previous two perspectives, the HI metric has wide variability ranging from partial human control such as with automated throttle – human steering bus systems through fully automated people movers at theme parks, airports, and congested city centers.

The ALFUS framework can be used in the evaluation of UGVs used for transit and freight mobility systems. The following describe two systems and an analysis as to their autonomy level determination.

B. Case Study: CMU Houston-Metro Automated Bus

The Automated Highway Systems (AHS) demonstration in August of 1997 on a 12 km segment of Interstate 15 near San Diego demonstrated the feasibility of automated transit when CMU outfitted two Houston Metro buses with automated throttle, brake, and steering capability [6]. The RALPH software developed for the "No Hands Across America" vehicle was also used here. These buses were envisioned to traverse the Houston area High Occupancy Vehicle (HOV) dedicated lanes and were not intended to coexist with normally piloted manned vehicles. Table III depicts the CMU Houston-Metro Automated Bus aligned within the ALFUS framework.

TABLE III

ALFUS SUMMARY MODEL OF CMU HOUSTON-METRO AUTOMATED BUS

	MC	ED	HI
10			X
9			
8			
7			
6			
5	X		
4			
3		X	
2			
1			
0			

C. Case Study: CityMobil – CyberCars

The CityMobil Project is a European Commission sponsored mobility solutions program with urban demonstration deployments at Heathrow, Castellon and Rome. The CyberCar vision is to provide door-to-door on-demand service for the delivery of people and goods in congested urban areas [7]. Ultimately, the technology necessary for full vehicle autonomy negotiating urban pedestrian-rich environments will be coupled with fleet management systems to optimize routes and vehicle distribution. Currently, the operating environment is restricted to private property or dedicated/restricted lanes. Mission complexity does contain elements of vehicle-to-vehicle and vehicle-to-infrastructure communication with limited to no human interaction and no teleoperation. Table IV depicts the CyberCar aligned within the ALFUS framework.

TABLE IV

ALFUS SUMMARY MODEL OF CITYMOBIL CYBERCAR

	MC	ED	HI
10			X
9			
8			
7			
6	X		
5		X	
4			
3			
2			
1			
0			

V. EXTRATERRESTRIAL ROVERS

A. Analysis of Autonomy Levels in Extraterrestrial Rovers

Atmospheric and space vehicles have wide varieties of autonomous capabilities. Various organizations have deployed unmanned air vehicles (UAVs) to perform activities including monitoring, surveillance, communication relay, and military strike. In space travel, autonomy is critical where human performance is unreliable or in situations where communication latencies prohibit remote operation. This section will analyze the application of the ALFUS autonomy levels to planetary rovers: NASA's Spirit and Opportunity [8] [9], which are currently exploring the Martian surface.

1) *Mission Complexity:* The mission complexity associated with planetary rovers depends on constraints resulting from the rover's payload, the requirements of the science goal, and communication latency issues. These rovers must make decisions such as avoiding obstacles and traversing hazardous terrain. Poor judgments regarding hazardous terrain can jeopardize the whole science mission of the rover if a maneuver results in damage to the payload or loss of the vehicle.

2) *Environment Difficulty:* The environmental difficulty for planetary rovers results from terrain variations, soil composition, and weather. Obstacles consist of terrain features such as craters, rocks, and geologic formations. Steep grades and cliffs present additional hazards. Daylight patterns have an affect on the rovers' ability to maintain battery charge. Inclement weather can constrain a rover's operational capability.

3) *Human Independence:* Early rovers depended on significant human intervention. As the technology has progressed, the amount of human intervention has decreased. Activity planners depend on human input for the current rovers, but the route planners function mostly autonomously. Communication latency has forced the development of improved route planning.

The ALFUS framework can be used in the evaluation of UGVs used as planetary rovers. The following sections describe Spirit and Opportunity's systems.

B. Case Study: NASA Rovers: Spirit and Opportunity

In January 2004, Spirit and Opportunity landed on the Martian surface and began operations. They have exceeded their designed life. Part of the mission management includes software called Mixed Initiative Activity Plan Generator (MAPGEN), which depends on operator analysis and intervention. The route and path planning software consists of two major components: AutoNav and guarded moves. The first of these uses decision making algorithms to traverse between a start and goal point. The guarded moves uses manually specified maneuvers, but can prohibit actions deemed dangerous. The rover has VisOdom or visual odometry which uses stereoptical images to compute odometry for fusion with an Inertial Navigation System (INS). These capabilities allowed Spirit to traverse over 4.5 km with slopes of less than 20 deg.

TABLE V

ALFUS SUMMARY MODEL OF NASA'S SPIRIT

	MC	ED	HI
10			
9			
8			
7		X	
6	X		X
5			
4			
3			
2			
1			
0			

VI. OFF-ROAD MILITARY VEHICLES

A. Analysis of Autonomy Levels in Off-road Military Vehicles

Autonomy in off-road military vehicles has enjoyed a financial and technical focus that has only recently begun to be matched in other sectors. A number of successful projects in this area have demonstrated the effectiveness and usefulness of autonomous vehicle technology in the domains of force protection, reconnaissance, search-and-destroy, as well as others. This section will analyze the application of the ALFUS framework to two of these recent programs: DEMO III, and The Crusher.

1) *Mission Complexity:* The mission complexity associated with UGV off-road military systems can vary depending on the specific tasks the vehicle must perform, and the nature of its environment both in terms of terrain and hostilities. For military applications, both the tasks and environment can be unpredictably dynamic, particularly when one is affected by the other, such as when the mission task switches from reconnaissance to force protection due to a change in the environmental hostilities that are present.

2) *Environment Difficulty:* The environmental difficulty of the mission in a military application can be extreme due to both terrain, and hostilities aspects. The nature of a hostile environment also requires a UGV to distinguish friend from foe, and to make appropriate cost/benefit judgments from this information. For example, a UGV should not consider hostile action against an enemy target if that action will endanger members of its group, and conversely, should perhaps sacrifice itself if doing so would maintain the integrity of a human member of its own group.

3) *Human Independence:* Like the previous two perspectives, the HI metric has wide variability ranging from teleoperation to full autonomy. Military UGV programs exist within this full spectrum, with many of the vehicles able to operate under varying degrees of autonomy depending on the requirements of the mission.

The ALFUS framework can be used in the evaluation of UGVs used for off-road military mobility systems. The following briefly describe two such systems, and provide a high-level analysis of their autonomy levels within the ALFUS framework.

B. Case Study: XUV DEMO III

The Experimental Unmanned Vehicle (XUV) DEMO III project included participants such as the Army Research Laboratory (ARL), the National Institute of Standards and Technology (NIST), and NASA, as well as others. The XUV Demo III [10] is a vehicle designed primarily for mission support roles like reconnaissance, route trafficability, enemy detection, etc. With a powerful suite of sensors, including rear- and forward-facing vision, FLIR, RADAR, and LADAR, the vehicle is able to create a sophisticated world model of its immediate surroundings. Computational processing power and communications antennas enable the vehicle to fuse its sensor data to perform path planning in difficult terrain, locate and identify other machines and warm-body elements, and share this information with other members of its group. Vehicle to vehicle communications enable tactical behaviors such as platooning, and cooperative search. The platform can also be fitted with various weapons systems for direct action missions.

TABLE VI

ALFUS SUMMARY MODEL OF XUV DEMO III

	MC	ED	HI
10			
9			X
8			
7			
6	X	X	
5			
4			
3			
2			
1			
0			

C. Case Study: The CRUSHER

The Crusher [11] is a heavy duty autonomous vehicle that is able to travel at high speeds across difficult terrain thanks in part to its six large wheels and sophisticated suspension system. The vehicle is also symmetric, and if flipped upside-down, it will autonomously reconfigure its control and continue on its mission with a delay of little over a minute. The Crusher's suite of sensors is used to create terrain maps while traveling at high speeds. Telescoping sensors also provide wide-area terrain data while remaining largely hidden. The vehicle's controls also employ machine learning techniques. As the vehicle's strategy for avoiding obstacles is often to not avoid them, the design focus has been on increasing ruggedness of the platform to perform high-speed navigation on rugged terrain.

TABLE VII

ALFUS SUMMARY MODEL OF CRUSHER

	MC	ED	HI
10			
9			
8			
7	X	X	X
6			
5			
4			
3			
2			
1			
0			

VII. DARPA GRAND CHALLENGE 2005 VEHICLES

A. Analysis of Autonomy Levels in DARPA Grand Challenge Vehicles

While other UGVs may be said to legitimately score higher on one or even two of the axes, probably the best composite representation of UGV autonomy that is currently available (as of the summer 2007) is the set of vehicles that completed the 2005 DARPA Grand Challenge. Five vehicles finished the race that year and can be considered the pinnacle of UGVs, at least until DARPA's next challenge is completed later this year. This section will analyze the application of the ALFUS autonomy levels to the DARPA urban challenge vehicles (generically first) and then apply this framework to the winner of the challenge: Stanford's Stanley.

4) *Mission Complexity:* The Grand Challenge provided a straightforward mission to the competitors: drive over varying terrain along a specified course in the California desert in the shortest amount of time possible. Three issues drove the complexity in this mission: course length, uncertainty in terrain, and the competition element. The length of the course required substantial endurance from the hardware and software components in the vehicle. Varying terrain had to be accommodated as it was specified as part of the challenge. Finally, as part of a race, the mission required some consideration of the potential performance of other competitors.

5) *Environment Difficulty:* Since the race course was situated in the California desert, and therefore not trivial, it can be said to have been a milestone in UGV environmental difficulty. The general ruggedness of the terrain increased the challenge but commercially available, off-road-capable vehicles were sufficient to master the terrain. However, the drivers of the human-driven trail vehicles for each autonomous vehicle said it was oftentimes difficult to keep up with the autonomous vehicle on the rugged terrain. The 2005 course contained a mixture of well-maintained roads, dirt and gravel tracks, washboard and washouts, rutted tracks, poorly defined berms, open areas, choke points, obstacles, winding mountain roads, and steep drop-offs. Each of these poses challenges for UGVs but in all cases the environment and the obstacles in the environment were assumed to be static.

6) *Human Independence:* The vehicles were required to demonstrate substantial human independence to complete the challenge. Beyond the most obvious aspects of human independence, namely automatic route following and obstacle avoidance, the vehicles were required to automatically meet multiple objectives and resolve conflicts (e.g. simultaneous waypoint corridor adherence and obstacle avoidance), adapt to GPS dropout, and recover from any hardware or software failures that might otherwise prevent the vehicle from completing the race.

The ALFUS framework can be used in the evaluation of UGVs that participated in the 2005 DARPA Grand Challenge. The following describes the winning vehicle, Stanford's Stanley, and provides a brief analysis as to its autonomy level determination.

B. Case Study: Stanley

As the winner of the Grand Challenge, Stanford's Stanley represents a significant milestone on the path to full vehicle autonomy. Stanley completed the 132-mile course through the California desert in 6 hours and 53 minutes, and aside from two instances when DARPA race officials manually "paused" the vehicle, it traversed the entire course with no human intervention. The route for the course was specified in advance, so the primary challenge faced by Stanley was to identify the drivable terrain along that route. Five SICK laser scanners were used successfully to that end. In addition to the laser scanners, Stanley utilized a GPS and IMU to follow the specified route and estimate vehicle pose, and a color camera to identify drivable terrain beyond the range of the laser scanners. Because the global route was provided by DARPA, the path planning problem for Stanley was one of local obstacle avoidance rather than global route generation [12]. Table VIII shows where Stanley is estimated to fall within the ALFUS framework.

TABLE VIII

ALFUS SUMMARY MODEL OF STANLEY

	MC	ED	HI
10			X
9			
8			
7			
6		X	
5			
4	X		
3			
2			
1			
0			

IV. CONCLUSION

The field of autonomous vehicles or unmanned ground vehicles has evolved over the last 2 decades, primarily driven by government research funding. While the military and space programs pioneered early development, interest from the commercial transportation sector is just beginning. The three axes used in the ALFUS setting allow a comparison of current state of the art in autonomous vehicles and perhaps show areas of opportunity for planners. The assessments presented are subjective, although all bias has been consciously avoided. It is worth noting that it was very challenging to decouple the three axes without using an evaluation tool to analyze the lowest level subtasks of these categories. This challenge reached its height when evaluating vehicles that have not been directly involved in a public demonstration. For example, the lack of literature on these vehicles made it difficult to determine what type of mission it is able to complete, and what level of human dependence it would need to complete that mission. It was most challenging to decouple mission complexity and environment difficulty. There is significant research opportunity to further the state-of-the-art in both of these areas. The human independence accomplishments have been excellent across all markets such as military, space, mass transit and hobbyists. However, this is most likely a result of lower mission complexity and environmental difficulty.

The three metrics of MC, ED and HI are summarized in Figures 1, 2 and 3. The military is leading the research front in mission complexity while the hobbyist/academic community achieved a high degree of human independence.

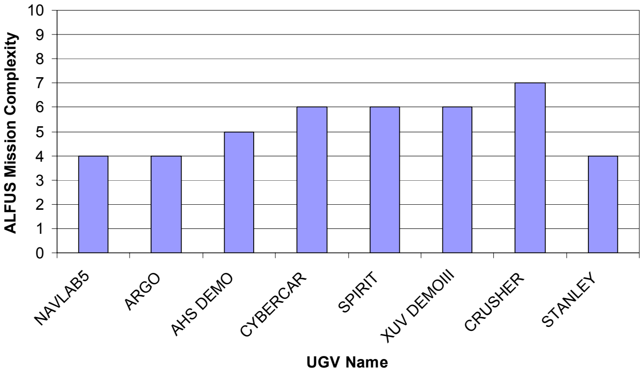


Fig. 1 Average ALFUS Mission Complexity has been low in the UGV research arena

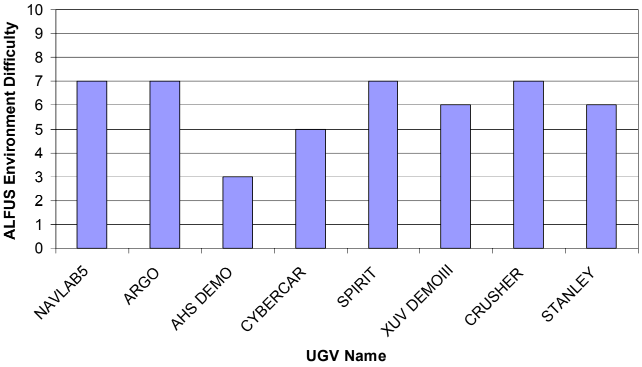


Fig. 2 ALFUS Environment Difficulty has remained steady

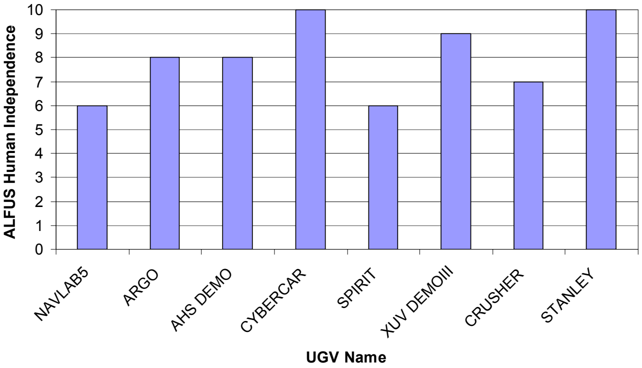


Fig. 3 UGV research community has excelled in ALFUS Human Independence – the most visible norm of autonomy

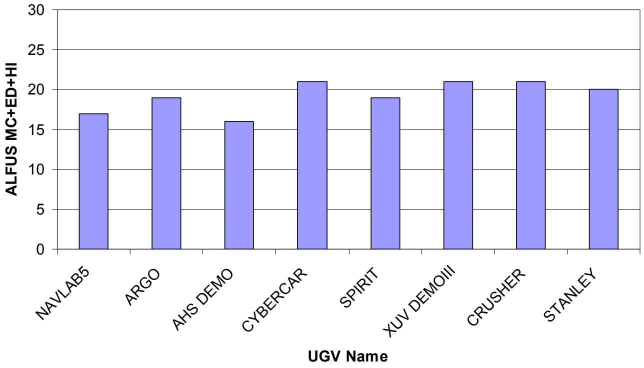


Fig. 4 The sum of ED, MC and HI is very close for all vehicles

The sum of the capabilities of every vehicle is very close as shown in Figure 4. While the three axes can vary based on application area needs, it is worth noting challenges in one direction usually necessitate simplification on other directions. An ideal vehicle with rating of 30 may perhaps be needed for autonomous military vehicles carrying human cargo in a hostile urban setting. An urban autonomous vehicle in civilian applications may never need a mission complexity level of 10.

Acknowledgement

The authors would like to thank the ALFUS Working Group participants. In particular, the following participants offered guidance during the completion of this paper:

Hui-Min Huang, NIST
Elena Messina, NIST
Charles Bishop, Boeing

References

[1] Huang, H., Albus, J., Messina, E. "Toward a Generic Model for Autonomy Levels for Unmanned Systems (ALFUS)," Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, 2003.

[2] Huang, H., Pavek, K., Novak, B., Albus, J., Messina, E. "A Framework For Autonomy Levels For Unmanned Systems (ALFUS)," Proceedings of the AUVSI's Unmanned Systems North America, 2005.

[3] Huang, H., Pavek, K., Albus, J., Messina, E. "Autonomy Levels for Unmanned Systems (ALFUS) Framework: An Update," Proceedings of the 2005 SPIE Defense and Security Symposium, 2005.

[4] Carnegie Mellon University (July 1995*), Carnegie Mellon Researchers Will Prove Autonomous Driving Technologies During a 3,000 Mile, Hands-off-the-Wheel Trip from Pittsburgh to San Diego,* Press Release.

[5] Broggi, A., Bertozzi, M., Conte, G., Fascioli, A. *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*. World Scientific 1999, ISBN 981-02-3720-0.

[6] Thorpe, C., Jochem, T., Pomerleau, D. “The 1997 automated highway free agent demonstration,” IEEE Conference on Intelligent Transportation Systems, pp. 496-501. 1997.

[7] www.cybercars.org

[8] Leger, P.C. et al. “Mars Exploration Rover Surface Operations: Driving Spirit at Gusev Crater.” 2005.

[9] Ai-Chang, M. et al. “MAPGEN: Mixed Initiative Planning and Scheduling for the Mars Exploration Rover Mission,” IEEE Intelligent Systems. 2004.

[10] Shoemaker, M.C., Bornstein J. “Demo III Program: A Testbed for Unmanned Ground Vehicle Autonomous Navigation,” IEEE Symposium on Intelligent Control, 1998.

[11] Carnegie Mellon University (April 28, 2006). *Carnegie Mellon’s National Robotics Engineering Center Unveils Futuristic Unmanned Ground Combat Vehicles*. Press Release.

[12] Thrun, S., Montemerlo, M. et al. “Stanley: The Robot that Won the DARPA Grand Challenge,” *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

A Methodology for Testing Unmanned Vehicle Behavior and Autonomy

Gertman, D. I.[1], McFarland, C.[2], Klein, T. A.[3], Gertman, A. E.[4], and Bruemmer, D. J.[1]
[1]Robotic and Human Systems Department, Idaho National Laboratory (INL)
[2]Johns Hopkins University, Department of Engineering, [3]Oregon State University, College of Engineering
[4]Albertson College of Idaho, Department of Mathematics
Contact: david.gertman@inl.gov, david.bruemer@inl.gov

***Abstract*—**This paper discusses approaches developed at the Idaho National Laboratory (INL) for quantifying and analyzing the performance of human-robot teams across different domains. These methods reflect experience and insights gained from previous INL experiments that have focused on landmine detection and marking; mapping and localization for robot positioning, mobile manipulation for explosive ordinance disposal (EOD), radiation characterization , and urban search and rescue operations. An overarching goal of this work has been to enhance our understanding of how the robot, the control and display interface, the task context, and the human contribute to or hinder mission success. Our approach to performance measurement was developed in concert with the iterative design cycle of our intelligent robotic control system, the *robot intelligence kernel (RIK)*.[1] In extending and refining the RIK for various applications, three factors key to holistic human-robot performance assessment were identified: comprehensive planning; the inclusion of end users in the design and performance evaluation phases of the study; and combining automated data collection with subjective measures. The paper discusses lessons learned in developing and applying performance metrics and provides a brief overview of measures that we are currently using to support the assessment of autonomy. In particular, the paper emphasizes the application of these metrics to behaviors for complex and potentially dangerous missions.

***Keywords*:** *Idaho National Laboratory, Unmanned Vehicle, Performance Metrics , Autonomy, Data Logging*

I. Introduction and Background

During the last few years, the Idaho National Laboratory (INL) has been conducting experiments to iteratively assess and improve a suite of intelligent behaviors called the *robot intelligence kernel* (RIK). The RIK is a collection of robot behaviors and interface display methods which allow the user to choose the appropriate level of autonomy and interface perspective for effective performance of the task at hand. Throughout these studies, an approach to experimental design and data analysis has evolved that emphasizes the need to facilitate, understand and predict human robotic interaction (HRI).[1,2]

Although these studies vary in terms of users and task domains, the functional allocation of tasks between robot and human has always been a focus. By using on-board robot intelligence it is possible to free up the operator to attend to other duties; however, the goal is not to create full-autonomy, but rather to understand how the human and robot work together to produce optimal performance. In the course of HRI laboratory and field studies conducted at the INL with a variety of robot platforms and missions [2,3], three key factors were determined to be important: comprehensive planning including experimental design, selection and inclusion of end users, and an approach to data collection that recognizes combined contribution of objective and subjective data. One of the outputs of the planning and experimentation cycle is the development of performance metrics. Some of these metrics such as operator cognitive workload, error, and frustration focus on the user. On the other hand, others metrics such as the number of collisions, the distance traveled or sensor performance in terms of detections and false positives focus on the robot. Finally, some of these metrics apply to the interface including communication bandwidth, joystick usage, and automated logging of a wide variety of control inputs. In addition, we have developed and used application-specific performance metrics for urban search and rescue (USAR); urban reconnaissance; robot mapping and positioning; landmine detection; and radiation source detection and localization. In performing these studies, we have reached the conclusion that neither objective nor subjective data alone are sufficient to provide an in-depth understanding of performance and performance issues.

Under the auspices of the Office of the Secretary of Defense (OSD), the Joint Ground Robotics Enterprise (JGRE) Technology Transfer Program the INL and the Space and Naval Warfare Systems Center in San Diego, the INL has developed components of perception, behavior, communication, and world modeling. The RIK uses these components to support five separate control modes: fully teleoperated, safe, shared, collaborative tasking mode and fully autonomous. In the teleoperated condition the human user has navigational control over the robot using a high bandwidth video feed. In safe condition the robot is able to protect itself from running into an obstacle while the user is still under full navigational control. In shared condition the robot is responsible for driving and the operator may provide directional cues intermittently. In the collaborative tasking mode, a variety of map-based "drive by intent" tools are used such as a visual hotspot which tells the robot to provide video of a region or the target icon which tells the robot to path plan and drive to a given location. In autonomous condition the robot has an end-to-end mission and accepts no user input. This paper discusses how these different modes of control have been developed and evaluated in a spiral cycle. Of particular interest is the way in which the experimental design and use of metrics has changed throughout this process. These changes have been due to the increasing capabilities and reliability of the robotic system as well the lessons of experience. More to the point, this paper indicates that different experimental methods and metrics must be used depending on the level of technology readiness, robustness and reliability.

II. Background Research

Before rigorous testing with human participants, it was necessary to measure the effectiveness of the actual robot performance. Across all modes of control, the RIK uses a positioning system that incorporates a Simultaneous Localization and Mapping (SLAM) method developed by the Stanford Research Institute [4, 5]. A measure of the overall positioning accuracy and reliability was a necessary prerequisite to other more in depth experimental questions. This testing and verification was accomplished by the Remote Sensing Lab in Nevada which used a laser theodolite system to track actual robotic movement in three dimensions. This testing showed that the robot was able to localize to within + / - 5 cm while moving for long periods of time within an open area of approximately 1500 square feet. In addition, an innovative ground truth system was used to test the ability of the robot to follow preset waypoint paths within a dynamic environment. This method was to place a black tape strip on the ground that could be sensed by a downward facing light sensor in the center of the robot. This sensor monitored when the robot's center was positioned over this two inch wide tape. The black tape was in place in order to match a "virtual rail" created as an AutoCAD drawing. The INL GUI software used this to create a waypoint path plan which was then transmitted to the robot. The robot then used its behaviors to follow this plan. The experiment was used to metric the percent of time that the robot was able to stay over the black tape when the robot traveled at various commanded speeds. Although the robot could not visibly be perceived to be off of the "virtual track," the experiment showed that the robot remained on the black tape approximately 75% of the time.[6]

With issues of positioning addressed, the next step was to assess the performance of the robot behaviors for obstacle avoidance and the value of shared control. In one of our early studies, 107 participants drawn at random from attendees of the INL annual community exposition in 2004 were given the task of locating sources (a simulation for finding people) [7]. In this study, each of these novice participants had 60 seconds to locate as many of five different items as they could. This USAR study focused jointly on the usability of the interface, the performance of the users and the performance of the robot behaviors. This was a preliminary study focused on understanding the benefits of robot behaviors and fundamental issues surrounding use of autonomy. The study allowed us to demonstrate that the use of behaviors was valuable in terms of task efficiency, providing a statistically significant increase in the number of objects found across ages and gender. However, despite these objective performance results, one of the insights from this study was that the combined use of subject feedback in conjunction with performance metrics provided the best overall understanding regarding performance during the experiment.

Another study, performed in 2004 was focused on measuring the benefits of mapping. The experiment introduces a virtual three-dimensional (3-D) map representation that supports collaborative understanding of the task and environment. The goal of the 3-D display is to provide a workspace for collaborative understanding between the human and robot. The virtual 3-D component has been developed by melding technologies from the INL [8], Brigham Young University [9], and Stanford Research Institute (SRI) International [4,5]. When used in place of video, the 3-D map reduced operator workload and navigational error. By lowering bandwidth requirements, use of the virtual 3-D interface was also shown to enable long-range, non-line-of-sight communication [10].

Finally, we consider a previous study focused on a sensor payload. In particular, this study was focused on detection and marking of landmines. In 2005, the INL, the US Army Maneuver Support Center (MANSCEN), and the US Army Test and Evaluation Command (TECO) conducted an experiment to test the INL Autonomous Robotic Countermine System (ARCS). The robot that the test used was developed at Carnegie Mellon University (CMU). The robot system employed ground marking equipment developed by the SPAWAR. In this study as well, the planning was comprehensive involving many entities. The study was performed out at the INL unmanned aerial vehicle (UAV) airstrip on an unimproved dirt road. To test the system, six anti -tank mines were buried in the road at depths between 6 and 8 inches. For this experiment, four elements were used to metric the ARCS including: finding the landmines, marking the landmines, reporting the landmines to the control interface, and marking the proofed lanes for dismounted troops to move through. To measure the performance of the robot for the experiment, the Autonomy Levels for Unmanned Systems (ALFUS) ratings were used as an additional metric. The ALFUS ranking was developed by the National Institute of Standards and Technology (NIST). In this ranking there are three scales, human intervention, environment complexity, and mission complexity to indicate the overall level of autonomy exhibited by an unmanned system. The higher the average ALFUS score the greater the autonomous performance of the robot [11]. The use of ALFUS was an important recent milestone in our approach to applying metrics. Figure 1 presents an interface screen shot available to operators during the experiment that show fusion of GPS, map and real time aerial imaging. The results of the real-world experiment showed that the proposed autonomous robot countermine system accurately marked, both physically and digitally, 124 out of 131 buried mines in an average time of less than six minutes.

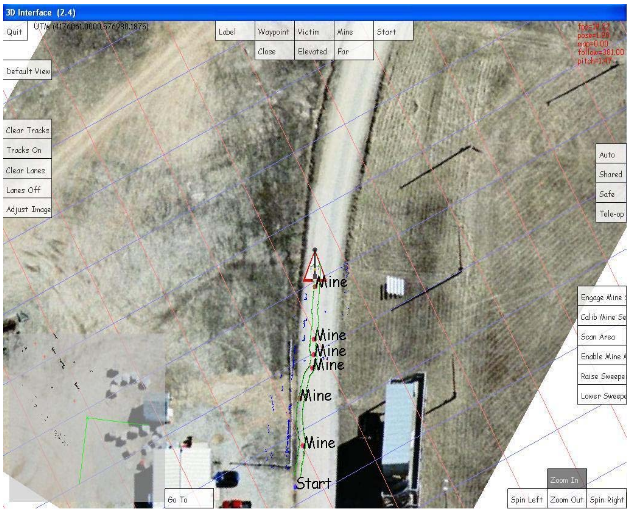


Figure 1. Interface screen shot for countermines

Across each of these experiments, determining what to measure, who to include as part of the subject pool, how to properly balance laboratory and field issues in experimental design, the level of robot autonomy and the design of the display and control interfaces were large issues. In the remainder of this paper we address these factors while reviewing performance metrics. We develop further the argument that the approach to performance metric development and usage should be a flexible one, where meaningful measures depend on the level of technology readiness. Moreover, we believe that studies and the metrics that accompany them can build systematically on one another. Lessons learned from each experiment can be used to guide the development and execution of the next phase of study.

III. Approach

The goal of understanding the performance of the robot and human; the usefulness of the control and display interface within the mission context; and subtleties regarding the linkage between preference and performance can be identified to provide insights for fielding systems, training, and future collaborative design. The following sections present our approach regarding the conduct of field studies. This includes: subject selection; comprehensive planning and study design, data logging, and employing subjective and objective measures to assist in tabulating, interpreting, and characterizing human-robot performance.

IV. Subject Selection

Pedahazur and Schmelkin (1991) describe many of the pitfalls in research related to selection and handling of study participants. [12] Although the main thrust of their arguments center around complications and threats to validity arising from differences in assumptions between subject and researcher including perceptions regarding the immediate environment, their admonishment to select the right subjects is aligned with our way of thinking. Novices and experts do not perform alike. As with Pedhazur and Schmelkin (*Ibid*), we also have been concerned with how subjects perceive the task, understand the context for task performance, and harbor assumptions different than those of the experimenters. Anecdotally, years ago, Israeli students were given a Rorschach to respond to. The experimenters did not know that the students had mistakenly believed that this was a test of imagination and that the best strategy for doing well was to produce as many divergent and obscure answers as they could within the time permitted. It wasn't until a subject debrief was held that the experimenters learned of the students' assumptions. Had they not participated in a debriefing session, the students would have been diagnosed with severe personality disorders.

Also, we have found that end users can help during the planning process to make scenarios more relevant and to highlight important task behaviors; they also participate in and provide feedback during the experiment. An important issue is that the experiment design and debriefing must carefully distinguish between the various kinds of end users and the different levels of user experience. All so called "subject area experts" are not created equally. In fact, in a recent study with radiological hazard detection, consider that three different kinds of subjects were involved including personnel with robot operation and dirty bomb training; those with only dirty bomb experience and those subjects with general training with radiation detection. In the experiment, treating these users as if they were all the same would have been an unfortunate mistake since certain features of the robot behaviors and interface were used very differently and, in fact, user during analysis experience turned out to be a statistically significant factor.

V. Planning and Study Design

In terms of evolving a comprehensive planning process, user insights and operational experience are key to determining the appropriate level of realism, difficulty, and task conditions found in the field. Every successful study requires a multidisciplinary team effort. In our recent dirty bomb experiment [in publication] different types of experts including radiation health and facility planning experts were brought into the planning process. Chemical, biological, nuclear, radiological and explosives (CBRNE) trained military explosive ordinance disposal (EOD) personnel were included in the early planning stages of the experiment to develop experimental goals consistent with systems that are operating in the field. In formulating the study design, we have used many of the study design features as practiced by the behavior sciences research and development communities. We have coupled this to traditional usability test paradigms. Even though the technology in robotics is dramatically changing, we have attempted to contain the complexity that always seems inherent in field studies. Realizing that obtaining a large number of subjects from a user population is always difficult, we have used randomization and counterbalancing in our designs. We have also employed computer-based data logging to reduce labor requirements and to improve reliability. All subjects receive "hands on training" and are encouraged to provide feedback during structured sessions. Inherently, fielded study designs will always reflect a number of real-world challenges including the availability of subjects and facilities; technical challenges associated with moving robot intelligence from one platform to another, and requirements for integrating improvements in sensor technology or to implement new behaviors.

VI. LOGGED DATA

One of the basic goals of any experiment is to generate meaningful findings from basic metrics of success. In past experiments, the approach to data collection has been to attempt to figure out *a priori* which data would reveal significant results. During experiments, various equipment problems often ensued with the result that some portion of dependent variables were not recorded across all subjects. Likewise, because of human error in transcription or observation some dependent measures were not available for analysis. Lastly, sometimes measures are selected that fail to produce significance. Often experimenters know that they should be either measuring something differently or something different.

Early on we decided that the answer to these pressing problems was to record video and screenshots of the experiment that could be retrieved after the experiment was finished, but due to human error and inconsistencies, this was also often lacking key pieces of information. Even with video taping, the general metrics used in these studies were often simplified down to broad aspects of performance such as the degree to which the task was completed, and the time in which the task was completed [2]. Although we believed that there were other meaningful data to report, there seemed to be no easy way around this predicament.

Facing the continual problem of the lack of significant results that were in stark contrast to subject self-report and observations of the experimenters, and believing the cause to be directly the result of attempts to draw conclusions from a limited data collection approach, we sought to determine a means by which to implement a more finely tuned approach. Harbour, et al (2006)[13] state that performance improvements for high technology systems often come in small increments, thus more finely tuned performance-based metrics are needed to better analyze the differences between systems. We now use "data logging" as a way of further measuring smaller aspects of performance, such as joystick bandwidth, and joystick vibration and others [14]. As a *caveat*, not every small performance measure is going to prove to be successful, thus the flexibility provided by data logging described below is key to being able to evolve the right measures.

A) Development of data logging capability

Data logging stores the communications stream between the robot and the user interface automatically. By logging the communication stream, researchers are able to capture and review the experiment experience from multiple perspectives. Being able to extract new metrics and variables from the saved communication stream allows unforeseen issues and questions to be addressed. In addition to helping to create more objective metrics for the system, data logging also helps to prevent the human error found in earlier experiments.

B)Data logging as a two edged sword

Data logging also carries certain potential problems. One such problem is that the abundance of data can be overwhelming. This problem may be remedied by reducing the search space for dependent measures to those things that are theoretically meaningful andr allotting sufficient time for data analysis. Another potential problem that results from data logging is the potential for errors within the computer. These errors can be remedied by thoroughly testing the system before the actual experiment takes place. The data files can be subject to various tests regarding data range and standard deviation and any outliers reviewed to determine whether problems are present.

By utilizing data logging, useful variables and metrics can be recorded in a more precise manner, and data that was previously difficult to capture is now easily retrieved. Some useful data that we have identified through data logging experience that would be otherwise unavailable or costly to determine includes: robot initiative, percent time active, and operator confusion measures. Robot initiative reflects the number of times that the robot had to protect itself. Operator confusion represents the number of times that the operator tried to force the robot to continue to follow a path after he/she had previously been told that the robot was unable to follow that path. Percent time active is indicative of the amount of time logged that the operator was using the joystick or mouse to control the robot divided by total time. Other logged measures include joystick usage in terms of back and forth versus side to side movement and average time between actions.

For example, '% of time active' is a relatively new metric; in the past we used joystick movements as a way to record this information. Recently, it was found that since there were some instances in which a joystick was not used as a preferred strategy, that joystick movement was no longer an adequate measure of performance. We hadn't decided to use "time active" *a priori*. In the past, without detailed data logging, we would have been forced to wait until the next experiment to evaluate this metric and determine whether it was an adequate correlate for workload and localization. This "% of time active" measure was synthesized from computer generated data logs and showed results similar to the video summaries. As it turned out, the users spent much less time interfacing with the system in target mode than they did in other modes (F =.0001, df=2, Tukey significant at p <0.05, for target versus joystick, and target versus joystick and map). Thus, we were able to find a "hard measure" for workload only after being informed by softer, qualitative measures. Had we limited ourselves to simple quantitative measures, these differences would likely have not been discovered.

Data logging is important because it allows for a number of hard measures to be collected that can be further analyzed, better allowing for a more accurate depiction of the many dimensions of an experiment. Thus, in our most recent experiment, automated data logging was used to develop measures for robot initiative, operator confusion, and active time measures. These measures allowed us to distinguish between levels of performance in instances in which almost all of the subjects completed the task in similar periods of time. By generating these measures of robot initiative, operator confusion, and active time from logged files it was straightforward to prove the usefulness of data logging. The art is in determining which of the many measures are most useful in terms of predicting user performance. Data logging is best served when used in conjunction with additional subjective and objective measures that constitute a holistic approach to data collection described elsewhere in this paper. Later, in sections below we present data logging measures and discuss their value in aiding human- robot performance characterization.

Robot initiative per square foot was another variable generated by data logging. Robot initiative as an absolute number was non-significant whereas initiative adjusted for the square foot of distance traveled was significant. The insight to control for distance first surfaced during subject interviews. Figure 2 presents these findings excerpted from the dirty bomb study; note that the mean target mode condition is associated with lower initiative than the other two conditions.

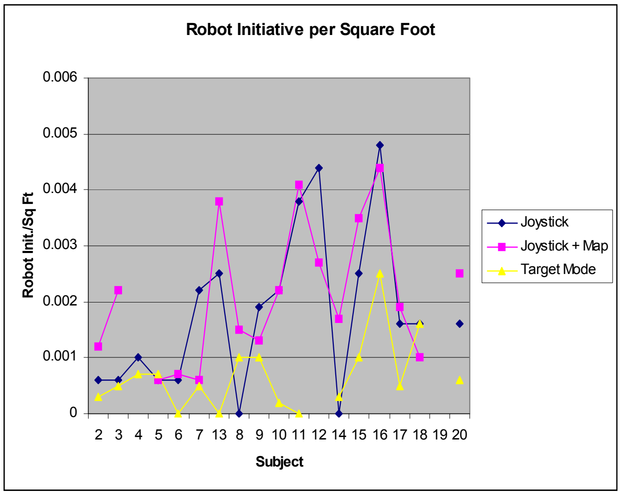


Figure 2. Robot initiative as a function display mode

VII. Considering Objective and Subjective Measures

Good experiment planning provides a structure for data analysis by selecting potentially valuable performance metrics to record. Finding crucial performance metrics to aid in characterizing and understanding the nuances of human-robot team performance in relation to mission objectives requires an *a priori* understanding of the human-robot teaming experience. In the most extreme case, an almost infinite number of independent variables with an infinite number of interactions can be conceived. This list is reduced by review of the mission objectives, task procedures, military doctrine, and experience of personnel in conducting similar missions. Selection and analysis of a limited number of planned metrics reduces the measurement challenge to something desirable, however, to effectively establish the relationship between important elements and performance gains across multiple scenarios the right mix of measures must be considered. This means considering objective and subjective measures. It is preferred to identify a select group of measures that can easily be either combined or reduced to cover the problem space.

Through statistical analysis, objective data often yield information regarding different treatment conditions or levels of independent variables thought to be important. To some extent, they have proven useful in heightening our ability to characterize and predict human-robot performance. Notionally, objective data give a system-centric view of the experience, where as subjective data provides a human-centric view of the experience. Quantification of subjective data allows researchers to perform similar analysis to that which is performed with objective data, but from a human centered perspective. No matter how well a system performs, it is still possible for the subjective experience of the operator to fail to reflect objective performance gains. Human-centered analysis allows researchers to understand the perspective of operators and their assessments regarding the system and system performance. Thus, in our studies we use subjective rating, debrief, and interview data. We have found the debrief sessions involving end-users are often the most revealing in terms of what robot behaviors are the most valuable.

A) Finding appropriate objective/quantitative metrics with help from subjective data

The appropriate objective data are often not obvious. In early INL studies, basic statistical procedures were used on the data logged during the experiment. The computer-based data logging that we now conduct consists of the computer keeping track of subject actions, system status, and experimenter input during the administration of the experiment. In one instance, using counter balancing methods, it was possible to preclude learning effects from skewing the data. We also performed qualitative data analysis of three data sources: video, researcher journal comments, and debriefing interviews, that indicated that source location had an effect on subject performance. Through interviews it was determined that the measure of the area to be searched was an influential variable, this analysis determined that the square footage associated with different search areas was the primary cause of performance discrepancy between runs. Normalization of the a number of quantitative data with respect to square footage provided much cleaner differentiation between the groups and between the conditions while removing the dependence of findings on source location. The effect of statistically removing the influence of distance from the logged data was to indicate a potentially important, i.e., statistically significant finding that otherwise would have been lost.

B) Developing an understanding of a system level result

Researcher comments during the experiment and post-experiment observations of videotapes have proven useful adjuncts to objective data. For example, during a recent radiological localization and mapping experiment, there were striking differences between user perceived physical workload and stress as measured by the NASA task load index (TLX) [15] and researcher's assessment of that same workload using postural differences as an indirect measure of workload. The postural differences were sensitive to robot autonomy conditions, when users in the "video only" mode attempted to accomplish the navigation and localization tasks there was visible strain in their posture as the users curled their back to place their face as close to the monitor as possible. Those same users were seen in a more casual posture while operating the system in target mode, which indicates a lower level of physical workload and stress. This result was used to aid in understanding and interpreting other subjective and objective data.

C) When subjective measures can be misleading

Subjective/qualitative data provided additional information about the discrepancy between workload ratings and actual workload as determined by performance measures. During the debrief interviews it was revealed that participants subjectively anchored their workload ratings based on their own experiences. This had not been a major problem in the past, subjects often use their own frame of reference. Although variations in rating scale design such as the behaviorally anchored rating scales (BARS) exist and have been used in a multiple settings such as clinical research and a variety of disciplines including management science we did not consider them to be necessary. However, the frame of reference used by many subjects was in-theatre combat!!! In comparison, all of our conditions were low in stress, complexity, and time pressure. In our review of the rating data, all of the modes were found to be very light on workload, which *washed out* any differences. Because during the interview and debrief process we learned how subjects were anchoring their responses, we were able to find look for more appropriate "Hard measures" of workload that were sensitive to levels of autonomy and display mode.

D) When the most "objective" measure is not the best

In our recent radiological localization, i.e., dirty bomb scenario, the ultimate goal was to compile subject scores communicating the accuracy of radiological source locations during experimental trials. One of our assessment tools for task completion required the participants to mark those locations on a floor plan of the test facility INL researchers decided against a completely quantitative and objective measurement strategy, i.e., just measuring the distance from the subject's mark to the actual source, in favor of a measure that retained an element of contextual content. This measurement strategy took into account not only the numeric distance, but combined this with the operators' grasp of both the global and local environment. These data were converted to a ten point scale which was chosen to match the breath of possible answers and uncertainty inherent in the map marking process (the maps themselves were not perfect). Interviews and discussions with EOD experts suggested that landmarks on the map were the key contextual requirement for location communication between robot operators and other human team members. The global portion of the localization was evaluated by subtracting from the score the number of landmarks (corners, pillars, walls, doors,) between each true source location and the location marked by the subject. The local portion of the metric was evaluated by cataloging the four most important landmarks at each actual source location and subtracting the number of cataloged landmarks which were not present at the location marked . This metric is more complex, yet retains more of the user definition of good localization.

The results, as shown in, Table 1 convey the benefits of the target mode for the localization task as a function of user experience. For the civilian support team (CST) and explosive ordinance disposal (EOD) subject populations, the per cent failure declined dramatically as the level of autonomy increased.(The joystick mode is associated with the least autonomy followed by joystick and map. The target + map mode represents a nearly fully autonomous condition) For nuclear engineers (NE), the group least familiar with robots and emergency response procedures, the level of autonomy failed to have a positive influence upon the per cent failures.

Table 1: Results of source localization via map analysis

		Joystick	Joystick + Map	Target + Map
EOD	% Excellent	57.1%	66.7%	100.0%
	% Fail	28.6%	0.0%	0.0%
CST	% Excellent	42.9%	50.0%	50.0%
	% Fail	42.9%	16.7%	0.0%
NE	% Excellent	50%	50%	75.0%
	% Fail	25.0%	50.0%	25.0%

*Excellent scores range from 90-100
**Failed scores range from 0-60

Within the EOD group, the increase in "% excellent" source location scores improved with each corresponding increase in robot autonomy level. One of the most interesting findings is not just the metrics themselves, but the realization that across several of the metrics, it seems that one of the most important advantages of autonomy is that it allows us to place tight bounds on performance. Essentially, for a number of different measures, the standard deviation is much tighter in instances where a higher level of autonomy is utilized. In the dirty bomb study, the chart presented in Figure 3 shows this to clearly be the case for the ability to localize the source.

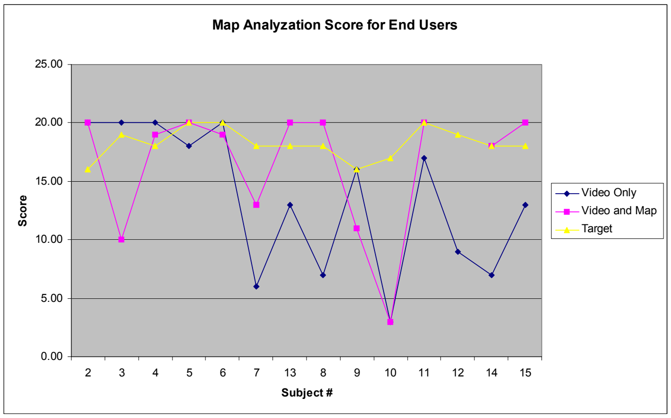


Figure 3. Accuracy scores for ability to localize sources by subject

E) *Determining a composite performance measure*

As we reviewed the various measures we have tried to consider which mix of metrics might have the best predictive ability. Below is our first attempt to produce a composite metric integrating objective and subjective measures of robot behavior developed over the course of multiple studies. (See Appendix A for this equation). Data regarding the ranges, means, variances, and assumptions for the measures are the subject of another paper. Also, this metric is to be refined over the course of additional planned studies. Table 2 presents the scores obtained when applying the metric to recent human-robot performance data.

Table 2. Composite performance scores for group experience and robot autonomy conditions

	Joystick	Joystick + Map	Target + Map
EOD	0.723	0.728	0.838
CST	0.623	0.645	0.796
NE	0.618	0.603	0.650

Note that when the composite metric is applied that the best mean task performance is obtained for the most experienced group in the highest. i.e., target + map, autonomy mode. When compared with the other less experienced users, the EOD user group had the highest mean performance scores. Finally, the highly automated target mode was associated with better performance than either of the two other conditions.

VIII. Summary and Discussion.

Over the past few years INL has developed an integrated approach to conducting field studies that emphasizes the discovery and development of metrics key to human-robot system performance. Comprehensive study planning, involving end-users in design and metric development, and the use of objective and subjective measures are part of the field study process. Laboratory studies can be useful in establishing systems performance, debugging equipment, and establishing general bounds for performance, however, field study experiments have a direct realism and relevance for end-users. Ideally, as the technology readiness level of the systems increase, a transition is needed from laboratory studies which are more controlled and precise to more open ended field studies. The evolution of testing methods described in this paper illustrates one attempt to accomplish this transition over the course of several years.

In many studies, we have noted large performance and preference differences among less expert and more expert end users. Although it is more difficult to enlist the participation of end-users, particularly those with qualified robotic experience, the data, insights, and lessons learned are significantly more valuable in guiding decision makers as well as future developments for control strategies and behaviors. The dependent measures identified in the studies reviewed in this paper: operator confusion, robot initiative, and % time active, would not have been developed without the aid of data logging leading dependent measure synthesis and the incorporation of end user comments. Further, there is some evidence that grouping users based on the level of their experience can unearth important trends and insights. Simple dichotomies such as user-novice can overstate findings, collapsing these groups for analysis purposes may prove to be worse.

Understanding the issues of robot autonomy can not be accomplished in a single study, it must be done over a period of years where nuances relating to the interplay between the human and robot behaviors can be looked at singly or in conjunction with one another. Our approach has always been to ask "What is the fundamental benefit of additional behaviors as opposed to applying our efforts to optimize a single, specific application?"

REFERENCES

[1] D. J. Bruemmer, D. I. Gertman, C. W. Nielsen, D. A. Few, and W. D. Smart, Supporting Complex Behaviors with Simple Interaction Tools, Advances in Robotic Systems, edited by Aleksandar Lazinica, Vienna, , in press, expected December 2007.

[2] D. J. Bruemmer, R. L. Boring, D. A. Few, J. L. Marble, and M. C. Walton. "I call shotgun!": An evaluation of mixed initiative control for novice users of a search and rescue robot. *(2004) Proceedings of the 2004 IEEE conference on Systems, Man, and Cybernetics.*

[3] D. J. Bruemmer, D. A. Few, C. W. Nielsen. 2006. "Spatial Reasoning for Human-Robot Teams," in Emerging Spatial Information Systems and Applications, ed. Brian Hilton, Idea Group Inc. Hershey, PA, pp. 350 – 372

[4] K. Konolige, "Large-scale map-making," in *Proceedings of the American Association for Artificial Intelligence (AAAI)*, San Jose, CA, 2004, pp. 457–463.

[5] J. S. Gutman and K. Konolige, "Incremental mapping of large cyclic environments," in *ProceedingsConference for Intelligent Robots and Applications(CIRCA)*, Monterey, CA, 1999, pp. 318–325.

[6] D. J. Bruemmer, D.A. Few, S. Sirin, H. Hunting, M. Walton, and F. Carney. Autonomous robot for sensor characterization. *Robotics and Remote Systems for Hazardous Environments*, Gainesville, Florida, March 28-31, 2004

[7] D. J. Bruemmer, D.A. Few, R. L. Boring, J. L. Marble, M. Walton, and C. Nielsen. "Shared Understanding for Collaborative Control." *IEEE Transactions on Systems, Man, and C*, 2005*ybernetics , Part A.* Systems and Humans, vol.35, no.4, pp. 505-512, Jul. 2005. .

[8] D. J. Bruemmer, D. A. Few, R. Boring, M. Walton, J. L. Marble, C. Nielsen, and J. Garner, 'Turn off the television!:' Robotic exploration experiments with a 3-D abstracted map interface," in *Proc. 38th Hawaii International. Conference on System Sciences*, Waikoloa, HI

[9] D. J. Bruemmer, D.A. Few, R. L. Boring, J. L. Marble, M. Walton, and C. Nielsen. "Shared Understanding for Collaborative Control." *IEEE Transactions on Systems, Man, and Cybernetics , Part A.* Systems and Humans, vol.35, no.4, pp. 505-512, Jul. 2005.

[10] C. Nielsen, M. Goodrich, and J. Crandall, "Experiments in human–robot teams," in *Proc. Int. Workshop Multi-Robot Syst.*, 2003, pp. 241–252.

[11] D. J. Bruemmer, and M. O. Anderson. "Intelligent Autonomy for Remote Characterization of Hazardous Environments", *in Proceedings of the IEEE International Symposium on Intelligent Control*, Houston, TX. October 2003

[12] E. J. Pedhazur and L. P. Schmelkin Measurement, Design, and Analysis: An Integrated Approach, Lawrence Erlbaum & Associates, Hillside NJ. 822 pgs. 1991.

[13] J. L. Harbour, D. J., Bruemmer, and D. A. Few "Measuring Unmanned Vehicle System Performance: Challenges and opportunities." *AUVSI Unmanned Systems North America*, Orlando, FL. August 29-31, 2006

[14] D. A. Few, D. J. Bruemmer, M. C. Walton "Improved Human - Robot Teaming through Facilitated Initiative." In Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (ROMAN). Hatfield, United Kingdom, September 6-8 2006. — 193kB PDF

[15] S. Hart, "NASA Task Load Index (TLX)," Human Performance Research Group, NASA Ames Research Center, Moffett Field, CA. Version 1.0

APPENDIX A - COMPOSITE PERFORMANCE MEASURE

$$\frac{\left\langle \begin{array}{l} \left[(1-(\frac{TotalTimeTaken}{MaxTimeTaken}))+(1-(\frac{Time/SqFt}{MaxTime/SqFt}))+(1-(\frac{OperatorConfusion}{MaxOpCon}))+(1-(\frac{OpCon/SqFt}{MaxOpCon/SqFt}))+(1-(\frac{RI}{MaxRI}))\right] \\ +\left[(1-(\frac{RI/Time}{MasRI/Time}))+(1-(\frac{EndDistanceFromStartLocation}{MaxDistance}))+\frac{StartSA}{10}+\frac{Source1SA}{10}+\frac{Source2SA}{10}\right] \\ +\left[(1-(\frac{AverageWorkload}{20}))+AverageSA\right] \end{array} \right\rangle}{12}$$

DISCLAIMER

This paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product, or process disclosed in this paper, or represents that its use by such third party would not infringe privately owned rights.

Standardizing Measurements of Autonomy in the Artificially Intelligent

Amy R. Hudson
National Institute of Standards and Technology
University of Maryland College Park
Gaithersburg, MD
ahudson5@umd.edu

Larry H. Reeker
National Institute of Standards and Technology
Gaithersburg, MD
larry.reeker@nist.gov

***Abstract*—** The amount of control that an intelligent system has over their actions, whether they are able to act independently from their creator, plays a major factor in describing systems and in distinguishing them from each other. Different levels of autonomy reflect the different abilities of the machines as well as where and how they can play a part in our daily lives. We may begin to comprehend these abilities and possible applications into human society once we can classify the levels of autonomy. The goal of this project is to set a framework for establishing a standard of autonomy in the scientific community by examining past and current methods of measurement as well as exploring different levels of autonomy's ethics and implications. Once this framework is made available to the scientific community, more tests and experiments will be conducted to refine and further ingrain the framework so that classifications of artificial intelligent agents are available universally. If we are to continue improving upon our machines, developing them to be more adept at communicating and accomplishing tasks, then a set of standards must be established for the safety and convenience of mankind.

Keywords: *autonomy, artificial intelligence, standards*

Artificial Intelligence (AI) is a relatively new area of research and study, having been around for only fifty years. Its waters are uncharted, and as time progresses, there are more and more possibilities of research brought to the foreground, a multitude of things we may accomplish in collaboration with humans and the artificially intelligent. DARPA (Defense Advanced Research Projects Agency) is extremely interested in employing robots to take over the more dangerous and dirty parts of being a soldier, such as searching for enemy combatants in a crowd or unknown territory, to fetching injured soldiers from the battlefield. The threat of quivering in surgery has become obsolete with the use of robots to help out in making precise incisions into patients, just as countries with aging populations are considering the benefit of having robots available to care for the elderly. Through communicating when pills should be taken, providing or aiding in transport, the burgeoning AI generation will care for our elderly, taking our dispositions on retirement and our progressing age and making them more pleasant and convenient to both the aging and the people who have to care for them. This lightening the load, so to speak, captures the drive behind further investigating AI and using it as a source of helping society to cope with unpleasant tasks, as well as providing a new curiosity with which its citizens are free to explore and philosophize about. Now that the importance of AI research and testing has been (at least tentatively) established, we can proceed to emphasize the importance of having standards of measurement for artificially intelligent machines, both expediting and simplifying the research process, going about in an organized way so that the AI world will be on the same page and better prepared to provide insights that can easily be jumped off from to reach a new idea and new way in which our community may benefit.

The authors feel that the most pressing part of a machines makeup to standardize is their autonomy level. Autonomy (or the closeness of it) is THE major factor in determining what task a robot is able to complete; knowing how much human supervision and maintenance provisions a company will have to provide for a machine can help the company in either budgeting for these provisions, or perhaps investing in developing a program for the machine that allows it to conduct its task in its environment with much less human intervention, and therefore less costs. If autonomy is standardized then companies across the world would be able to accurately gauge expenses, and computer scientists interested in AI and robotics could test their latest exploits in making a machine more independent by using the standards both their competition is using, and the same standards they themselves have used in the past. There are already a wide variety of machines relying different amounts on their human creators, from vacuums that sense surfaces and clean your house without you pushing them or directing them, to cars that are able to navigate across deserts. With standard autonomy levels we will have a convenient way of measuring improvement in the field of AI.

There are five areas in particular where a human may intervene on the behalf of a machine. If these five areas summarize autonomy levels, then measurements of these areas can then be translated into a general measurement of autonomy, even as standardizing these measurements would standardize measurements of autonomy. To accurately standardize these measurements, research must be done on the spectrum of abilities our machines have now, matching today's autonomy levels with levels we can deduce after learning more about AI's possibilities in future research. This paper provides a framework which may prove useful to the scientific community when they are compiling these autonomy levels [see Epilogue below].

The first three areas where a machine may require aid from a human will be relatively simple to measure, especially if an ontology of autonomy is created amongst robots, so that the same scale and checking box can be used for every robot. Whether a robot can replicate, or can change programming or physical aspects of itself to better suit what it interprets its task to be, reflects its creating abilities. A one would be given to the system that can only be created with human help, a two to the system that can only create software and program or the system can only recreate physically, and a three to the system that is able to replicate itself completely; these numbers are just suggestions. A robot could be dependent on humans or other robots for movement to complete its task, being worn or carried perhaps. Is the system able to move without human interaction? The less independence it has, the lower score. Same thing for maintenance; is the system able to fix any malfunctions, physical and/or technical? Robots can get damaged in the course of working and a company that invests in a robot that is able to fix glitches in its system will save money. Also, if new programming updates are constantly being added by the robot automatically then the technology will always be up-to-date, and the robot will be able to accomplish more, faster! The machine must continue to adapt to its environment in order to be truly autonomous.

Communication is often an area where a human could intervene, maybe always present to translate what the robot senses into terms that they are able to, if not understand, then at least use to complete their tasks. Or the human (or another robot) would have to explain to other humans or robots what this machine is trying to show. Is the system able to effectively communicate with other machines and other humans? For measuring communication autonomy, the robots and people that the machine interacts with may be polled, with 50% of robots understanding the machine all the time, and only 10% of humans understanding it (without a "translator"). If the majority of the interactants do not understand the machine at anytime, the machine has the lowest communication autonomy, whereas if the majority of interactants understand all the time, then the machine gets the highest score. Surveys and questionnaires may aid in providing the researchers with the opinions of the surrounding public that has interacted with the machine, and would need to be modified as our communication trends are constantly being altered. Similar to communication, society's views on learning and testing for understanding vary significantly across both time frames as well as different cultures. Therefore, standardizing these components of autonomy should prove challenging.

The learning component of autonomy is one of the harder facets to measure quantitatively. We have created a general scale to organize the learning levels on, shown below:

1. Human solves problem with computational and data management

2. Computer interface adapts to human preference interactivity

3. Human specifies list of items and attributes and system clusters them

4. Supervised Learning: Human gives exemplars and computer learns

5. Reinforcement Learning: human interacts in learning by providing evaluation functions for system outcomes at various places

6. Means-ends analysis

7. Computer learns and transfers knowledge

8. Computer learns and can deal with new environment

This scale still leaves learning with an open-ended definition, one which different readers will interpret to mean various things. In the study of artificial intelligence, we are inflicting machines with our intelligence. It follows that these machines should be able to be tested on their learning skills with the same methods we employ on ourselves to reflect exactly what the person taking the test is capable of learning. Humans often do not completely understand each other (syntax and diction getting in the way), making analysis of tests shaky as answers need to be interpreted accurately. By participating in human discourse and grasping a greater understanding of it, we may develop AI systems that can communicate with humans and learn from them even better than humans can, as a human could adapt to the machine and then the machine might respond by adapting to the human.

Many studies have been conducted using the Structure of Observed Learning Outcomes (SOLO) taxonomy to determine how advanced a person's ability of learning and analysis is. The essay structure section of the analysis in SOLO is not applicable to our machine situation, but just as essays need to be well-ordered and structured, with lots of examples and specifics, machines the most independent from their creators will have an organized programming system and be able to create and interpret programs in that same organized fashion. English teachers are trained in reading a paper and knowing exactly which grade it deserves, due to how they have to sit and grade a multitude of papers at once. If English has found a way to change a qualitative assessment into a quantitative one, programmers should have training of assessing the structure and content of programs included in their education repertoire to accomplish a similar feat.

1. Misses the point
2. Single point
3. Multiple unrelated points
4. Intermediate
5. Logically related answer
6. Unanticipated extension

There are also many different types of learning, so that reinforcement learning, case-based learning, and others, could be assessed and compiled to all aid in reflecting the general learning capabilities of the machine. The scientific community will have to determine through experimentation how many of the different learning techniques should factor into the general learning assessment, and which of the many techniques even apply to learning in machines. Using only one or two of the known methods of testing learning may still be enough to obtain the score and level that is needed to interpret autonomy. If the accuracy of these tests are also recorded, then using just these tests are good enough for our purposes…is there really going to be a major difference between having an autonomy score of 3 and 3.5? and what will it be? Perhaps the standards of autonomy should become satisficed, since a more general term of autonomy will be more understandable to the public and maybe mean more for future calculations.

Once all of these five aspects of autonomy have been measured, they then can be added together to obtain another number. Sheridan's Model can then be broken up into appropriate number ranges for a 1 level of autonomy etc., so that the number you obtain from the five aspects of autonomy falls in a range for a specific level of autonomy.

Sheridan's Model: Levels of Autonomy in Decision Making [5]

100 % Human Control

1- Human considers decision alternatives, makes and implements a decision.

2- Computer suggests set of decision alternatives; human may ignore them in making and implementing decision.

3- Computer offers restricted set of decision alternatives; human decides on one of these and implements it.

4- Computer offers restricted set of decision alternatives and suggests one; human may accept or reject, but decides on one and implements it.

5- Computer offers restricted set of decision alternatives and suggests which one it, the computer, will implement if human approves.

6- Computer makes decision; necessarily informs human in time to stop its implementation.

7- Computer makes (implements) decision; necessarily tells human after the fact what it did.

8- Computer makes and implements decision; tells human after the fact what it did and only if human asks.

9- Computer makes and implements decision; tells human after the fact what it did only if it, the computer, thinks human should be told.

10- Computer makes and implements decision if it thinks it should; tells human after the fact if it thinks it should.

100 % Computer Control

The ranges of numbers to assign for the levels of autonomy also fall into the category of research and testing that needs to be conducted to obtain a solid ground for standardization, which may be updated as the years progress and more machines are created and not able to be labeled as a specific level. Updates to these standardizations will make for a much more robust way of measuring autonomy, being able to fix qualitative data to quantitative data so that much less bias is involved in the measurements.

One way in which we propose to instill these standards of autonomy in the future is to modify intelligent systems' ontologies to include an autonomy scale, created based off of the aforementioned qualities of learning, communication, movement, maintenance and creation. If the machine is able to rate itself on the scale, it will be able to communicate to people how much of a task it can be entrusted to perform without human input; how reliable it is. With this knowledge, people will be able to specify the best purpose the machine would serve, enabling the progression of forays into the AI realm of thought to be both organized and efficient.

EPILOGUE: SIX YEARS LATER: AUTONOMY & DEEP MEASUREMENTS FOR SCIENTIFIC CONSTRUCTS

We have made some suggestions on measuring autonomy in this paper, since it was a theme in this year's PerMIS, as well as being widely used today. "Theoretical Constructs and Measurement of Performance and Intelligence in Intelligent Systems" [Reeker, 2000] had a different role of trying to use measures and discover how they might help in developing scientific theories (e.g. Archimedes finding a way to measure amounts of gold and silver in a crown resulting in the study of hydraulics). In the 2000 paper, we also briefly talked about robustness and autonomy in extraction of information from text.

Our discussions to this point in this paper are more about measurements of autonomy in the engineering fields rather than a precise measurement leading to a part of a scientific theory. In the 2000 paper, engineering (by itself) is classified as a science of the artificial (certainly true with robotics). But engineering and pure science are linked in many areas, as one can see computational and scientific theories in the work of Herbert Simon and Allen Newell[*]. Testing for Autonomy by using levels of Autonomy or readiness levels and finding out how the system performs for each level, in what can be called a suite of performance metrics, specifies the machines abilities. Of course it doesn't tell whether the system is truly intelligent, and it probably would not satisfy Lord Kelvin, for instance. One can have multiple suites and multiple requirements for readiness, the different uses given in a suite on particular tasks.

The idea of the 2000 paper was to show that a young science like artificial intelligence should be looking for measures that are not only useful in developing technologies but can help put together a new science. This new science will have theories that then can be predictive through a calculus - or, as Simon and Newell had advocated - a computing program of the sort that is called today "computational science". Such theories tend to be able to predict what is going to happen in various situations that stand up within empirical experiments, and it was called in that paper "deep measurements" (instead of "surface measurements"). In fact, the importance of being able to develop computational systems as part of a theory is of interest to many of us, and computational science continues its ascent.

We most often see computational science used in [Bekey 2005] and [Mataric, 2007] and not only in robotics, but also in The University of Massachusetts' Autonomous, Learning Laboratory's all types of learning. It is often used in discussing systems of agents, which can encapsulate information -- for Simon, "the allegory of the watchmakers" in [Simon, 1969].

So what is in this concept of autonomy? In the first part of this paper, we talked about human interaction with autonomous systems, for robots and other systems. In the spirit of the 2000 paper, we would like to look at aspects of autonomy that might be deeper: Robustness (already mentioned in the 2000 paper), stability, adaptability, capability, and scalability. Stability is associated with control theory. Its usefulness was around in physical systems long before AI, but some new ideas for the information age can be found at e.g. [Reeker, L.H. and Jones, A.T., 2001].

Adaptability is one of the most interesting of the "ility"s and is changing the world because information can be sent so quickly, whether it is in fly-by-wire systems in airplanes to the world-wide web and cell phones. In AI systems, we see lots of these adaptability possibilities. In the 2000 paper, these ideas have been discussed through learning and transfer of learning. The ideas of machine learning are getting better, and testing is important to them. For example, the ensemble methods have shown how to take learning systems and make interchanges between variance, noise (e.g. outliers), and bias. They can be tested using ROC charts (which are hardly different) that tell likely false positives and false negatives. New ideas have emerged for finding information in great amounts of text, and it has been shown that both ROCs and the standard precision and recall measurements used together are better than only one of either.

Ontologies, which were only mentioned in the 2000 paper, are clearly needed for communication between intelligent systems, whether human intelligence is studied by cognitive systems or artificial intelligence systems. Ontologies need more study and they are getting it through different efforts, such as the Semantic Web. They need tests and metrics, too; PerMIS is doing an important job in developing computer and information science and changing the world.

[*] Many people do not realize the amount of philosophy that both Allan Newell and Herbert Simon used in their work. They are thought of as renaissance men, mostly as in computer science, mathematics and statistics, cognitive science, and more, but they were also both philosophers, important in helping to develop the new science of artificial intelligence.

To conclude, there is still a lot of need for metrics of all types, both at the surface and in depth.

References

[1] Bekey, G.A., [2005], Autonomous Robotics, MIT Press

[2] Mataric, M. J. [2007], The Robotics Primer, MIT Press

[3] Reeker, L.H. [2000], Theoretic Constructs and Measurement of Performance and Intelligence in Intelligent Systems, Proceedings of the 2000 PerMIS Workshop. Available from NIST/MEL

[4] Reeker, L.H. and A. Jones, [2001], Measuring the Impact of Information on Complex Systems, Measuring the Performance and Intelligence of Systems: Proceedings of the 2001 PerMIS Workshop. <http://www.isd.mel.nist.gov/research_areas/research_engineering/Performance_Metrics/past_wkshp.html>

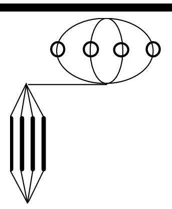
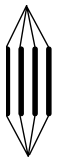
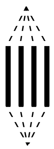
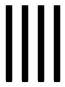
[5] R. Parasuraman, T.B. Sheridan, and C.D. Wickens, “A Model for Types and Levels of Human Interaction with Automation Transactions on Systems, Man, and Cybernetics- Part A, vol.30, pp. 286-297, 2000

[6] Simon, H.A., [1969], The Sciences of the Artificial, MIT Press

[7] The Chinese University of Hong Kong, <www.cuhk.edu.hk/clear/download/PDC/n23_SOLO_assessmt_grid.doc>

Appendix A:

The SOLO taxonomy as a guide to setting and marking assessment

SOLO category	Representation	Type of outcome	Solution to problem	Structure of essay
Unanticipated extension		Create Synthesise Hypothesise Validate Predict Debate Theorise	Solution to problem which goes beyond anticipated answer. Project or practical report dealing with real world ill-defined topic.	Well structured essay with clear introduction and conclusion. Issues clearly identified; clear framework for organizing discussion; appropriate material selected. Evidence of wide reading from many sources. Clear evidence of sophisticated analysis or innovative thinking.
Logically related answer		Apply Outline Distinguish Analyse Classify Contrast Summarise Categorise	Elegant solution to complex problem requiring identification of variables to be evaluated or hypotheses to be tested. Well structured project or practical report on open task.	Essay well structured with a clear introduction and conclusion. Framework exists which is well developed. Appropriate material. Content has logical flow, with ideas clearly expressed. Clearly identifiable structure to the argument with discussion of differing views.
Intermediate			Solution to multiple part problem with most parts correctly solved but some errors. Reasonably well structured project or practical report on open task.	Essay fairly well structured. Some issues identified. Attempt at a limited framework. Most of the material selected is appropriate. Introduction and conclusion exists. Logical presentation attempted and successful in a limited way. Some structure to the argument but only limited number of differing views and no new ideas.
Multiple unrelated points		Explain Define List Solve Describe Interpret	Correct solution to multiple part problem requiring substitution of data from one part to the next. Poorly structured project report or practical report on open task.	Essay poorly structured. A range of material has been selected and most of the material selected is appropriate. Weak introduction and conclusion. Little attempt to provide a clear logical structure. Focus on a large number of facts with little attempt at conceptual explanations. Very little linking of material between sections in the essay or report.
Single point		State Recognise Recall Quote Note Name	Correct answer to simple algorithmic problem requiring substitution of data into formula. Correct solution of one part of more complex problem.	Poor essay structure. One issue identified and this becomes the sole focus; no framework for organizing discussion. Dogmatic presentation of a single solution to the set task. This idea may be restated in different ways. Little support from the literature.
Misses the point			Completely incorrect solution.	Inappropriate or few issues identified. No framework for discussion and little relevant material selected. Poor structure to the essay. Irrelevant detail and some misinterpretation of the question. Little logical relationship to the topic and poor use of examples.

Assessment of Man-portable Robots for Law Enforcement Agencies

Carl Lundberg
National Defence College
115 93 Stockholm, Sweden
carl.lundberg@fhs.se

Henrik I. Christensen
Georgia Institute of Technology
Atlanta, GA 30332, USA
hic@cc.gatech.edu

***Abstract* -** This project has involved testing a Packbot Scout within a SWAT-unit[1] for five months. This was done to explore the tactical benefits of the system and to test the robot's technical performance with end users. Another objective was to compare earlier results – obtained by investigating military during training – with results from deployment during true risk. The SWAT-team, equipped with and trained to use the robot, set a standard to bring it with them on regular missions. Using the robot during negotiation proved to be the most beneficial application. Other uses would be for long-term surveillances and deploying non-lethal weapons. Early results indicate that the Stockholm SWAT-unit, consisting of 80 active officers, could deploy the robot at least 20 times a year.

Keywords: *SWAT police, user study, man-portable robot, Packbot*

I. Introduction

Robots are already an established tool for high-risk applications such as EOD[2]. Other applications could benefit from the use of robots, although a number of issues must be considered to enable deployment on a regular basis. The technical design must be adjusted to meet special requirements for other applications, requiring detailed knowledge about the end users and the tasks they face. Relevant niches in which robots can perform successfully need to be identified, and methods for deployment have to be developed. Robot systems need to be versatile, not only serve multiple purposes for one particular user, but also adapt to several different professions. Keeping the assorted end users in the loop during product development, while simultaneously exploring methods for deployment is crucial to achieve successful and rapid implementation.

In previous studies we have investigated man-portable robots for Military Operation in Urban Terrain[3] [1]. These studies were performed during military training maneuvers which in general provided a realistic setting. One aspect, however, was not accurately represented during training – the relation to mortal danger. As a consequence we decided to perform a parallel study involving a user group in actual risk, namely the Stockholm SWAT-unit. SWAT-units do, just as MOUT-soldiers, target people rather than artifacts or substances such as in EOD, CBRN[4], and USAR[5].

The objectives of the project were to:

- Investigate if users at real risk render results that significantly differ from results obtained during training maneuvers.
- Broaden the scope of knowledge regarding the feasibility of robots within another high-risk work group.
- Perform continued user-governed assessment of the Packbot Scout[6] in realistic settings.
- Survey a user group to identify opportunities for continued research.

This paper presents initial findings gained through two sets of interviews and one written mission report[7]. The first set of interviews was performed with the SWAT-unit's chief and a member of their Training and Development team, at the time the robot was handed over for test[8]. The second set of interviews was performed with the two officers selected to operate the robot after having had the opportunity to deploy the robot for five months[9]. The results were verified with the respondents.

This article is organized with related work in section 2, a description of the users in section 3, and a description of the robot in section 4. Section 5 describes how the robot was dealt with during the trial and how it could be deployed in the future. Section 6 discusses the results and suggests future work.

[1] Special Weapons And Tactics.

[2] Explosive Ordnance Disposal, i.e., removal, disarmament, and destruction of explosives.

[3] MOUT

[4] Chemical, Biological, Radiological, and Nuclear detection and decontamination.

[5] Urban Search And Rescue. The goal in USAR is to localize humans confined in destructed buildings. The victims are considered to be static unlike the targets of MOUT or SWAT-missions.

[6] The robot system is described in Section 4.

[7] A one page police report describing a live mission performed with the robot.

[8] This interview was performed with both respondents at the same time and lasted for 1 hour and 40 minutes.

[9] These interviews were performed with one respondent at the time and lasted 45 min each.

II. Related Work

Various studies have previously investigated high-risk workers deploying field robots. The most common application, bomb removal or destruction, has been successively refined since the first attempts in Northern Ireland in the beginning of the 1970's [2]. Today this is a well established robot niche with several mature systems available as demonstrated at the European Land-Robot Trial 2006 [3]. Other areas of robot deployment shared by the police and military are security, surveillance, reconnaissance, and tactical support [4, 5, 6; 7, 8, 9]; these are areas that have received substantial investments, although much of the research is not published in detail [10]. The task of CBRN contamination control seems to be a prominent next step as sensor payloads are maturing for deployment on robots that are already in daily use [7, 11, 12, 13; 14]. Rescue robotics, and especially Urban Search and Rescue, is one of the areas of field robotics currently receiving the most attention in academic research. Countermeasures against, and preparedness for terrorist attacks and earthquakes have invigorated efforts to push robot technology into use [15, 16, 17, 18, 19].

Human-robot interaction outside the scope of high-risk field workers has been targeted for research as well. An early example is the integration of the SURBOT [20] for mobile surveillance in a nuclear power plant. More recent examples consist of testing of the robot seal Paro amongst elderly [21], the fetch-and-carry robot CERO by a partially impaired person [22], and a number of long-term tests of tour guide robots such as the RoboX9 at Expo02 [23]. By now space applications have been tested substantially through NASA's deployment of rovers on Mars [24].

Robot deployment within SWAT-missions specifically[10] is performed and occasionally reported in news-media [25, 26, 27]. Most of these cases seem to be ad hoc solutions in which EOD-robots are used for other applications. Although the academic community has published little on robotics for SWAT-tasks [28, 29, 30], there are commercial products aimed at the application [12, 14, 31, 32].

III. User Description

A. Organization, demography, and training

Sweden has three main SWAT-units: Malmö, Göteborg, and Stockholm, who attempt to keep methodology and gear aligned since they occasionally perform joint missions. The Stockholm unit, 85 members strong and the largest of the three, is organized into eight SWAT-teams, each consisting of 8-9 officers. Each team works four shifts per week. The number of teams on service varies with the expected amount of crime, with at least one team on duty at any given time[11]. During daytime it is common to have one team on alert, and another scheduled for training acting as backup. Although the teams have an appointed leader, most decisions are made jointly; only under time-pressure is hierarchal leadership enforced. The Stockholm SWAT-unit has four mission commanders who handle crime-site command and communication with the police chief. There are 22 negotiators associated with the SWAT-unit. Most of them are stationed elsewhere but are on call. Due to physical demands, the members of the SWAT-teams are currently all male[12]. The negotiators on the other hand, always work in a pair of one male and one female, for tactical advantage purposes. It is moreover attempted to have a diverse ethnical background amongst the negotiators.

The average age within the SWAT-team is 36 years. Average time spent with the unit is 8-9 years. A minimum of five years of police service is required before being considered for the 3-month special SWAT-training[13]. 20% of the working hours are spent on training, which to a large extent is handled within the teams. To be able to act swiftly and in a synchronized manner, the SWAT-teams use predefined and well practiced concepts based on reference scenarios[14]. Despite all teams receiving the same basic training and having the same gear, they occasionally develop their own behavior depending on experiences encountered; individualization is discouraged by management in the interest of interoperability. In the past all SWAT-team members were encouraged to be able to handle all techniques and equipment. Recent increases in technical complexity have required the team members to assume specialized roles. Keeping the competence for different technical aids high is considered a problem; new gear is not always properly evaluated.

B. Tasks

In contrast to many other police units, whose objective is to prevent crime, the SWAT-teams are mainly reactive; although they are occasionally deployed proactively to demonstrate suspicion and readiness to strike. Their main objective is to target dangerous situations. Common tasks include resolving hostage situations, arresting potentially aggressive suspects, and taking suicidal or violent mentally deranged persons into custody. In other cases they are called upon to perform rapid arrests or searches to prevent suspects from disposing of evidence. The SWAT-teams may also be used for riot control or routine missions such as high-risk escorts or searching for missing persons.

Missions are initiated either by alarm of an ongoing crime, or by the request of assistance by another unit (response respectively planned missions). Responding to an ongoing crime is more frequent. Apartments or homes are the most

[10] EOD-robots excluded.

[11] SWAT-units are organised in shifts to provide permanent service over time. Military units are to more extent deployed the entire unit at once with periods of recuperation in-between.

[12] A program to equalize the gender distribution is ongoing.

[13] The police officers are older, have more experience, and are allowed to have an opinion in larger extent than the soldiers [1].

[14] This, although, the SWAT-police considers them self to be less oriented towards training and relying fixed behaviours than the military. Larger space is left to individual solution from one case to another.

frequently targeted environments, but *open-air* missions occur as well. The SWAT-units are equipped and trained to perform their duties wearing gas masks. Targeting suspects in possibly toxic environments occurs 2-4 times per year[15]. The Stockholm SWAT-unit on average performs close to one high-risk mission per day. 600 missions were performed during 2006. Of these, half were classified as high-risk missions. The most common tasks include dealing with severe criminals or organized crime.

C. *Typical scenario*

In advance of planned missions, the requested units usually survey the strike scene in detail[16]. This includes gathering evidence, getting to know the suspects, their armament, their vehicles, and the layout of the strike area. If the suspects reside at different addresses, the arrests are often synchronized. Planned missions usually occur before or after the crimes are committed, in order to minimize risks to third parties.

During crime response missions, the first objective is to locate and confine the suspects to prevent escape or hostage taking. Subsequently, the mission commander, the SWAT-team commander, and negotiators decide how to address the situation. A defensive approach, which entails that the suspect surrenders according to conditions stated by the police, is preferred. Negotiation makes up a large portion of this situation and can be a tedious process[17]. Long negotiations challenge the SWAT-teams' ability to maintain a high level of readiness. Missions lasting longer than 6-9 hours require a relief unit.

Offensive actions are based on forceful confrontation with the purpose to shock and overwhelm the suspects. Distractions such as teargas, pepper spray, or shock grenades might be used. The use of distractions or deliberate weapons fire (for other than self defense purposes) has to be sanctioned by the police chief.

The Swedish police are increasing efforts towards non-violent solutions through negotiation[18]. Decreasing human violence is regarded far more important than avoiding material damage. Breaching doors is the most common destruction during SWAT-missions.

D. *Limitations*

When asked about the main limiting factor, the robot operators responded that the restrictions imposed by the commanders[19] were the most constraining to their performance. Despite proper competence, knowledge, and tools to act, the SWAT-teams feel they are held back from solving cases.

Personal risk was not reported to be a very limiting factor; mission commanders usually take preventive measures to avoid risks to third parties or the suspects, long before the SWAT-officers regard themselves endangered[20]. The most life threatening moments were considered to occur during emergency vehicle transports or vehicular pursuits. The SWAT officers argued that their being aware prepares them for dangers, whereas the police in general to greater extent encounter high risks by surprise. They also reported that they are often able to demonstrate enough superiority to cause the suspects to surrender without resistance.

IV. The Robot System

A. *The Robot*

The iRobot PackBot Scout is a man-portable robot tele-operated using a video link (Fig. 1). The track propulsion system includes articulated tracked arms (flippers) which can be rotated 360 degrees. The flippers enable significant off-road abilities considering the small dimensions of the robot; in addition they enable recovery from roll-over. The top speed of the robot is 3.7 m/s and the Ni-Cd batteries enable an operating time of about three hours. The PackBot is equipped with fish-eye daylight video camera, IR-camera[21], IR-illuminator, GPS receiver, electronic compass, and absolute orientation sensors (measuring roll and pitch).

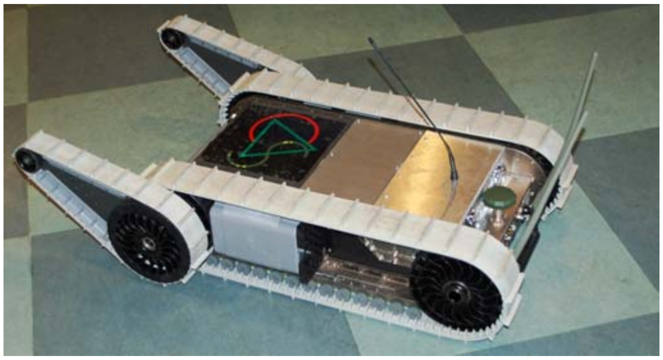


Figure 1. The Packbot Scout with the distraction siren (centered on top of the robot).

[15] The Swedish Emergency Management Agency is funding acquisition of sealed CBRN-vehicles to provide the police with the capability to operate in hazardous environments; robots could play a role in within this.

[16] This was also reported by Jones et al. [28]. The military will, in comparison, most likely be less informed [1].

[17] On one occasion a negotiation lasted for 44 hours.

[18] The ambition to achieve non-violent solutions was pointed out to vary greatly between countries. In particular, Australia and United Kingdom were mentioned to favour negotiation before violence.

[19] Police chief as well as the mission commander.

[20] This on the contrary to EOD-technicians or MOUT-soldiers who report risks to be a crucial limitation [34].

[21] Infra Red, in the close to visible spectrum.

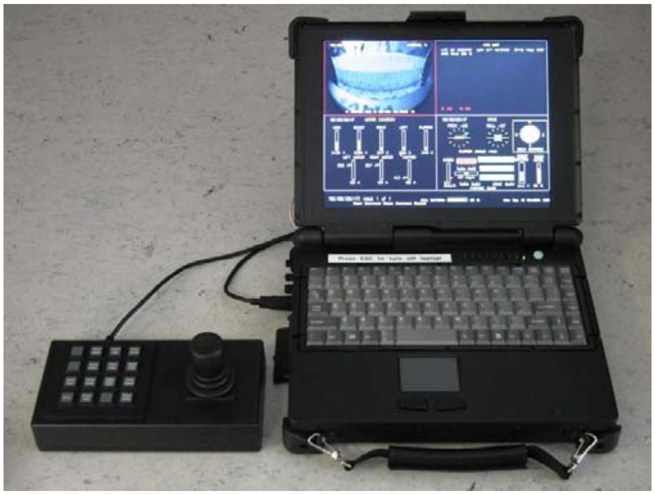


Figure 2. The operator control unit.

The operator control unit consists of an Amrel Rocky Patriot rugged laptop fitted with a joystick[22] allowing for three degrees of freedom, and a keypad for toggling functions on/off (Fig. 2). Communication between the robot and the user interface is achieved using double IEEE 802.11b radio links.

A carrying system was added to both the robot and the operator control unit to enable hands-free portability. Other field adaptations included fitting the joystick, keypad, and cable connectors with protective covers. A small whiteboard was attached to the laptop with Velcro so that it could be easily removed and used by the operator to sketch the explored region. Extra batteries and chargers, both for wall-socket and vehicle charging were provided, as well as protective cases for transport and storage.

B. *The Payload*

During the project the robot was equipped with a distraction siren (Fig. 1). The siren is originally an alarm siren for intruder deterrence, developed and manufactured in Stockholm by Inferno[23]. The patented siren generates a high-pitch noise which is intolerable to the naked ear. Four different frequencies are modulated to cognitively overload the auditory organ while not causing hearing impairment (123-127 db(A)). Wearing hearing protection or plugging ones ears blocks the effect.

V. Robot Deployment

A. *Deployment during trials*

The joint study was initiated in mid-December 2006[24] when researchers met with representatives from the Development and Training group of the unit. The meeting addressed working out guidelines and legislation issues for the trials. The police also gave a general overview about their work. It was decided to perform the testing with one of the eight SWAT-teams until May 2007. The appointed team was trained in the basics of robot operation a few days later.[25] It was left up to them to use the robot as they considered appropriate, during training and real missions. The one-day training session included a brief description of how the military had been using the system in urban intervention [1]. Two team members were appointed robot operators for the duration of the trials. It was declared that real mission deployments were of interest to the study, while it was not of great concern whether the robot was damaged. The distraction siren was added to the robot system by March 2007[26]. The interviews with the operators were performed at the beginning of May 2007[27].

After handover, the two operators continued to train with the robot about once per week. In addition, they gave the other team members the opportunity to familiarize with the robot's performance and try operating it. Training – performed both outdoors and indoors – included passing obstacles and operating under different lighting conditions. The most frequently trained task was mapping of previously unknown premises and locating suspects. During three training sessions, the operators first explored a premise before executing a strike mission into the investigated area and finally evaluating the benefit of previous knowledge.

The distraction-siren payload was evaluated in a mock hostage situation during which one officer acted hostage taker and one officer acted hostage; both were previously unacquainted with the distraction-siren. The test showed that the noise, although extremely annoying, does not completely disrupt willpower (Figure 3).

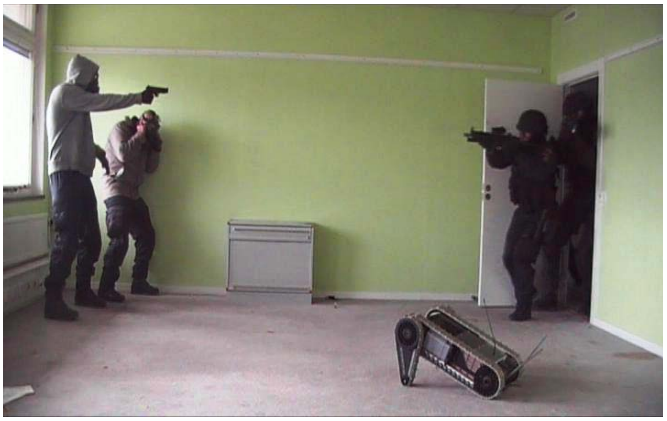


Figure 3. Tactical test of the distraction-siren. From left to right: the officer acting as criminal; the officer acting as hostage; the two SWAT-officers attacking. The hostage taker was instructed to shoot at the police, which he succeeded in despite the siren. The hostage immediately plugged his ears with his fingers. The electronically filtered hearing protection used by the police protected them from the noise.

[22] Sideways, forward/backward, and twisting the knob (to control the flippers).

[23] www.inferno.se

[24] 14 December 2006

[25] 19 December 2006

[26] 18 March 2007

[27] 8 May 2007. The trials are continuing.

Once the team had familiarized themselves with the robot, they decided to include it on missions involving five or more police officers. This was the case for about half of all missions performed. On missions with fewer than five participants, the team in general considered that no one could be spared to operate the robot. In addition, the jeep used for transport of a small number of people did not have much extra space; accommodating the robot was not a problem for large teams since they had access to a van. Since only one SWAT-team was trained to bring the robot, and did so on half of their missions, the robot was available approximately 10% of the total time.

The robot was deployed in one real mission during the five-month trial[28]; it was used to investigate a suspect bomb in a staircase outside an apartment[29]. The robot enabled the police to keep the suspicious object, as well as the surroundings, under surveillance with standoff. Once the bomb squad arrived, the robot was used to gain initial information about the object and the surroundings. While the object was targeted by a bomb-technician wearing a bomb suit, the robot was used by the others to monitor progress.

The robot was also considered for exploration of a smoke-filled shop which was not on fire. After the team broke the door of the shop, they intended to use the robot to search for victims, but the fire brigade arrived and took over before the mission was initiated.

The operators reported that it is usually possible to find a safe spot for the operator[30]. Handling the robot was not found too challenging for field operation, though the control unit lacks key-backlight which is required in darkness. The operators considered the video feedback to be fairly adequate. However, they thought an improvement in resolution would be beneficial, as well as the ability to pan/tilt the camera, since having to elevate the front of the robot with the flippers to view upwards (Fig. 4) proved time consuming. A backwards facing camera was suggested to make backing out of narrow spaces more convenient. A zoom function was further suggested to enable closer inspection[31].

The range of the radio link was considered sufficient to cover apartments, which is the type of premise targeted the most. Operations were usually performed from a staircase or neighboring apartment. Ruggedness and reliability were satisfying as well, although the users claimed the operator control unit and the robot sometimes failed to synchronize[32].

Spiral staircases were the only obstacles said to pose a problem. This problem became evident during the live mission targeting the suspected bomb. The police vehicles can generally approach the mission area fairly close making the distance the robot has to be carried not being very far. The robot was considered heavy though not a major obstacle[33]. The size became a problem only during vehicle transportation.

The users immediately noticed the absence of two-way audio, which would make voice communication possible with suspects and victims. Missions including negotiations might, as mentioned, span for an extended period of time. Battery replacement and the possibility to charge batteries, both from wall sockets and vehicles, are needed. The operators additionally suggested the ability to charge the batteries while mounted in the robot, instead of first having to remove them.

The distraction siren was considered to be of significant interests as it is less violent compared to shock grenades or chemical agents, and therefore might be less restricted for use. Suspects' and victims' reaction to the robot is an open issue; the robot might appear frightening, increase aggressiveness, or be ignored. The trials did not give any opportunity to investigate this issue, which can hardly be examined with validity during training.

B. Considerations on future deployment

Apart from the mission actually performed (inspection), the respondents indicated a number of possible applications. The most prominent task suggested was to use the robot as a tool during negotiation[34]. In the first phase it could be used to establish communication with the suspect either by bringing in a cell phone/radio or establishing a two-way audio link on the robot[35]. During negotiation, the robot could be used to transport items to and from the suspect (the counter-parts often demand food, cigarettes etc.). The robot could furthermore be used for retrieving weapons in case of surrender.

Using the robot for the mentioned applications would provide the opportunity to observe the suspects' aggressiveness, rationality, armament, the premise, and possible hostages. If negotiating with suicidal individuals, the robot might be used to monitor their mental state. As demonstrated in the live mission, the robot can also be used for visual inspection of objects[36]. A robot equipped with non-lethal weapons could be used for distraction if negotiations fail. Adding non-lethal weapons such as tear gas to the robot, however, poses a risk, as the weapons could come into the offenders' possession. It was suggested that the robot should

[28] 18 February 2007

[29] The suspected bomb was located outside an apartment used for persons being under protection.

[30] The enemy's location will be less know during MOUT which requires the operator to be protected by other soldiers [1].

[31] Backwards facing camera and zoom are features available on the URBOT [7].

[32] This error might have been caused by the fact that the OCU does not work properly after having been put in, and taken out of, the laptop's standby-mode. The standby-mode is activated by hitting the on-button while the ESC-key is used to turn off the lap-top. Making the mistake to attempt a reboot using the on-button might have been the cause of the robot comms lost error.

[33] Military missions might, on the contrary, include covering significant distances on foot which makes weight more important. MOUT trails proved the weight of the Packbot to be right on the limit to what can be accepted for a man portable system [1]. Something that is verified by work with the URBOT that weights 30 kg [30].

[34] This was also an application pointed to be of interest in MOUT [1].

[35] Features that have been taken into consideration by Robotic FX [14].

[36] This has also been suggested by the military and would benefit of a snap-shot and zoom function in the user interface [1].

have a self-defense system, such as the ability to administer electrical shocks.

Another suggestion was to use the robot for long term surveillance of a door or a passage to relieve police officers[37]. The robot could also enable the police to manifest their presence without exposing personnel to risks. Additionally, the robot could be used for missions in hazardous environments if equipped with appropriate sensors. The operators stated that the robot mainly would be used for defensive purposes on missions, i.e., to locate suspects and initiate negotiations, rather than to target them. The robot was not considered to be suitable for offensive deployment as it does not have the ability to act against the counterparts and as it is too slow. To circulate and map an area holding the suspect did not seem to be a likely application[38]. It was pointed out that outdoor operations could come into question, although this was not tested to any large extent. Considering the restrictions for using violence, the operators did not regard equipping the robot with lethal capabilities to be of any interest[39].

The main benefits robots could bring to SWAT-deployment were as an enabler of a number of new features during negotiation, and also some new tactical advantages in case the mission had to be solved offensively. The users did not consider the system to have a major influence on their personal risk[40]. The police did not consider the robot to have imposed any major disadvantages. The only negative issue mentioned was that a robot system would entail yet another high-tech utility requiring maintenance, training, transport, etc. It was not believed that the option of a robot would make the police officers decline to perform risky duties themselves[41]. In addition, it was mentioned that the doer-mentality and high ambition to achieve immediate results might prevent the SWAT-police from deploying the robot[42].

C. Acquisition

The operators were asked to estimate how often the robot would be deployed if the suggested improvements were included. They felt that their team had encountered unusually few opportunities to deploy the robot during the evaluation period, but one of the operators estimated that the robot could be part of every fifth high risk mission of the Stockholm SWAT-unit (about once per week).

One of the two operators distinctively argued that the tested system should be acquired once two-way audio and key-backlight had been incorporated. The other operator was more ambiguous. Although he stated that the robot could be valuable, he argued that acquisition depends on cost and stated the price limit to be about 29,000 USD. The other operator projected the price limit to about 43,000-57,000 USD[43].

Neither of the operators could suggest any alternative equipment they currently lack, that would be preferred over the robot. On the other hand, they did indicate occasional shortage of personnel to be a limiting and risk-increasing factor. When asked to compare the benefits of the robot to night vision goggles, both operators argued night vision goggles to be more useful[44].

Both respondents agreed that one robot would fulfill the tactical needs of the entire unit. Having a second system for training and for backup would be convenient. It is currently being evaluated if the unit should be equipped with a designated vehicle for the new technical equipment; it was suggested that the robot should be stationed in the tech-vehicle. Estimating how many robots would be destroyed during a year proved difficult as the suspects' reactions to robot encounter had still not been experienced. One operator argued that it probably would not be very many while the other chose not to speculate.

VI. Discussion And Future Work

Performing tests in a real setting is of benefit to accuracy, but can also convey practical difficulties; especially when targeting high-risk applications. It has, in this study, not been possible to gain data from several parallel methods to verify validity through triangulation[45]. As indirect observations were the only source of information, it would have been particularly beneficial to have a large data set, i.e., many operators with extensive experience; unfortunately, this was not possible either. Only two respondents were available and their experience was, despite the rather long trial period, limited. In addition, there is an obvious risk of bias between the respondents since they work in the same team.

One of the reasons for selecting the SWAT-teams was to study a user under real risk. But, according to the two robot operators, they did not consider themselves to be highly endangered. From that aspect the setting might be considered inadequate to meet the objective, even though the risk-defying attitude might be the result of SWAT culture.

Despite limitations in data collection and misalignment with one of the objectives, we consider the results to provide a general overview and a starting point for continued studies. Apart from continuing and widening the ongoing trials, we believe that a theoretical analysis of the police-report records would provide statistical data useful for estimating the robot's value. Moreover, we consider the socio-technical and

[37] This would require a motion detection system as observing a video screen is a task that can not be performed with reliability over time [1].

[38] Contrary to MOUT where combat reconnaissance was pointed out to be one of the primary applications of the Packbot [1].

[39] Weaponization was considered highly interesting in MOUT [1].

[40] Reduced risks are the prime benefit for robots in EOD and MOUT. In MOUT are robots, in addition, believed to reduce weapons deployment [1].

[41] The military entertained apprehension that the robot would delay advance, revile presence, and might make the soldiers less willing to take risks [1].

[42] Behaviour commonly observed during the MOUT-trials [1].

[43] 200,000 SEK respectively 300,000-400,000 SEK. These amounts correspond fairly well with the tolerable price limit of 20,000-30,000 USD, reported by Ciccimaro et al. [30].

[44] Military considered the robot to be as valuable as night vision goggles during MOUT [1].

[45] For example through comparison of results from observations, interviews, and numerical data from experiments [33].

psychological aspects of robot-person interaction to be of particular interests.

Many of the presented findings align well with results from previous studies of both the police and military. For example, using the robot as a mean for communication is suggested by both groups. Considering the robot not to be suited for the most offensive and time-constrained tasks is another resemblance [1]. This and previous work on SWAT-teams result in similar estimations of tolerable price, and the anticipated mental, as well as physical, demands that can be placed on the robot operator [30]. There are striking differences as well[46]. While the MOUT-users demand longer radio range and improved visual feedback, the police officers are generally satisfied with the robot's performance. Military users show a significant interest in weaponization, while the SWAT-officers do not regard lethal abilities as a realistic application. Reduced risk and decreased weapon deployment are considered to be the primary benefits in MOUT. In SWAT, the system is seen as having the most potential as a tool for negotiation and surveillance over time.

VII. Conclusions

The question of whether robots should be acquired for SWAT-units calls for a comparison between frequency and importance of benefits, and the costs of implementation. Bringing the robot as an excuse to communicate or deliver items, and at the same time observe the surroundings, the suspects, and hostages was stated as a primary benefit. Once in place the robot could be used to deploy distractions during arrests. Long-time surveillance was considered as a suitable application as well. Unlike in MOUT and EOD, risk reduction was not considered as a main benefit of the robot. Nor was it of interest to give the robot lethal abilities such as suggested for MOUT. The investigated users were in general satisfied with the performance of the robot. Two-way audio, increased field of view, motion detection, and the possibility to store images for later viewing are desired improvements.

The interplay between the robot and those encountering it stands out as the most significant open issue. Limited experience of actual deployment and only two respondents with experience of the system are the primary limitations of the study. This prevented a reliable estimation of deployment frequency; however, if regarding the one mission performed during the five months test period as representative, the system would be deployed about 20 times per year. It was estimated that one robot would fulfill the tactical needs of the Stockholm unit. Acquisition is the primary cost connected to the introduction of systems like the Packbot. Costs for training, basic maintenance, and tactical development can be handled through available recourses with a slight expansion. The users estimated a tolerable price limit to be somewhere around 30,000-50,000 USD.

Acknowledgements

The participation of the Stockholm SWAT-unit as well as financial support from the Swedish Armed Forces is gratefully acknowledged.

References

[1] Lundberg, C., Forthcoming: *Assessment and evaluation of robots for dismounted high risk workers in urban settings,* PhD Dissertation Royal Institute of Technology, Stockholm, Sweden, October 2007.

[2] Birchall, P., *The Longest Walk: The World of Bomb Disposal.* London: Arms & Armour, 1997.

[3] Schneider, F., *European Land-Robot Trial* (ELROB), FGAN, Hammelburg, Germany. Retrived Febuary 1, 2007 from: http://www.m-elrob.eu/

[4] Carroll, D., Nguyen, C., Everett, H.R., and Frederick, B., Development and Testing for Physical Security Robots, *SPIE Proc. 5804: Unmanned Ground Vehicle Technology VII*, Orlando, FL, USA, March 2005.

[5] Barnes, M., Everett, H.R., and P. Rudakevych, ThrowBot: Design Considerations for a Man-Portable Throwable Robot, *SPIE Proc. 5804: Unmanned Ground Vehicle Technology VII*, Orlando, FL, USA, March 2005.

[6] Ebert, K.A. and B.V. Stratton, Supporting the Joint Warfighter by Development, Training and Fielding of Man-Portable UGVs, *SPIE Proc. 5804: Unmanned Ground Vehicle Technology VII,* Orlando, FL, March 2005.

[7] SPAWAR System Center San Diego, Man-Portable Robotic System, Retrieved March 7, 2007 from: http://www.nosc.mil/robots/land/mprs/mprs.html

[8] U.S. Army, Future Combat System. Retrieved May 11, 2007 from: http://www.army.mil/fcs/sugv.html

[9] Department of Defense,. *Report to congress: Development and Utilization of Robotics and Unmanned Ground Vehicles*, USA. October 2006.

[10] Ashley, J., FCS Update. *Unmanned Systems*, Vol 24, Nr 2, pp. 17-23, Mars/Apr 2006.

[11] Smith Detection, Smiths Supplies Lightweight Chemical Detector to Advanced CBRN Detection Robot, Retrieved February 5, 2007 from: http://www.smithsdetection.com/PressRelease.asp?autonum=114

[12] Foster-Miller, Inc., TALON Robots, Retrieved February 5, 2007 from: http://www.foster-miller.com/lemming.htm

[13] Gardner, C.W., Treado, P.J., Jochem, T.M., Gilbert, G.R., Demonstration of a Robot-based Raman Spectroscopic Detector for the Identification of CBE Threat Agents, *Proceedings of 25th Army Science Conference*, Orlando, USA, 2006.

[14] ROBOTIC FX, NEGOTIATOR, Tactical Surveillance Robot. Retrieved May 11, 2007 from: http://www.roboticfx.com/

[15] Murphy, R.R., Human-robot interaction in rescue robotics. *IEEE Systems, Man, and Cybernetics Part C: Applications and Reviews, special issue on Human-Robot Interaction*, Vol. 34, No. 2, pp: 125-137, May 2004.

[16] Hisanori, A., Present Status and Problems of Fire Fighting Robots, *Proceedings of Society of Instrument and Control Engineers Annual Conference*, Osaka, Japan, 2002.

[17] Matsuno, F., Tadokoro, S., Rescue Robots and Systems in Japan. In *Proceedings of IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, 2004.

[18] Scholtz, J., Young, J., Drury, J.L., and Yanco, H.A., Evaluation of human-robot interaction awareness in search and rescue. *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, New Orleans, USA, 2004.

[19] Yanco, H.. Drury, J., and Scholtz, J., Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition, Human-Computer Interaction Journal, No 1, January 2004.

[46] The level of acceptance vs. criticism to new gear might be influenced by cultural differences within the two organizations. The police has traditionally not had the recourses to finance custom development, but, been obliged to use COTS. The military, on the other hand, has a history of technical development according to their exact specifications.

[20] White, J., Harvey, H., Farnstrom, K., Testing of mobile surveillance robot at a nuclear power plant. Proceedings *of IEEE International Conference on Robotics and Automation*, Raleigh, NC, USA, 1987.

[21] Wada, K., Shibata, T., Saito, T., Sakamoto, K., Tanie, K., Robot assisted activity at a health service facility for the aged for 17 months: an interim report of long-term experiment. *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts*, Nagoya, Japan, 2005.

[22] Huttenrauch, H., Eklundh, K.S., Fetch-and-carry with CERO: observations from a long-term user study with a service robot. In *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, Berlin, Germany, 2002.

[23] Tomatis, N., Terrien, G., Piguet, R., Burnier, D., Bouabdallah, S. Arras, K.O., Siegwart, R., "Designing a secure and robust mobile interacting robot for the long term. In *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[24] Leger, P.C., Trebi-Ollennu, A., Wright, J.R., Maxwell, S.A., Bonitz, R.G., Biesiadecki, J.J., Hartman, F.R., Cooper, B.K., Baumgartner, E.T., Maimone, M.W., Mars Exploration Rover surface operations: driving spirit at Gusev Crater. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Waikoloa, HI, USA, 2005.

[25] Kumagai, J., Techno Cops - police robotic and electronic technology, *Spectrum, IEEE,* Vol 39, Issue 12, pp. 34–39, Dec 2002.

[26] Scheible, S., Robot helps end standoff: SWAT team uses new technology in arrest of Hingham man suspected of assault. Retrieved May 11, 2007 from: http://www.patriotledger.com/articles/2007/04/02/news/news01.txt

[27] Metacafe, Video: Robots Making Police Jobs Safer. Retrieved May 11, 2007 from: http://www.metacafe.com/watch/399031/robots_making_police_jobs_safer/

[28] Jones, H.L., Rock S.M., Burns D., & Morris, S., Autonomous robots in SWAT applications: Research, design, and operations challenges. In *Proceedings of AUVSI International Conference on Unmanned Vehicles*, Orlando, FL, USA, 2002.

[29] Nguyen, H., Bott, J., Robotics for Law Enforcement: Applications Beyond Explosive Ordnance Disposal, *SPIE Proc. 4232: Technologies for Law Enforcement*, Boston, MA, USA, November 2000.

[30] Ciccimaro, D., Baker, W., Hamilton, I., Heikkila, L., Renick, J., MPRS (URBOT) Commercialization. *Proceedings of SPIE Defence & Security Symposium*, Orlando, FL, USA, April 2003.

[31] Mesa-Robotics, MATILDA Robotic Platform. Retrieved May 11, 2007 from: http://www.mesa-robotics.com/matilda.html

[32] iRobot, Packbot. Retrieved May 11, 2007 from: http://www.irobot.com/sp.cfm?pageid=109

[33] Silverman, D., Interpreting Qualitative Data, SAGE Publications, London, Great Britain, pp.291, 2006.

[34] Lundberg, C., Reinhold, R., Christensen, H.I., Evaluation of robot deployment in live missions with the military, police, and fire brigade. *Proceedings of SPIE Defence & Security Symposium*, Orlando, FL, USA, 2007.

Performance Metrics and Evaluation of a Path Planner based on Genetic Algorithms

Giovanni Giardini
Department of Aerospace Engineering
Politecnico di Milano
Via La Masa 34
20156 Milano, Italy
giardini@aero.polimi.it

Tamás Kalmár-Nagy
Department of Aerospace Engineering
Texas A&M University
College Station
TX 77845, USA
permis@kalmarnagy.com

Abstract — This paper focuses on the analysis of the performance of an innovative genetic path planner designed for a single agent exploration. The proposed method is a generalization of the well-known Traveling Salesman Problem (TSP) that we call *Subtour* problem and it can be formulated as finding the shortest possible path for visiting a subset of n given targets over a known area. The algorithm is based on a Genetic Algorithm coupled with a heuristic local search method. To evaluate the proposed planner, an extensive performance evaluation has been done.

Keywords: *Genetic Algorithm, Traveling Salesman Problem*

I. Introduction

The capability of a system to plan and to act autonomously represents an important direction in the field of autonomy and artificial intelligence. Many applications, from space exploration [1]–[3] to search and rescue problems [4]–[7], have underlined the need of autonomous systems that are able to plan strategies both with or without the human feedback. Commonly, the main requirement for an autonomous vehicle is to navigate through an area while avoiding hazards, and, for search and rescue applications, determine the location and condition of victims. However, most of these systems can not be considered totally autonomous, since human instructions still play a crucial role.

In general, many autonomous vehicles are not provided with decision or planning capability. They are usually able to execute given commands (e.g. reaching a particular point or using a required instrument), but they are not able to decide by themselves a sequence of tasks or a plan to achieve. In other words, the mission strategy and the goals to accomplish are often decided by human operators.

Our goal is to realize a fully ‘autonomous’ system, where the autonomous navigation is linked with an autonomous planning and scheduling system onboard the vehicle [8], [9]. Our system should give to the vehicle the capability of deciding by itself what to do, allocating the goals to accomplish and generating a feasible strategy for achieving them.

As a first step toward this complete autonomy, we propose an innovative planner for finding the near-optimal strategy that allows the vehicle to accomplish a *given* set of interesting targets (or mission objectives) in the shortest amount of time.

In this work we consider the mission objectives as interesting locations displaced over a known map, and ‘strategy’ means a path -a sequence of targets- for visiting these locations (or a subset of these). As a consequence, our planning problem is closely related to the well-known Traveling Salesman Problem (TSP) [10]–[12], where an agent (the salesman) has to visit a given set of n targets such that each target is visited exactly once, with the additional constraint that the salesman will need to return to its starting locations.

More generally, our planning problem is stated as a *variant* of the TSP that we call *Subtour* problem: for a given set of n targets, the agent has to find the shortest path (the Subtour) that visits k targets out of the n possible ones.

Our method is based on an innovative Genetic Algorithm [13]–[15] coupled with heuristic local search techniques. The environment (in terms of targets displacement) is represented using a graph theory approach ([16] and [17]) and the proposed Genetic Algorithm is designed for searching optimal solutions over the resulting graph. Performance metrics of the algorithm is defined in terms of optimality of the solution and computational time.

The outline of the paper is as follows. After the introduction of the basic mathematical notations necessary for formalizing the combinatorial problems of interest (section II), the proposed Genetic Algorithm is described (section III). For evaluating the performance of the algorithm, an extensive campaign of tests have been conducted and the results are reported and discussed in section IV.

II. Mathematical formulation

Graph theory ([16] and [17]) has been widely used to describe vehicle routing problems [18]–[20] and therefore it is the natural framework for this study.

Given a set $V = \{v_1, \dots, v_m\}$ of m elements referred to as *vertices* (or targets) and a set $E = \{(v_i, v_j) | v_i, v_j \in V\}$ of *edges*, a graph G is defined as the pair (V, E). In particular, if the vertices in V are connected to each other, the graph is called *complete* (or fully connected) and it is denoted by $K_m(V)$, where $m = |V|$.

If a *weight* (or *cost*) $w(v_i, v_j)$ is assigned to every edge (v_i, v_j), G is a *weighted* graph. If $w(v_i, v_j) = w(v_j, v_i)$, G is also a *symmetric* graph.

In a graph, a *path* P is a sequence of edges with each consecutive pair of edges having a vertex in common. The *length* of a path is the number of its edges, and P_k is a path of length k. Similarly, a *cycle* C is a closed path, that starts and ends at the same vertex. The length of a cycle is the number of its edges and C_k is a cycle of length k.

The total cost W of a path P_k is the sum of the weights of its edges

$$W(P_k) = \sum_{i=1}^{k} w(x_i, x_{i+1}). \qquad (1)$$

Analogously, for a cycle C_k,

$$W(C_k) = \sum_{i=1}^{k-1} w(x_i, x_{i+1}) + w(x_k, x_1). \qquad (2)$$

After having introduced the necessary notation, we are now in the position to formalize the combinatorial problems of interest.

Let $T = \{t_1, \ldots, t_n\}$ be the set of n possible *targets* to be visited. The i-th target t_i is an object located in Euclidean space and its position is specified by the vector $\mathbf{r}(t_i)$. The agent is denoted by a and $\mathbf{r}(a)$ is its position.

Let us define the weighted and symmetric complete graph $K_{n+1}(V)$ generated by the augmented vertex set $V = T \cup a$ (see figure 1). The weights associated with the edges are given by the Euclidean distance between the corresponding locations, i.e. $w(v_i, v_j) = w(v_j, v_i) = \parallel \mathbf{r}(v_i) - \mathbf{r}(v_j) \parallel$, with $v_i, v_j \in V$.

The Subtour problem is now defined as finding a path $P_k \in K_{n+1}(V)$ of length k (that is also the number of targets to be visited), starting at vertex a and having the lowest cost $W(P_k)$.

The Traveling Salesman Problem (TSP) poses to find a cycle $C_{n+1} \in K_{n+1}(V)$ of minimal cost starting and ending at vertex a, that visits all the n targets once.

III. Solving combinatorial path planning problems with Genetic Algorithms

The obvious difficulty with the Subtour and the TSP is their combinatorial nature. In both cases, a brute force approach is infeasible for large n. A variety of exact algorithms (e.g. branch-and-bound algorithms and linear programming [21]–[23]) have been proposed to solve the classic TSP up to 30000 targets, and researchers have developed approximation methods based on evolutionary algorithms (e.g. Genetic Algorithms, Simulated Annealing, Ant System) for its solution [24]–[26]. These latter sacrifice the optimality for a near-optimal solution obtained in shorter time [27].

Our method is based on a *Genetic Algorithm* [13]–[15], and it has been implemented for solving the Subtour problem, as well as the classic TSP.

Briefly, a Genetic Algorithm (GA) is a search technique used to find approximate solutions of optimization and search problems [13]. Genetic Algorithms are a particular class of evolutionary methods that use techniques inspired by Darwin's theory of evolution and evolutionary biology, such as inheritance, mutation, selection, and crossover (also called recombination). In these systems populations of data compete and only the fittest survive.

In a GA, every possible solution is represented by a *chromosome* (also called plan or individual), which is a sequence of values (called *genes*). The algorithm works with *population* of candidate solutions (set of chromosomes).

A typical GA starts with a random population and at every generation step some individuals are chosen and mated. Mating is achieved through the use of *genetic operators* ([13] and [28]), described in section III-A. The newly generated chromosomes, the *offsprings*, are then inserted into the new population. Once a new population is created, its individuals are evaluated by a predefined *cost function* $c(.)$. The *fitness* is simply the reciprocal of the cost function. After this so-called *evaluation* phase the weakest (least fit) chromosomes are discarded. The GA tries to minimize the cost of the chromosomes by repeating the process of combining and modifying them according to a set of rules until desirable solutions are found. During this evolution phase, the number of genes composing the chromosomes could be either fixed or variable, depending on the analyzed problem.

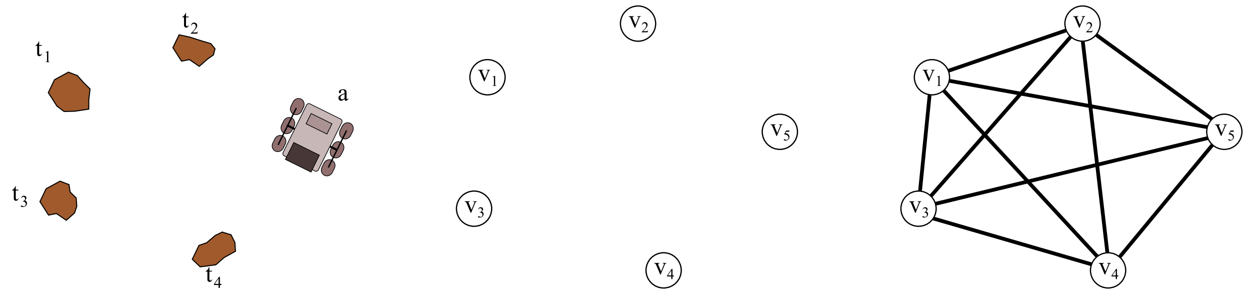
In this work, a Genetic Algorithm has been designed to solve the Subtour problem on the complete graph $K_{n+1}(V)$, where $V = T \cup a$, $T = \{t_1, \ldots, t_n\}$ is the set of n targets and a is the agent. The algorithm looks for the shortest possible path $P_k \in K_{n+1}(V)$ between $k \leq n$ targets starting from $\mathbf{r}(a)$.

Since the solutions of our problem are paths, chromosomes are easily coded as the sequence of targets of the path in the order they are visited by the agent (*order based representation*). Clearly, the first element of a chromosome is always a and targets must be visited only once.

A. Genetic Operators

The performance of a GA strongly depends on the recombination (mating) process, where the genetic materials of the chromosomes are combined. Clearly, depending on the genetic operators, the algorithm improves or reduces its speed of convergence (that is the number of generations necessary to converge toward the final solution) or its goodness (that is the fitness of the final solution).

Consequently, in order to better evaluate the performance of the algorithm, we decide to mainly focus on the description of the genetic operators, referring to [29] for a more detailed



(a) Targets and agent. (b) Vertex set $V = T \cup a$. (c) Complete graph $K_5(V)$.

Fig. 1. Given the set of targets $T = \{t_1, \ldots, t_4\}$ and the agent a, in (b) the augmented vertex set $V = \{v_1, \ldots, v_5\} = T \cup a$ is shown ($v_5 = a$), while (c) shows the complete graph $K_5(V)$ generated by the augmented vertex set V.

and complete description of the whole Genetic Algorithm.

Genetic operators combine existing solutions into new ones (crossover operators) or introduce random variations (mutation operators) to maintain genetic diversity. These operators are applied in a fixed order (shown in figure 2) with a priori assigned probabilities. In addition to these operators, a heuristic method to directly improve the fitness of the offsprings is introduced (see section III-A.4).

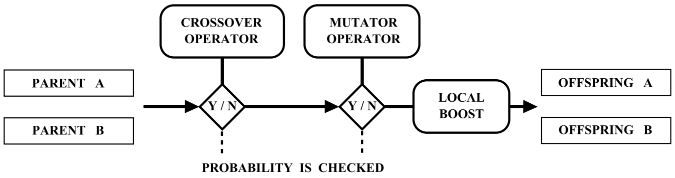


Fig. 2. Order of Application of the Genetic Operators. Depending on a given probability, at first the crossover operator is applied and then the mutation occurs. Note the last operator is the local heuristic boost method.

The crossover operator generates two new offsprings by combining the genetic materials of two chromosomes (the parents) opportunely chosen [29].

Different crossover typologies have been developed for solving the classic TSP, considering the order-based representation and the TSP constraints. The most common are those described in [13]: Partially Matched Crossover, Order Crossover and Cycle Crossover operators.

Even though operators are quite different, they are all based on the constraint that in a TSP the solutions include all the targets. However, since this is not the case for the Subtour problem, the application of these operators is inappropriate. Because of this, a new set of operators has been defined and two different crossover methods have been developed: the single cutting point crossover and the double cutting point crossover, described below.

1) Single cutting point crossover: with the single cutting point crossover, parents are halved at the same gene (the *cutting point*). The cutting point is chosen either randomly or to break the longest edge in the parents (the probability of which one of the two methods is applied is specified a priori). Once the parents have been halved, two offsprings are created combining the first (second) half of the first parent with the second (first) half of the second parent, respectively. Care is taken to avoid duplication of genes (as every target should only be visited once) and the length of the chromosomes is kept constant. See [29] for details.

2) Double cutting point crossover: the double cutting point crossover operator is similar to the previous one, but the parents could be now divided at two different genes. The cutting points are again selected in two ways, depending on pre-assigned probabilities: they are chosen either randomly or to cut the longest edge in both the parents. The deterministic selection method introduces an improvement over the single cutting point operator, where only one parent was cut along its longest edge and this point was also used for the other parent. An important consequence of having two different cutting points is that the halves will in general have different number of genes. A simple recombination would thus lead to two offsprings with different lengths. In order to maintain the original size of the chromosomes (which is necessary for producing feasible solutions), the offsprings are created in an appropriate way, described in [29].

3) Mutation operator: after the application of the crossover operator, the mutation operator is applied to the new chromosomes. The mutation operator generates a new offspring by randomly swapping genes and/or randomly changing a gene to another one not already present in the chromosome. Note that with the simple TSP this second type of mutation would not be possible, because for that problem a chromosome already contains all possible genes. The GA selects the method to apply with a given probability.

4) Improving offsprings: A common approach for improving the TSP solutions is the coupling of the Genetic Algorithm with a heuristic boosting technique. The local search method adopted here is the well-known *2-opt method* for the TSP [30]–[32]. This method replaces solutions with better ones from their 'neighbourhood'.

Let us consider a set T of n targets and the corresponding complete and weighted graph $K_{n+1}(V)$ (with $V = T \cup a$ and a the agent). Let us consider a Subtour $P_k \in K_{n+1}$, with $k \leq n$. The 2-opt method is based on the inequality

$$w(x_i, x_{i+1}) + w(x_j, x_{j+1}) > w(x_i, x_j) + w(x_{i+1}, x_{j+1}), \quad (3)$$

TABLE I

SIMULATION CASES TO TEST EFFICIENCY OF DIFFERENT GENETIC OPERATORS. THE 2-OPT METHOD IS NOT APPLIED

Crossover Type	Mutation	Mean fitness	Variance of fitness
Double point	**Applied**	**1**	**1**
Single point	Applied	0.93	1.32
Double point	Not applied	0.92	1.47
Single point	Not applied	0.69	2.12

where x_i, x_{i+1}, x_j and x_{j+1} are four vertices of P_k. If the inequality is satisfied, edges (x_i, x_j) and (x_{i+1}, x_{j+1}) are replaced with the edges (x_i, x_{i+1}) and (x_j, x_{j+1}), respectively.

This method provides a shorter path with no intersecting links. Consequently, the order of genes in the chromosome changes [33]. The computational cost of this method is proportional to the square of the number of the visited targets [31].

IV. RESULTS

A large number of simulations have been performed to test the performance of the implemented Genetic Algorithm. The tests described here are run for 250 generations with a population size of 200 chromosomes. The crossover and mutation operators are applied with a 90% and 30% probability, respectively.

A. *Influence of the 2-opt method on the performance of genetic operators*

To evaluate the performance of the different genetic operators and the 2-opt method, various tests have been performed. A target configuration for $n = 100$ targets randomly and uniformly distributed over a unit square map is generated. This configuration is kept fixed for all tests in this section to make comparisons meaningful. The Subtour problem is solved with $n = 30$ targets to be visited with a different combinations of the genetic operators and the 2-opt method.

First, to evaluate the importance of the various genetic operators, their performance are directly compared and tested without the 2-opt method. For the different cases of table I 100 simulations have been run and the mean values and the variances of the distribution of the best (highest) fitness values of the final populations are shown. Since the optimal solution is not known, the mean fitness values and the variances of fitness are normalized by the best result (the highest for the fitness values, the lowest for the variances of fitness). The comparison of the quantities in table I shows that the combined application of the double cutting point crossover and the mutation operator yields the maximum fitness value and the minimum variance of the solutions. On the other hand, the worst solutions are obtained with the standalone application of the single cutting point crossover operator. These results not only demonstrate the improvement introduced by the double cutting point crossover, but they clearly highlight the importance of the mutation operator (note the two best solutions have the mutation applied).

On the other hand, if the 2-opt method is applied, the results change, as shown in table II. In particular, independently of the adopted configurations of genetic operators, there is no difference between the fitness values of the obtained solutions. From these new results, it is clear that the main effect of the 2-opt method is in an overall improvement of the algorithm, which returns good solutions independently of the adopted operators.

TABLE II

SIMULATION CASES TO TEST EFFICIENCY OF DIFFERENT GENETIC OPERATORS TOGETHER WITH THE 2-OPT METHOD.

Crossover Type	Mutation	Mean fitness	Variance of fitness
Double point	Applied	0.993	3.12
Single point	Applied	1	1
Double point	Not applied	0.994	2.62
Single point	Not applied	0.998	1.31

B. *Speed of convergence and genetic operators*

The results of the previous section clearly demonstrate the great improvement introduced by coupling the genetic operators with the 2-opt method. However, even if the 2-opt method balances the results of the algorithm, its most important improvement is in the speed of convergence of the system.

To quantify the speed of convergence with various genetic operators and the 2-opt method, the required number of generations for the convergence of the algorithm and the associated computational time are compared.

For this purpose, the Genetic Algorithm was used to solve a 100-TSP problem, whose exact solution is known (the considered problem is the KroA-100 TSP [34]). For each different combinations of genetic operators, table III reports the number of generation (with its variance) necessary to reach within 1% of the known optimal solution. In every case, 500 simulations have been performed and the variances of the final results are normalized with respect to the minimum obtained value. From these results we conclude that the double cutting point crossover coupled with the mutation operator provides the highest speed of convergence, while a GA with the only single cutting point crossover needs more generation steps to reach close to the optimal solution.

For example, when the double cutting point crossover and the mutation operators are applied together with the 2-opt method, the converge of the GA is much faster than without (figure 3).

The improvement in speed is radical: for this example the local boosting technique yielded a 25-fold increase in computational speed to reach populations with the same fitness!

TABLE III

For each case, the 2-opt method is always applied and the number of generations necessary to converge toward the optimal solution is evaluated. The solved TSP is the KroA100 problem.

Crossover Type	Mutation	Normalized Number of generations	Variance
Double	**Applied**	**7.7**	**1**
Double	Not applied	8.7	1.14
Single	Applied	14.3	1.86
Single	Not applied	15.8	1.93

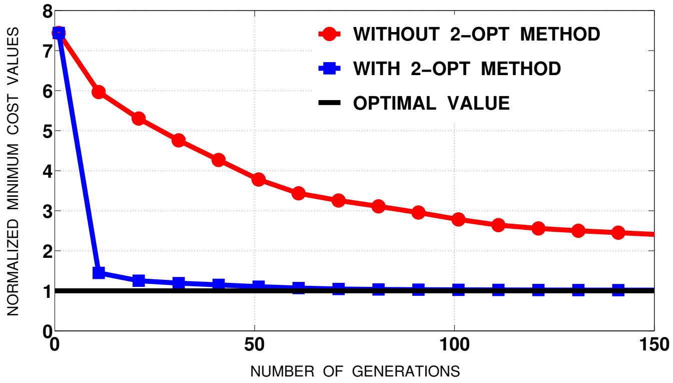


Fig. 3. The Subtour is solved with and without the implementation of the 2-opt method. With the application of the 2-opt method, the GA converges faster.

C. Subtour tests

The genetic path planning has been tested in order to demonstrate its capability to generate near-optimal Subtours. All the Subtour tests have been conducted considering a unit square map with a given target configuration and using the cost function (1). The double cutting point crossover, the mutation operator and the 2-opt method have been used. Moreover, to provide reliable averages, for a given configuration 100 simulations have been performed.

Figure 4 shows a sample Subtour for a problem where the total number of targets is $n = 100$ and the shortest path is sought connecting any $k = 20$ targets.

In order to rigorously evaluate the optimality of the Subtours generated by the Genetic Algorithm, a comparison with known optimal solutions is needed. To our knowledge, no benchmark solutions exist for the Subtour problem, so the simplest brute force approach was used to compute optimal paths for some suitably chosen test problems. Figure 5 shows the optimal $7-$Subtour for $n = 30$ targets and the solution by our Genetic Algorithm. Table IV summarizes the results for two Subtour problems, comparing the (brute force) optimal solution with those generated by the GA.

The error in the final solution can be attributed to the application of the 2-opt method on every chromosome at every generation step. In fact, the frequent use of this method restricts the random wandering of the Genetic Algorithm over the search space, thereby severely restricting the set of reachable solutions. If the 2-opt method is only applied with a given probability, much like the other operators, the results greatly improve and the optimal solution was easily found in almost all simulations.

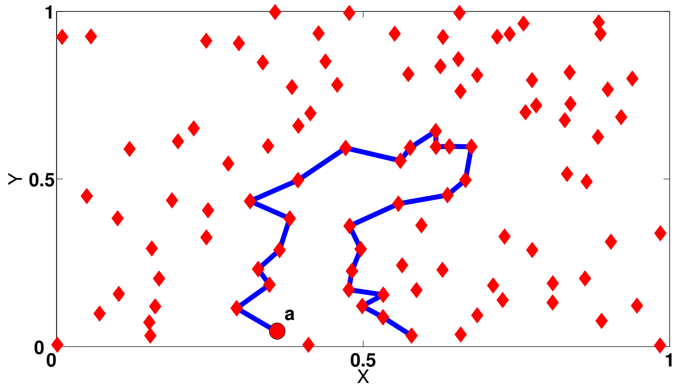


Fig. 4. 20-Subtour problem solution with a given set of 100 targets. The agent position is also shown.

TABLE IV

Comparison between exact and GA solutions for different Subtour problems based on 100 simulations.

Number of Targets	Subtour length	Optimal Solution	GA Solution	Error
30	7	71.27	72.42	1%
50	5	39.87	41.85	5%

D. TSP tests

Since the TSP is a limiting case of the Subtour problem (the agent visits all targets, i.e. $k = n$, with the restriction on returning to the starting position) the proposed algorithm can also be used to solve this classic problem. The double cutting point crossover, the mutation operator and the 2-opt method have already been applied. The suitable cost function is

$$\mathcal{W}(TSP) = w(x_{i+1}, x_1) + \sum_{i=1}^{n} w(x_i, x_{i+1}), \qquad (4)$$

where $w(x_i, x_j)$ is the Euclidean distance between the corresponding locations.

The algorithm has been tested with different TSPs whose optimal solutions have been published with the TSPLIB95 library [34]. This library includes different target configurations for the TSP and many related problems (Hamiltonian Cycle Problem, Sequential Ordering Problem, etc.) together with the exact solutions. We note that the TSPs in the TSPLIB95 library are solved with a cost function based on *rounded* distance between targets.

For every problem considered here, 100 simulations have been performed and the optimal solution was almost always reached. These results strengthen our claim that the

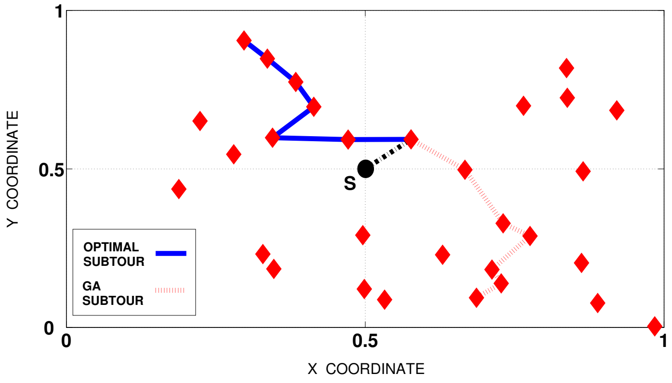


Fig. 5. Comparison between the exact 7-Subtour ($n = 30$ targets) and the GA solution.

implemented system is an efficient way to find near-optimal solutions of the proposed hard combinatorial problems.

E. Computational performance of the Genetic Algorithm

Finally, the computational performance of the proposed Genetic Algorithm is evaluated. With a fixed number of generations (200), different TSPs have been solved. In each problem, the targets are randomly distributed over a 100 square map. Every test has been conducted on a Pentium IV with a clock frequency of 1.86 GHz and 1 Gb of RAM, running the $GentooLinux$ Operation System.

For every configuration, 100 simulations have been run and the time t of each process has been recorded. Table V presents the mean values $\bar{t}$ and the standard deviation σ_t of the obtained results for the initialization and the evolution phases.

TABLE V
TABLE REPORTS THE MEAN TIME VALUES $\bar{t}$, TOGETHER WITH THE CORRESPONDING STANDARD DEVIATION σ_t, SPENT FOR THE INITIALIZATION AND THE EVOLUTION PHASES.

Number of	Initialization Phase		Evolution Phase	
Targets	$\bar{t}$ [s]	σ_t [s]	$\bar{t}$ [s]	σ_t [s]
50	0.23	0.006	4.97	0.01
100	1.02	0.015	14.23	0.07
150	2.46	0.032	29.38	1.36
200	4.57	0.054	46.8	1.05
250	7.39	0.086	77	5.9
300	10.95	0.12	144.22	9.87
500	33.15	0.27	482.54	57

Not surprisingly, the initialization phase takes up less than 10% of the total computational time.

Focusing on the evolution phase, it is possible to evaluate the performances considering the total time spent during the 200 iterations by each single operation. These results are shown in table VI. Together with these results, we also report the performance of the evaluation phase and of all the other operations [29] required by the genetic evolution (labeled as 'other operations').

TABLE VI
THE MAIN STEPS OF THE EVOLUTION PHASE HAVE BEEN EVALUATED AND THE MEAN TIMES $\bar{t}$ SPENT BY THE GA FOR EACH ONE ARE HERE REPORTED.

Number of Targets	100	200	250	300	500
Others operations	0.3	0.39	0.47	0.54	0.82
Genetic operators	3.48	13.11	29.1	66.88	251.36
2-opt method	5.56	24.4	36.6	63.66	209
Evaluation phase	4.57	8.4	10.3	12.45	20.22

These results show that the most time-expensive part of the algorithm is the application of the genetic operators and the 2-opt method. Here the time spent increases exponentially with the number of targets. On the other hand, the time required by the evaluation phase and the other operations depend linearly on the number of targets. Figure 6 shows the log-plot of time required vs. target number.

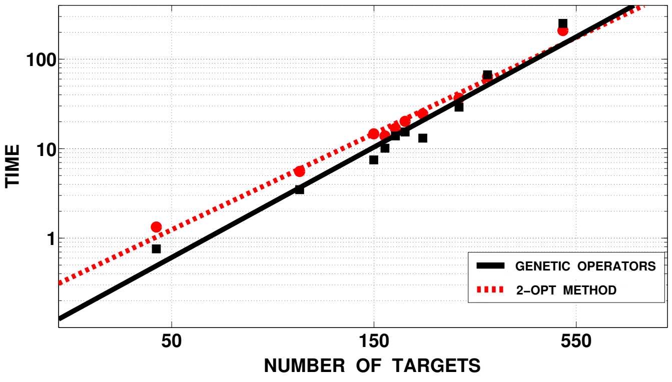


Fig. 6. Performance of the GA: the logarithms of the time spent for the application of the genetic operators and the 2-opt method are plotted with respect to the logarithm of the number of targets.

V. CONCLUSIONS

This work describes an innovative Genetic Algorithm path planner for generating a near-optimal multi-agent strategy for visiting a set of known targets. The method is based on the solution of an NP-hard combinatorial problem similar to the classic TSP, the Subtour problem.

The importance of this work is in the ability of the agent to plan a strategy -a Subtour- by organizing a sequence of targets autonomously. The proposed system finds application in problems where there is limited/no human feedback (like planetary space exploration or search and rescue problems, in collapsed buildings). For these kind of missions a high level of autonomy is required and an efficient planner is a crucial ingredient for autonomous vehicles for ground applications and space exploration.

The results presented here show the success of the implemented Genetic Algorithm, both for the simple Subtour problem and the classic TSP.

References

[1] http://marsrovers.nasa.gov/home/, *Mars Exploration Rover Missions*.

[2] S. Hayati and R. Volpe, "The rocky 7 rover: A mars sciencecraft prototype," in *IEEE International Conference on Robotics and Automation*, Albuquerque, 1997.

[3] E. T. Baumgartner, "In-situ exploration of mars using rover systems," in *Proceedings of the AIAA Space 2000 Conference*, Long Beach, CA, USA, September 2000.

[4] A. Birk, H. Kenn, S. Carpin, and M. Pfingsthorn, "Toward autonomous rescue robots," *Proceedings of the First International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disasters, Padova, Italy*, 2003.

[5] S. Sariel and H. Akin, "A novel search strategy for autonomous search and rescue robots."

[6] A. Jacoff, E. Messina, and J. Evans, "Experiences in deploying test arenas for autonomous mobile robots," *NIST Special Publication*, pp. 87–94, 2002.

[7] S. Carpin, J. Wang, M. Lewis, A. Birk, and A. Jacoff, "High fidelity tools for rescue robotics: results and perspectives," *RoboCup 2005: Robot Soccer World Cup IX*.

[8] R. Sherwood, A. Mishkin, S. Chien, T. Estlin, P. Backes, B. Cooper, G. Rabideau, and B. Engelhardt, "An integrated planning and scheduling prototype for automated mars rover command generation," NASA, Jet Propulsion Laboratory, California Institute of Technology, May 2001.

[9] K. H. Low, K. W. Leow, and H. M. Ang, "A hybrid mobile robot architecture with integrated planning and control," in *International Conference on Autonomous Agents*, Bologna, 2002.

[10] D. S. Johnson and L. A. McGeoch, *The traveling salesman problem: A case study in local optimization*, local search in combinatorial optimization ed., E. H. L. Aarts and J. L. (eds), Eds. John Wiley and Sons Ltd, 1997.

[11] http://www.tsp.gatech.edu/, *Traveling Salsman Problem*.

[12] G. Gutin and A. Punnen, *Traveling Salesman Problem and Its Variations*. Norwell, MA: Kluwer Academic Publishers, 2002. [Online]. Available: citeseer.ist.psu.edu/gutin02traveling.html

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley Professional, January 1989. [Online]. Available: http://www.amazon.co.uk/exec/obidos/ASIN/0201157675/citeulike-21

[14] K. Bryant, "Genetic algorithm and traveling salesman problem," 2000.

[15] K. Katayama and H. Narihisa, "An efficient hybrid genetic algorithm for the traveling salesman problem," *Electronics & Communications in Japan, Part 3: Fundamental Electronic Science (English translation of Denshi Tsushin Gakkai Ronbunshi)*, vol. 84, no. 2, pp. 76–83, 2001.

[16] J. Bondy and U. Murty, *Graph Theory with Applications*. Macmillan London, 1976.

[17] R. Diestel, *Graph Theory*. Springer, 2005.

[18] http://neo.lcc.uma.es/radi aeb/WebVRP/, *The VRP Web*.

[19] *Vehicle Routing Problem: Doing it the Evolutionary Way*, ser. GECCO-2002 Proceedings, San Francisco, CA, USA, 2002.

[20] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem," *Mathematical Programming*, vol. 94, no. 2, pp. 343–359, 2003.

[21] S. Tschoke, R. Luling, and B. Monien, "Solving the traveling salesman problem with a distributed branch-and-bound algorithm on a 1024 processor network," *Proceedings of the 9th International Symposium on Parallel Processing*, pp. 182–189, 1995.

[22] E. Balas, P. Toth, C.-M. University, and D. R. Center, *Branch and Bound Methods for the Traveling Salesman Problem*. Carnegie-Mellon University, Design Research Center, 1983.

[23] A. Schrijver, *Theory of linear and integer programming*. John Wiley and Sons, Inc. New York, NY, USA, 1986.

[24] A. E. Carter, "Design and application of genetic algorithms for the multiple traveling salesperson assignment problem," Ph.D. dissertation, Department of Management Science and Information Technology, Virginia Polytechnic Institute and State University, 2003.

[25] M. Ohta, "An implementation of rescue agents with genetic algorithm," in *Second International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED 2004)*, Lisbon Industry Fair (FIL), Lisbon, Portugal, June-July 2004.

[26] M. Kulich, J. Kubalik, J. Klema, and J. Faigl, "Rescue operation planning by soft computing techniques," in *IEEE 4th International Conference on Intelligent Systems Design and Application, Budapest, Hungary*, 2004, pp. 103–109.

[27] S. Thangiah, *Vehicle Routing with Time Windows using Genetic Algorithms*, L. C. (Ed), Ed. CRC Press, 1995, vol. 2. [Online]. Available: citeseer.ist.psu.edu/thangiah95vehicle.html

[28] M. Rocha and J. Neves, "Preventing premature convergence to local optima in genetic algorithms via random offspring generation," in *Proceedings of the 12th international conference on Industrial and engineering applications of artificial intelligence and expert systems: multiple approaches to intelligent systems*. Cairo, Egypt: Springer, 1999, pp. 127–136.

[29] G. Giardini and T. Kalmar-Nagy, "Genetic algorithm for combinatorial search problems," in *SSRR 2006, IEEE International Workshop on Safety, Security and Rescue Robotics*, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, August 2006, pp. 22–24.

[30] M. Matayoshi, M. Nakamura, and H. Miyagi, "A genetic algorithm with the improved 2-opt method," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, 2004.

[31] J. L. Bentley, "Experiments on traveling salesman heuristics," in *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1990, pp. 91–99.

[32] H. Sengoku and I. Yoshihara, "A fast tsp solver using ga on java," in *Third International Symposium on Artificial Life, and Robotics (AROB III'98)*, 1998.

[33] J. Watson, C. Ross, V. Eisele, J. Denton, J. Bins, C. Guerra, D. Whitley, and A. Howe, "The traveling salesrep problem, edge assembly crossover, and 2-opt," in *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*. Amsterdam, Netherlands: Springer, 1998, pp. 823–834.

[34] G. Reinelt, "Tsplib: A traveling salesman problem library," in *ORSA Journal on Computing*, vol. 3, no. 4, 1991, pp. 376–384, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

The Evolution of Performance Metrics in the RoboCup Rescue Virtual Robot Competition

Stephen Balakirsky & Chris Scrapper
NIST
100 Bureau Drive
Gaithersburg, MD, USA

Stefano Carpin
University of Calif. Merced
5200 North Lake Rd
Merced, CA, USA

***Abstract*—**This paper presents an overview of the 2007 RoboCup Rescue Virtual Robot Competition and the performance metrics that were used to judge the competition. For this competition, great effort was placed in bringing together researchers with diverse interests to competitively participate. The competition arenas and metrics used for scoring were specifically designed to create a "level" playing field for the various research disciplines. The specific metrics, how they evolved from the prior year's competition, and the way in which the competition was run will be discussed in detail. Defects that were noted in the metrics will also be discussed.

***Keywords*:** *robotics, competition, simulation, performance metrics, RoboCup*

I. Introduction

July 2007 saw the second annual running of the RoboCup Rescue Virtual Robot Competition in Atlanta GA. Robocup [1] provides an international forum where researchers meet to compete against each other in robotic competitions ranging from soccer to dance to urban search and rescue (USAR). Underlying these competitions are basic research thrusts focusing on core robotic technologies such as mobility, multi-agent cooperation, and fine motor control.

This year's USAR virtual robot competition consisted of 9 runs over 7 days and took place in complex indoor and outdoor domains. The scoring performance metrics were specifically designed to award research advances in the general areas of multi-agent cooperation, human-computer interfaces (HCI), and map building. Specific emphasis was placed on the formation of multi-agent communication networks, complex terrain navigation, and victim search and identification strategies. The use of *a priori* data and carefully constructed worlds allowed the researchers to concentrate their efforts in one or more research areas while maintaining competitiveness among groups performing in different research areas. Examples of the indoor and outdoor worlds that were used for the competition are shown in Figure 1 and Figure 2 respectively.

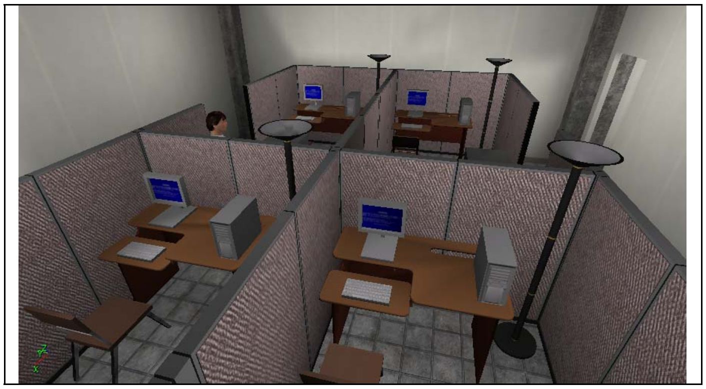


Figure 1: Example of the cubicle area from the indoor environment used in the RoboCup 07 competition.

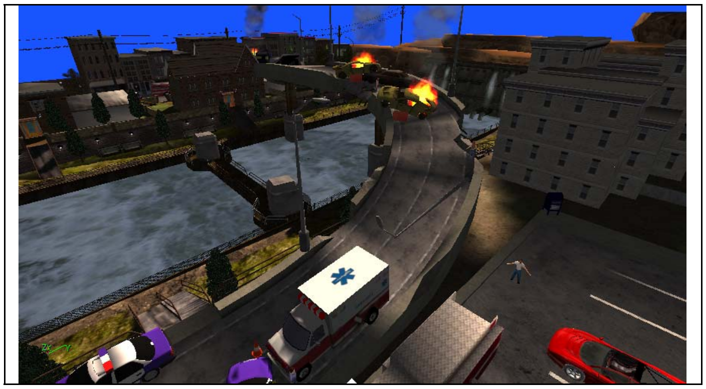


Figure 2: Example of the bridge accident scene from the outdoor environment used in the RoboCup07 competition.

II. Background

A. USARSim

The current version of Urban Search and Rescue Simulation (USARSim)[3] is based on the UnrealEngine2[1] game engine that was released by Epic Games as part of

[1] Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

Unreal Tournament 2004. This engine may be inexpensively obtained by purchasing the Unreal Tournament 2004 game. The USARSim extensions may then be freely downloaded from sourceforge.net/projects/usarsim. The engine handles most of the basic mechanics of simulation and includes modules for handling input, output (3D rendering, 2D drawing, and sound), networking, physics and dynamics. USARSim uses these features to provide controllable camera views and the ability to control multiple robots. In addition to the simulation, a sophisticated graphical development environment and a variety of specialized tools are provided with the purchase of Unreal Tournament.

The USARSim framework builds on this game engine and consists of:

- standards that dictate how agent/game engine interaction is to occur,
- modifications to the game engine that permit this interaction,
- an Application Programmer's Interface (API) that defines how to utilize these modifications to control an embodied agent in the environment ,
- 3-D immersive test environments,
- models of several commercial and laboratory robots and effectors,
- models of commonly used robotic sensors

USARSim does not provide a robot controller. However, several open source controllers may be freely downloaded. These include the community-developed MOAST controller (sourceforge.net/projects/moast), the player middleware (sourceforge.net/projects/playerstage), and any of the winning controllers from previous year's competitions (2006's winning controllers may be found on the Robocup Rescue wiki at: www.robocuprescue.org/wiki/). A description of the winning algorithms may be found in [2].

B. RoboCup Virtual Robot Competition

RoboCup is an annual competition that was held in 2007 in Atlanta, GA. Nearly 300 teams from 33 countries participated. The virtual robot competition (VRC) is part of the RoboCup Rescue Simulation League. The VRC, which this year saw its second annual running, is designed to foster collaboration and competition between research groups conducting research in the diverse areas of human-computer interfaces, map building, the formation of multi-agent communication networks, complex terrain navigation, and victim search and identification strategies. The competition was run over 7 days and consisted of two preliminary pass/fail rounds followed by three main competition rounds, 2 semi-final rounds, and 2 final rounds.

The preliminary rounds of the competition were designed to verify that teams met a minimum set of competencies. Teams needed to control their robots through the use of a provided communications server (a new requirement for this year in order to mimic the non-line-of-sight nature of a real disaster location), generate maps and find victims, and provide the judges with maps and victim locations that were in the proper format. Eight teams from five different countries participated in the preliminary rounds. All of the teams passed and moved onto the actual competition.

The competition rounds consisted of extensive indoor or outdoor terrain. The goal of the competition was to find as many victims while clearing as much area as possible before the batteries of the robot expired. Robots were given a battery life of approximately 20 minutes. In order to support a wide variety of research interests and lower the competition entry barriers by assuring that teams did not need to be experts in all fields, *a priori* data was provided on the difficulty of terrain traversal, the difficulty of communicating with the base station, and the difficulty of finding victims. Each of these areas had three levels of difficulty as defined as:

Mobility:
Easy – Flat floors
Moderate – Sloped floors, rolling areas, narrow passageways, small steps
Difficult – Stairs, rough terrain, drops and holes that can damage the robots

Communications:
Easy – Use of communications server required
Moderate – No direct communication between robot and base station
Difficult – No direct communication to base station and robot is prevented from reentering moderate or easy communication area

Victim Finding:
Easy – Static, exposed victims with minimum false alarms
Moderate – Dynamic (moving limbs), partially occluded, many false alarms
Difficult – Dynamic, significant victim occlusion (entombed or hidden), many false alarms

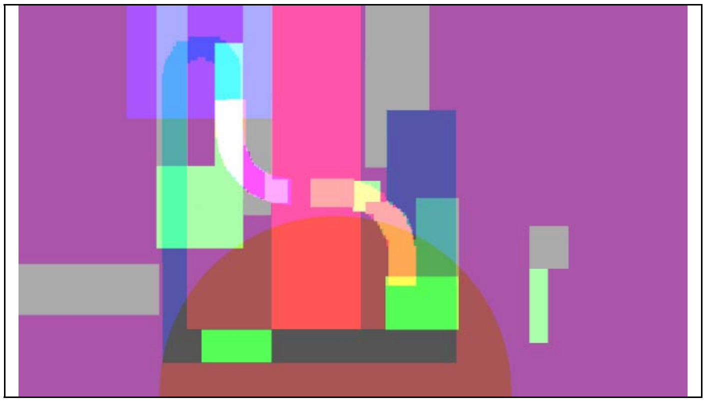


Figure 3: Example of a priori data from competition.

To balance the possible points that were awarded, an approximately even number of points were available in each area. For example, if we form the tuple (mobility difficulty, communications difficulty, victim difficulty), then the area covered by (Moderate, Easy, Easy) would have the same points available as the area (Easy, Moderate, Easy), and (Difficult, Easy, Easy). This allowed for teams with higher levels of competency or multiple competencies to score more points.

An example of the composite *a priori* data is shown in Figure 3 where mobility information is encoded in red, victim information is encoded green, and communication information is encoded in blue. The color ranges were from 0-255 with 85 being easy, 170 being moderate and 255 being difficult. In the figure, larger values for colors appear more saturated.

III. Performance Metric

The primary goals of the competition are to report the location of victims in the environment and to form accurate, attributed maps of the explored area. These two distinct areas have separate techniques that are used for judging competency, and the performance metrics utilized have evolved.

A. Victim finding

Since one of the primary goals of the competition is to locate victims (and in 2006 to determine the victim's health status), a technique for determining a team's competency needed to be developed. However, what does it mean to "locate" a victim? How does one autonomously obtain health status? Several possible interpretations exist ranging from simply requiring a robot to be in proximity of a victim (e.g. drive by the victim) to requiring the robot to employ sensor processing to recognize that a victim is located nearby (e.g. recognize a human form in a camera image) and then examine that victim for visually apparent injuries. While recognizing a human from a camera image is the solution most readily portable to a real hardware, it places an undue burden on both the competitors and the evaluation team. For the competitors, a robust image processing system would need to be developed that could recognize occluded human forms. No matter how exceptional the mapping and exploration features of a team were, failing to produce the image processing module would result in a losing effort. In addition, the evaluation team would need to develop an entire family of simulated human forms so that teams could not "cheat" by simply template matching on a small non-diverse set of victims.

It was decided that robots should be required to be "aware" of the presence of a victim, but that requiring every team to have expertise in image processing was against the philosophy of lowering entry barriers. Therefore, a new type of sensor: a victim sensor, was introduced. To allow for the metrics to be portable to real hardware, this new sensor would need to be based on existing technology.

For the 2006 competition, the victim sensor was based on Radio Frequency Identification Tag (RFID) technology. False alarm tags were scattered strategically in the environment, and each victim contained an embedded tag. At long range (10 m), a signal from the tag was readable when the tag was in the field of view (FOV) of the sensor. At closer range (6 m), the sensor would report that a victim or false alarm was present. At even closer range (5 m) the ID of the victim would be reported. Finally, at the closest range (2 m), the status of the victim (e.g. injured, conscious, bleeding, etc.) was available. Points were subtracted for reporting false alarms, and were awarded for various degrees of information collected from the victims. Bonus points were awarded for including an image of the victim with the report. This technique worked well for scoring the 2006 competition. However, several deficiencies were noted with this sensor system:

- The RFID tag was located in the victim's torso and operated on a line-of-sight basis. Therefore, it was impossible to have largely occluded victims.
- The operation of the sensor encouraged teams to drive quickly through the environment and did not require any user input or additional behaviors when a victim was located.
- While the sensor was based on existing technology (RFID tags), no actual victim locating system works in this way.

To rectify these problems, the victim sensor was significantly revamped for the 2007 competition. The new sensor is modeled after template based human form detection. The sensor performs a line-of-sight calculation to the victim and reports which of the 7 body parts identified in Figure 4 are visible. In the right side of the figure, the points represent the possible sensor hit-points. Yellow points are non-victim, and green points represent victim hits. The worlds also contained false alarms that would be consistent with a template matching algorithm.

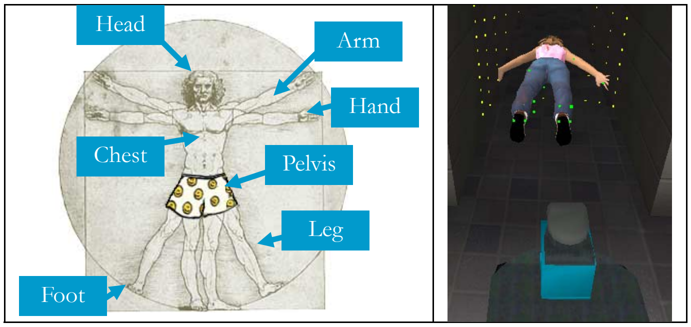


Figure 4: New victim sensor based on template matching of body parts.

The new sensor configuration required teams to attempt to gather multiple body parts from a victim (or have user involvement) in order to make a victim/false alarm determination. This usually required teams to pause upon finding a victim location in order to either alert an operator or to conduct a scan in an effort to find more body parts.

B. Map building

While knowing that a victim is located inside of a structure is useful, having a map of where this victim is located adds even more utility. Therefore, building a map of the environment is a basic requirement of the competition and performance metrics were developed to evaluate the maps. A major change between the 2006 and 2007 competitions was the requirement that all maps be delivered as geo-registered images with specific color mappings or vector files. This allowed the judges to directly compare competitor's maps to ground truth using geographic information services (GIS) software. The map quality score is based on several components; most of which have evolved from 2006 to 2007.

- Feature quality – In 2006, there was no technique available to overlay team generated maps with ground truth. Therefore, the feature quality of a map was scored automatically by examining the reported locations of "scoring tags". Scoring tags were single shot RFID tags (they could only be read once). A requirement of the competition was for the teams to report the global coordinates of these tags at the conclusion of each run. The automatic scoring program then analyzed the deviation of the perceived locations from the actual locations. The use of these tags had the undesirable result that errors occurring early in a run were penalized more than late errors (the error affected the locations of a greater number of tags). In 2007, feature quality was evaluated subjectively. As shown in Figure 5, geo-registered maps were overlaid on ground truth and were examined for the number of discrete errors. For example, on some maps it was obvious that a single error led to a piece of the map being rotated. False obstacle reports (a single wall being reported in multiple locations) and scaling issues were also noted. The maps were ranked from best to worst and then assigned points based on their ranking.

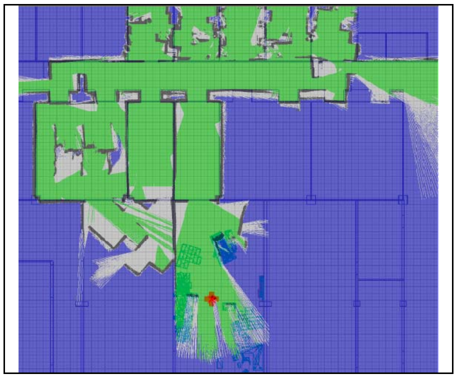


Figure 5: Competitors map overlaid on ground truth from an indoor scenario.

- Multi-vehicle fusion – Teams were only permitted to turn in a single map file. Those teams that included the output from multiple robots in that single map were awarded bonus points. This metric did not change between 2006 and 2007.
- Attribution – One of the reasons to generate a map is to convey information. This information is often represented as attributes on the map. In 2006, points were awarded for including information on the location, name, and status of victims, the location of obstacles, the paths that the individual robots took, and the location of RFID scoring tags. For 2007, teams were required to denote areas explored (gray color on map examples), areas cleared of victims (green color on map examples), and victim locations. The competition definition of cleared meant that no undetected victims exist in that area. Therefore, teams received penalties for any victims that were located in "cleared" areas and that were not reported. Teams were free to include any additional map attributes that they found useful. The best teams had explored space, cleared space, vehicle paths, victim locations, geo-registered victim images, names of grouped areas, confidence in information, and more. An example of an annotated map is shown in Figure 6. Points were once again awarded based on a rank ordering of the maps.

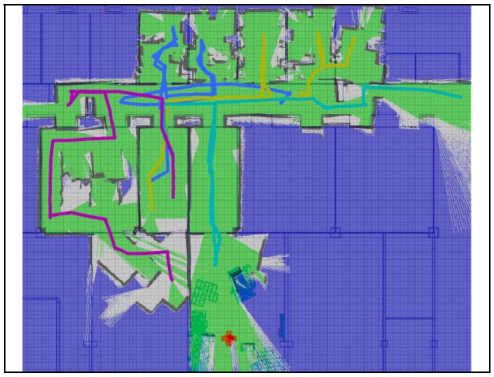


Figure 6: Annotations on map include area explored (gray), area cleared (green), victims located (red cross), and robot paths (multi-colored lines).

- Grouping – A higher order mapping task is to recognize that discrete elements of a map constitute larger features. For example the fact that a set of walls makes up a room, or a particular set of obstacles is really a car. Bonus points were awarded for annotating such groups on the map. An example of such groupings is shown in Figure 7. This metric did not change between 2006 and 2007.
- Skeleton quality – A map skeleton reduces a complex map into a set of connected locations. For example, when representing a hallway with numerous doorways, a skeleton may have a line for the hallway and symbols along that line that represent the doors. A map may be inaccurate in terms of metric measurements (a hallway may be shown to be 20 m long instead of 15 m long), but may still present an accurate skeleton (there are three doors before the room with the victim). The category allowed the judges to award points based on how accurately a map skeleton was represented. This metric did not change between 2006 and 2007.

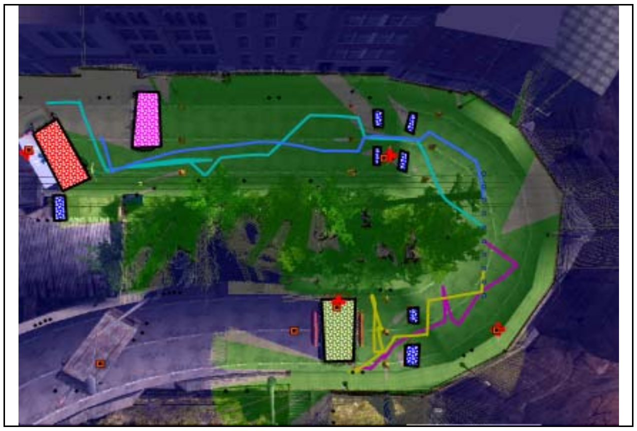


Figure 7: Example of fully annotated and group map. The colored rectangles are keyed to various groups (ambulance, barrier, etc.).

- Utility – One of the main objectives of providing a map was to create the ability for a first responder to utilize the map to determine which areas had been cleared, where hazards may be located, and where victims were trapped. Points were granted by the judges that reflected their feelings on this measure. This metric did not change between 2006 and 2007.

The above mentioned elements were numerically combined according to Equation 1 for 2006 and Equation 2 for 2007.

$$S = \frac{V_{ID} * 10 + V_{ST} * 10 + V_{LO{*}10} + t * M + E * 50 - C * 5 + B}{(1+N)^2} \quad (1)$$

$$S = \frac{V_{ID} * 5 + V_P * 5 + V_{IP} * 10 + M + E - C * 5 - FA * 5 - V_M * 5}{N^2} \quad (2)$$

The meaning of the variables is discussed below. This equation represents a schema that took into account merit factors that concerned (1) victims discovery, (2) mapping, and (3) exploration. The exact point calculations for each factor are presented below.

1. For victims in 2006, 10 points were awarded for each reported victim ID (V_{ID}). An additional 10 points were granted if the victim's status (V_{ST}) was also provided, and properly localizing the victim in the map was rewarded with an additional 10 points (V_{LO}). Also, at the referee's discretion, up to 20 bonus points were granted for additional information produced (B). For example, some teams managed to not only identify victims, but to also provide pictures taken with the robot's cameras. For this additional information teams were awarded with 15 bonus points. Taking a picture of a victim seemed like a really useful item. Therefore, in 2007, this became a part of the scoring metric (V_P) that was worth 5 points per victim. Correctly geo-referenced victims were worth 5 points if found using the victim sensor (V_{ID}), and 10 points if found using image processing (V_{IP}).
2. Maps were awarded up to 50 points based on their quality (M), as previously described. For the 2006 competition, the obtained score was then scaled by a factor ranging between 0 and 1 (t) that measured the map's feature accuracy. This accuracy was determined through the use of the RFID scoring tags.
3. Up to 50 points were available to reward exploration efforts (E). During the 2006 competition, as the robots were exploring the environment, their poses (on 1 s intervals) were logged. Using the logged position of every robot, the total amount of explored square meters (m^2) was determined and related to the desired amount of explored area. This desired amount was determined by the referees and was based on the competition environment. For example, in a run where 100 m^2 were required to be explored, a team exploring 50 m^2 would receive 25 points, while a team exploring 250 m^2 would receive 50 points, i.e. performances above the required value were leveled off. While this metric was easy to automatically compute, it seemed to reward teams for passing through a location as opposed to actually performing any behaviors while in the location. Therefore a major change was instituted for the 2007 competition.

   For 2007, teams needed to declare where they had explored and where they had cleared. Any victims that existed in a cleared area and were that were not reported by the teams were assessed penalties. The idea being that a map of the environment is useful to responders (therefore award points), and knowing where they do not have to look for victims is even more useful (so award more points). Points were awarded based on a linear scale ranging from 0 – 35 for area cleared and 0 – 15 for area explored. The amount of area that received a top score was the average of the top performing two teams. Exploration above this cutoff was not awarded with additional points. The amount of area explored and cleared by each team was automatically computed based on their maps.

On the penalization side, 5 points were deducted for each collision between a robot and a victim (C). The number of collisions was automatically determined. For 2007, false alarms reported as victims (FA) and victims missed (V_M) in the cleared areas also caused point deductions.

Another parameter that was used to determine the overall score was the number of human operators that were needed to control the robots. The idea was borrowed from the Rescue Robot league with the intent of promoting the deployment of fully autonomous robot teams, or the development of sophisticated human-robot interfaces that allow a single operator to control many agents. In 2006, the overall score was divided by $(1 + N)^2$, where N was the number of operators involved. So, completely autonomous teams, i.e. N=0, incurred no scaling, while teams with a single operator had their score divided by 4. No team used more than one operator. However, for 2007 it was decided that there is no such thing as a truly operator-less team. At a minimum, an operator must be available to deploy the robots and provide routine maintenance. Therefore, each team was allowed a single operator without a scaling factor.

B. After Action Evaluation

In addition to the scores that teams received during the competition, a large volume of real-time data was logged for post analysis. This information included the actual pose of every robot on a 1 s interval, and a recording of all of the runs. The hope is that teams will be able to combine this information with the environment's ground truth in order to learn from the competition experience.

IV. Future Work

The RoboCup rescue virtual robot competition community remains very active and plans are already underway for the 2008 competition which will take place in Suzhou China. While further evolution of the metrics is inevitable, the main thrust for this year is the automation of the scoring process. Currently, robot-victim bumps are automatically computed as well as the area explored and the area cleared. However, judging the map quality is a manual process. A process that compares competitor generated maps to ground truth and scores map accuracy and utility is an active area of research.

References

1. Asada, M. and Kitano, H., *RoboCup-98: Robot Soccer World Cup II*, Springer-Verlag, Berlin, 1999.

2. Balakirsky, S., Carpin, S., Kleiner, A., Lewis, M., Visser, A., Wang.J., and Ziparo, V., "Towards Heterogeneous Robot Teams for Disaster Mitigation: Results and Performance Metrics from RoboCup Rescue," *Journal of Field Robotics*, Vol. SUBMITTED, 2007.

3. Balakirsky, S., Scrapper, C., Carpin, S., and Lewis, M., "USARSim: Providing a Framework for Multi-robot Performance Evaluation," *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, 2006.

Robot Simulation Physics Validation

C. Pepper, S. Balakirsky, and C. Scrapper
Intelligent Systems Division, National Institute of Standards and Technology (NIST)
Gaithersburg, MD 20899-8230, USA
Email: {Christopher.Pepper, Stephen.Balakirsky, Christopher.Scrapper} @NIST.Gov

***Abstract* —** Computer simulation of robot performance is an essential tool for the development of robot software. In order for simulation results to be valid for implementation on real hardware, the accuracy of the simulation model must be verified. If developers use a robot model that is not similar enough to the actual robot, then their results can be meaningless. To ensure the validity of the robot models, NIST proposes standardized test methods that can be easily replicated in both computer simulation and physical form. The actual robot can be tested, and the computer model can be finely tuned to replicate similar performances on equivalent tests. To illustrate this, we have accomplished this task with the Talon Robot[1] on NIST standard test methods..

***Keywords*:** *performance metrics, physics validation, response robots, robots, simulation, urban search and rescue, USARSim*

I. INTRODUCTION

There is a growing trend in intelligent systems research to use a simulated environment in the initial phases of development. As simulations become more integral in the development process, it is important for them to become more accurate to protect the validity of experiments. The solution to this is to develop standard test methods [1] and validate the performance of the robot on the test methods in both reality and simulation [2].

A. The Benefits of Computer Simulation

There are a myriad of benefits to computer simulation for a researcher that make it an attractive option during the development process. An important attribute of simulations to a developer is repeatability, which allows for simplified debugging because the same scenario can be precisely generated to trigger a known error and check the solution. In addition to this, all vital data can be logged, including ground truth, to give developers an understanding of inconsistencies in their algorithm performance. In contrast to an actual environment, simulation gives developers access to cost prohibitive or unavailable sensors. Time can also be spent efficiently since many researchers can work on copies of a virtual platform simultaneously where physical platforms may be limited in availability. Additionally, the actual testing environment may not be accessible, or may only be accessible at certain times while the simulated environment is always available. Virtual access to different testing environments makes virtual testing very cost efficient. Simulation is also safer for researchers; and allows them to safely refine their assumptions about the robot and their algorithms. Therefore, computer simulations allow a development team to be more effective and efficient.

B. The Need for Standard Performance Metrics

In some cases, there are errors in the robot models that result in physics inaccuracies with friction, gravity, mass, force, etc. The consequence of this is that the simulation results can be unreliable. In some cases, models exhibit behaviors that are not possible for the actual robots they represent. Researchers cannot accurately evaluate the performance of the robot with a faulty model. The challenge is therefore to develop a method to expose and resolve the inconsistencies between virtual models and real robotic systems.

C. The Proposed Solution

Developers can use standardized test methods [2] to ensure that the model they use behaves as close to the actual robot as possible. Using the test methods reveals unknown inconsistencies between simulation and reality, and researchers can then identify the problem with the physics of the model. One can resolve the issue systematically with an understanding of the simulation physics parameters. These standard test methods can also be used to verify existing model performance. With the virtual models validated, researchers can develop their software and confidently integrate their work onto physical systems.

II. BACKGROUND

A. USARSim Simulation Environment

USARSim is a high-fidelity simulation of urban search and rescue (USAR) robots and environments, and is intended as a research tool. It builds upon a commercially available game engine produced by Epic Games [3] known as the Unreal Engine 2.0. Today's games often achieve a high level of complexity and realism, and the game engines have become general purpose simulation engines that can be used to implement multiple games based on the same foundation.

[1] Certain commercial equipment, instruments, or materials (or suppliers, or software...) are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

They are extremely customizable, and therefore are excellent candidates to be used to develop robot simulators and perform scientific investigations [4].

While the internal structure of the Unreal Engine is proprietary, developers can purchase the engine code. For most uses this has been made unnecessary by the University of Southern California's Information Sciences Institute interface known as Gamebots [4] [5] that allows an external application to exchange bi-directional information with the engine. This interface was created for research in artificial intelligence and is an open source project [5]. The Unreal Engine implements a Virtual Machine, a concept very similar to the Java Virtual Machine, which allows for external code to be executed by the engine. The code must be written in the UnReal host language called UnrealScript, which is an object oriented language with syntax resembling C++ and JavaScript. The code may then be compiled into an intermediate platform-independent bytecode that is executed by the Unreal Engine. Through UnrealScript, a developer has full access to all environmental variables and full control of the actors in the world.

USARSim sits on top of Gamebots and provides a standardized interface to robot actuators and sensors. Extensive research of USARSim, in various applications, has shown the simulation to behave in a predictable manner with high correspondence to reality. This research is detailed in [6], [7], [8], and [9]. USARSim has experienced wide community acceptance with over 17,000 component downloads to date. In addition, it is the basis for the RoboCup Rescue Simulation League Virtual Robots Competition [10]. Additional information on USARSim and related software may be found in [11].

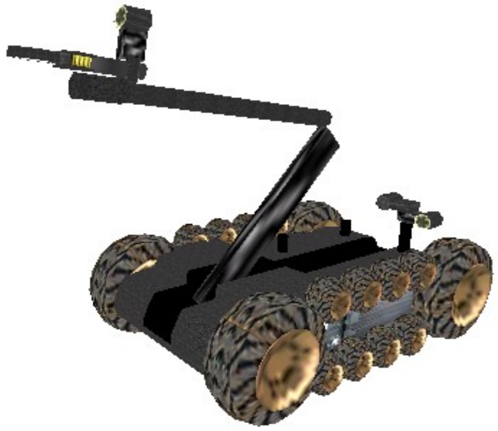


Fig. 1. USARSim Talon Model[2]

B. Karma Parameters

KActors are a class of objects that are controlled by the Karma Physics Engine. Karma is the game engine used by Unreal Tournament to control the vehicle physics, level physics, and rag doll physics [12]. Complicated systems, such as robot manipulators, can be created using Karma joints. Most objects in the simulation are static during game play, like static mesh actors[3]. KActors are dynamic and interactive, and each KActor has general Karma parameters, referred to as KParams, which define its own behavior in the simulation. The KParams that we use in this paper are KFriction, KangularDamping, and KCOMOffset. KFriction ranges between zero and one, where the KActor experiences no friction at a value of zero and total friction at a value of one [14]. KAngularDamping is the parameter that determines the magnitude of force to decrease the angular velocity of the KActor. KCOMOffset is a vector that defines the displacement of the center of mass from the center of the KActor. These are the Karma parameters that dictate most of the actions that we will change in the robot. More information on the Karma Physics Engine may be found in [12].

C. Talon Robots

Talon robots are robots produced by Foster-Miller, Inc. that are used for “explosive ordnance disposal (EOD), reconnaissance, communications, hazmat, security, defense rescue” [15]. We chose the Talon for this paper because NIST has access to the robot, allowing us to determine its capabilities through physical experimentation. It should be noted that the NIST robot is several years old and that newer, more capable Talon models exist.

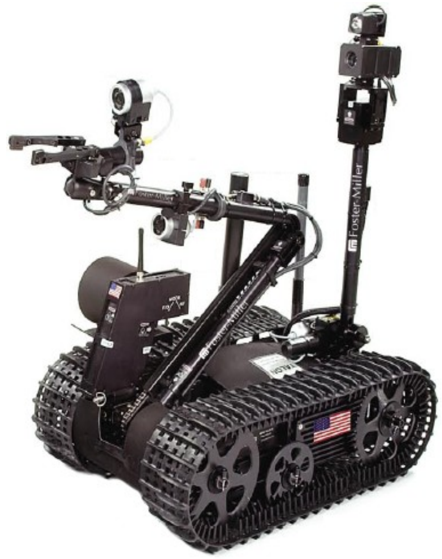


Fig. 2. Foster-Miller Talon Robot[4] [15]

III. NIST STANDARD TEST METHODS[5]

NIST engineers have developed standard test methods designed to analyze the performance of USAR robots in a repeatable and objective manner [1]. Each test was designed to test a specific attribute of a robot that is determinative of how successful it can be in a range of rescue situations. The

[2] Simulation results for particular payload shown in figure 1.

[3] Those unfamiliar with static mesh actors should read [13].

[4] This picture depicts a robot configuration different than that used by NIST in testing.

[5] Additional information about the NIST Reference Test Arenas for Autonomous Mobile Robots can be found in [16].

test methods are being developed in partnership with first responders, robot developers, and technical experts. The following test methods are a few of those created by NIST and others, several of which have been submitted to and approved by the Operational Equipment Subcommittee of the ASTM International E54.08 Homeland Security Committee [17].

A. Directed Perception

The directed perception test is designed to analyze the use of "robotic manipulators to perform a variety of tasks in complex environments" [17]. The test artifacts consist of cardboard boxes of uniform size with cutout holes. Each box has targets inside that different sensors can identify, such as lights and hazmat signs. Non-flat flooring also increases the difficulty of this test.

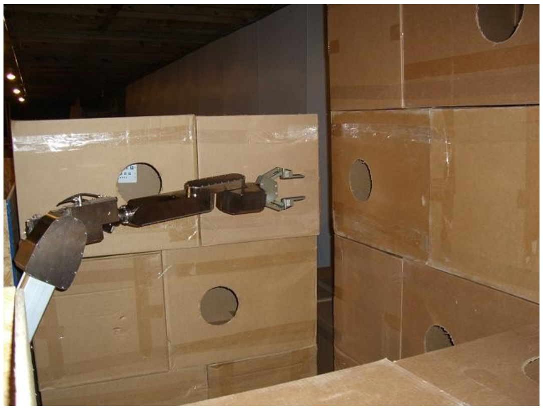


Fig. 3. Directed Perception Test at 2007 Metro Tech Event, NIST with teleMAX Bomb Disposal Robot

B. Grasping Dexterity

This test method analyzes the "requirement to retrieve objects, not necessarily configured for robot manipulators, within complex environments" [17]. The setup contains stacked shelves with items for the robot to pick up and place from one location to another on the shelving. The items are often blocks, simulated pipe bombs, or water bottles. The flooring is also often variable in terrain.

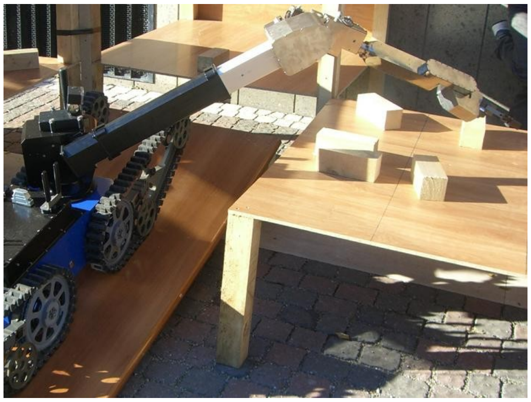


Fig. 4. Grasping Dexterity Test at 2006 RoboCamp Rome, Italy with teleMAX Bomb Disposal Robot

C. Stairs

Stairs test the ground mobility of a robot. The robots must be able to climb any variety of stairs, including stairs enclosed on the sides, with railings on the sides, with risers, or open stairs [17]. They can be constructed of different materials and at different slopes, presenting a difficult mobility task.

D. Step Field Pallet

The step field pallets are "repeatable surface topologies with different levels of 'aggressiveness'" [17]. A half step field pallet (also known as orange step fields) is classified as medium difficulty mobility, and a full step field pallet (also known as red step fields) is classified as high difficulty mobility. The computer generated random step field pallets are an abstracted test of the mobility of a robot. They are easily recreated and easily reconfigured. The step field pallets simulate uneven ground such as that seen in a rubble pile.

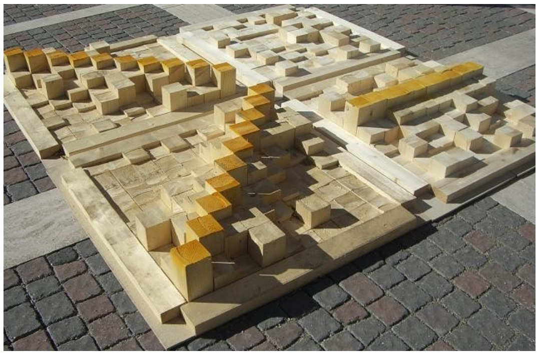


Fig. 5. Half Step Field Pallets at 2006 RoboCamp Rome, Italy

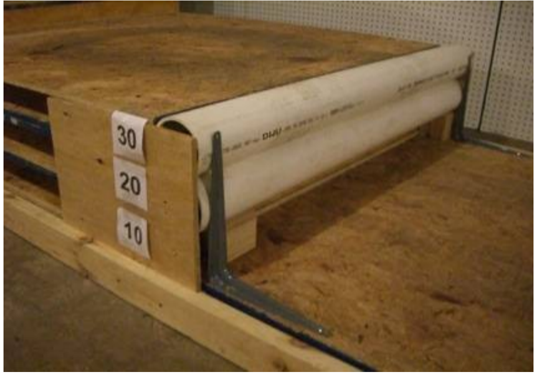


Fig. 6. NIST 30cm Step Test with Pipes

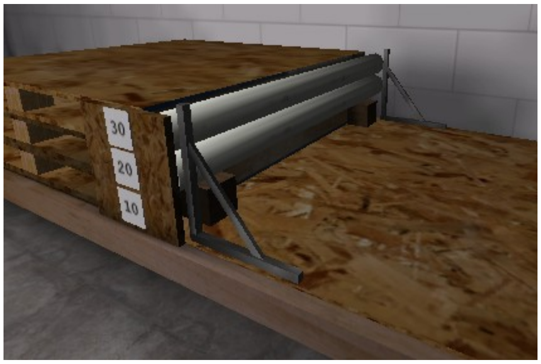


Fig. 7. NIST 20cm Virtual Step Test with Pipes

E. Step Test

The step test is designed to analyze the capability of a robot to climb increasingly higher plateaus. In some challenges, shelving brackets are used to hold polymer of vinyl chloride (PVC) piping at the edge, forcing the robot to not grip onto an edge for leverage. The free-spinning piping also simulates a slippery surface the robot may need to climb.

F. Mobility and Endurance (Zigzag and Figure 8)

These test methods are based on the step field pallet test. The formation of the step field pallets is designed to test the mobility and endurance of the robot. In this task, robots are to traverse a prescribed course of either a figure 8 shape or zigzag shape. Robots must be able to travel the length of the course quickly enough to avoid losing all battery life, and any field-repairs of the robot are timed. In the figure 8, multiple laps may be required.

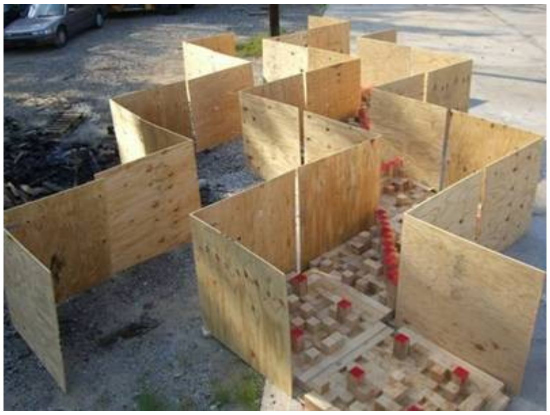


Fig. 8. Zigzag Endurance Test

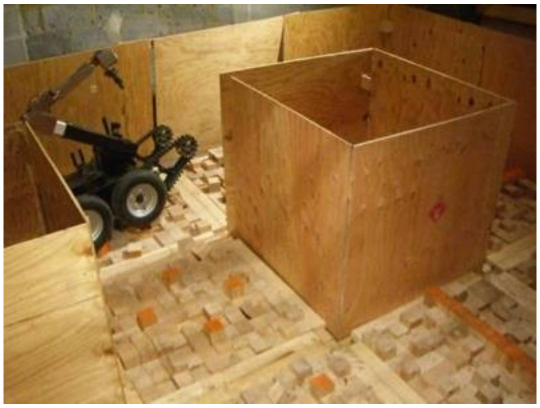


Fig. 9. Medium Difficulty Figure 8 Test
with teleMAX Bomb Disposal Robot

IV. VALIDATING TEST METHODS: THE STEP TEST

Prior to using test methods in simulation, we must first create the test methods and ensure *they* perform as expected. Researchers can determine the value of individual physics parameters with simple experiments and reasoned approximation. The model of the step test was created to the exact dimensions of the actual test. The important physics parameters in the real and simulated tests are the friction of the oriented strand board (OSB), the friction of the PVC piping, and the angular damping of the PVC piping.

A. Deriving the OSB Friction of the Step Test

A simple experiment was created to determine the actual frictional behavior of OSB. The test consisted of timing various sizes of OSB sliding on a larger OSB sheet at five different angles. Several trials were performed for each angle. At an angle of 9.9º, the approximately 35.6cm x 35.6cm (14" x 14") board had enough static friction to resist motion when at rest and enough kinetic friction to slow to a stop quickly when in motion. At 14.8º, the board took approximately 1.9s to slide down the approximately 122cm x 122cm (4' x 4') sheet. This behavior was replicated in a simulation through a heuristic derivation of the KFriction parameter, the final value of which was 0.56. Further testing revealed that this value at the remaining angles produced results with a strong correspondence to reality. These results are shown in Table 1.

TABLE 1

14"x14" PLYWOOD FRICTION TEST RESULTS

	Time for Sheet to Slide Down, seconds							
	Trial Number							
Ramp Angle	1	2	3	4	5	6	7	AVG
14.8°	1.91	/	/	/	/	/	/	1.91
15.2°	1.48	1.64	2.43	1.77	1.66	/	/	1.80
18.5°	1.06	1.27	1.19	1.11	1.06	1.45	1.61	1.25
27.8°	.81	.73	.72	.70	.72	.70	.72	.73

In the derivation of this parameter, several interesting results with the physics engine and its friction were recorded. The first observation was that the KMass of the object had no effect on the friction of the object. One would expect that this parameter was the coefficient of kinetic friction, $\mu_{k,}$ and that it would follow the classic relationship,

$$F=\mu_k N, \tag{1}$$

where N is the normal force and F is the force of the drag, but this was not the case. The second observation was that static meshes, with added KParams or without, do not affect the actions of a KActor. KActors seem to only be affected by other KActors. The pallets used in the step test were changed from static meshes to KActors, to allow the test to affect the vehicles. Because KActors are movable during simulation, the translational motion of them must be controlled. The motion of the pallets was limited by ball and socket joints, a KBSJoint Karma constraint in UnrealEd. These constraints prevent the pallets from sliding out from under the robot during the test.

B. PVC Piping Physics Parameters

Analysis of film from previous USAR events with the step test showed that there was little to no slip between robot tracks and the PVC piping. KFriction provides full friction, i.e. no slip, when set to a value of 1.0. A value of 0.9 for KFriction will allow tracks to mostly grip the pipe with a small amount of slip. Finally, the value of the angular damping parameter needs to be determined because the pipe experiences friction from the shelving bracket in reality. The value of 1.0 was chosen to prevent the pipe from spinning endlessly and to allow the robot tracks to easily spin the pipe.

V. IDENTIFYING MODEL INCONSISTENCIES

Based on behavior analysis of the model, the step test in reality and virtual simulation are now consistent. Testing with these methods may uncover differences between the actual robot and the virtual model. To do this, we simply analyzed data captured on the actual robot as it attempted different tasks on the test. It is important to note that the difficulty of the tests must be increased, for example raising the height of the step, until the physical robot is unable to accomplish the test. This provides an upper bound for what the simulation should be capable of performing. After analysis of this information, we tested the virtual model to determine whether the simulation behavior was accurate. Rigorous comparison shows how the virtual model needs to be altered.

This was the process for analyzing the Talon robot and model. The first experiment on the step test was driving the robot in a direct forward approach, the second was at an angled forward approach (figure 10), the third was a reverse approach (figure 11), and the final experiment was at an angled reverse approach[6]. The same procedure was then repeated for the model in simulation. The results of these tests are shown in Table 2, where a "yes" is climbing the step and a "no" is not doing so. Other observations were recorded, such as issues the implemetation of the approximation of the track behavior.

TABLE 2

RESULTS OF 20CM STEP TEST WITH PIPE

	Robot	Model	Correlates
Direct Forward	No	No	☑
Angled Forward	No	Yes	☒
Direct Reverse	No	Yes	☒
Angled Reverse	No	Yes	☒

A. Track Implementation

In the current version of the Unreal Engine, version 2, tracks on vehicles must be approximated. These tracks are approximated in one of two ways. The first has a static tread attached to the robot, and the robot uses the gears (that normally propel the track in reality) to propel the vehicle by directly interacting with the world as wheels. A vehicle model that does this is the teleMAX robot, developed by telerob [18]. Another method used to estimate the behavior of the track is to have many wheels of different sizes approximate the shape of the track.

The second method was used for the Talon, where the Talon has large front and rear wheels and little wheels in between. The small wheels can move translationally to simulate the flexing of the track. Currently, these wheels are rigid and oppose transltational motion. Testing has shown that the wheels on a side, which are supposed to behave as a single track, can spin at different speeds or in different directions, which is not possible for a track. The individual gears all contribute to the motion of the track, which is at a uniform speed at all points on the track. The implementation of the tracks needs correction to make the wheel motion uniform.

B. Model Climbs 20cm Height with Piping at Angle

The simulation model was able to climb the step test of 20cm with two PVC pipes. To do this, a controller had to drive the virtual robot such that it approached the piping at an extreme angle of incidence. The robot would begin to climb up with one track and turn such that the second track would also be on the pipe. Then moving forward it was able to completely pass the step. In testing with the actual robot, the robot would rotate into the direct forward approach when attempting the test at angles. With the robot directly approaching the piping, it spins its tracks and is unable to get on top of the pipes or step. The actual robot was unable to climb the same test height that the virtual model could.

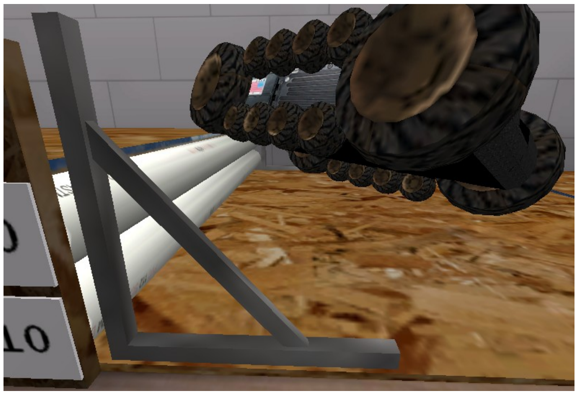


Fig. 10. Talon Model Angled Forward Approach

C. Model Climbs 20cm Height with Piping in Reverse

When the controller drove the Talon in reverse to the step, it was able to rise up the piping and nearly climb the step. The model in USARSim was able to climb the stairs in reverse with ease.

[6] All of the real and simulated tests were performed with the manipulator arm folded on top of the robot to keep a constant center of gravity. This position is the start pose of the robot arm and can be seen in figure 11.

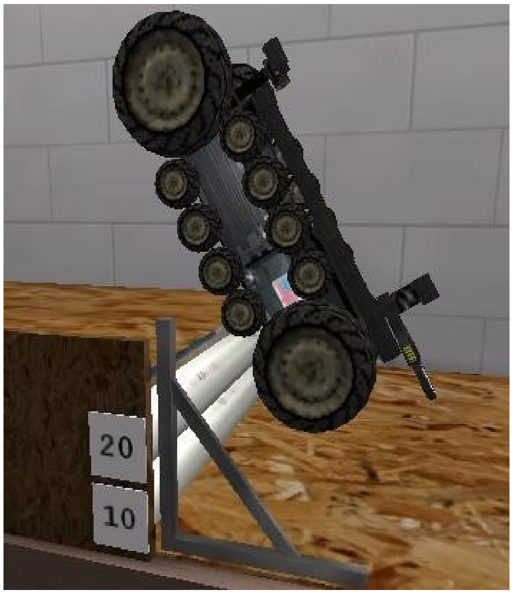


Fig. 11. Talon Model Climbing of Pipe in Reverse

VI. CORRECTING THE ROBOT MODEL

Once a difference between the model and the robot is identified, one can then adjust and fine tune the behavior of the robot model with an understanding of the Karma parameters. This is accomplished by changing the physics and retesting the robot. This process is repeated until a model can be verified through its performance on the test methods.

A. Track Implementation

For the track to behave properly as a group of many wheels, the wheels must all have the same angular velocity. As it is currently, the tires are all spawned by the KDTrack.uc class. Each track spawned is issued commands by USARSim. These commands are then directed to each wheel. When the entire track is issued a command to drive forward, each wheel attempts to do just that. Because the wheels interact differently with the simulation environment, the actual response of the wheel is then calculated on an individual basis. The individual wheels of the track can respond differently to a single command. Another issue was the small wheels needing realistic linear damping to simulate the flex of the track. The KLinearDamping parameter value must be lowered in TalonTrack.uc to produce accurate results.

B. Model Climbs 20cm Height with Piping at Angle

The tracks of the Talon model are able to grip onto the pipe enough to pull the robot on top of the pipe. This is an unrealistic part of the model that produces the uncharacteristic behavior. The classes that control the behavior of the track are the TalonTrackTire.uc and TalonTire.uc. These classes extend KTire, the Unreal Tournament class that characterizes the tires of the Unreal vehicles. Because of this, they inherit control of the friction, slip, and normal properties of the KTire class. In the Talon track classes, the tire properties must be changed to correct the model. The lateral friction on the model is too high if the robot rotates sideways when approaching at high angles of incidence, and the roll friction must be reduced for the track to not be able to grip the pipe. Lastly, the motor torque of the robot must be decreased to lessen its climbing ability. Other parameters such as tire softness, tire adhesion, and the slip rate can affect the performance of the track. These changes have proven to successfully correct the behavior of the robot in testing.

C. Model Climbs 20cm Height with Piping in Reverse

Some of the above parameters that changed with the adjustments on the track will help lessen the problem of the simulated robot climbing the step test in reverse. The class that defines the behavior of the Talon is Talon.uc. The actual robot not being able to climb forward but able to in reverse indicates a center of mass that is not at the center of the robot. The Karma parameters of an actor are defined within the KParams of that object. The property that will change the center of mass is the KCOMOffset, which has not been set in the current model. The center of mass is defaulted to the origin of the robot. By measuring the actual robot to find its center of mass, the KCOMOffset can be accurately changed to be accurate. Should the robot be unavailable for measurement, it is reasonable to estimate the center of gravity from the location of the heavy battery packs in the front of the vehicle. The offset would be near halfway toward the front of the vehicle. This assumption proved accurate in final testing of the behavior of the modified robot model.

VII. MODEL VERIFICATION

The test methods are not only used to alert researchers of physics problems, but are also used to show that a model is accurate. With several of the same test methods, testing revealed that the robot model behaved as the actual robot.

A. Directed Perception

The arm and manipulator control of the Talon are accurately replicated for the Talon model in USARSim, which is illustrated by the directed perception test. The Talon manipulator uses joint level control to move each link individually. The performance data captured in simulation shows a close correspondence to data captured on the actual test method. The range of motion for each joint has been set to realistic values that may be inaccurate to the actual range of motion for the Talon manipulator. This can be corrected after a few tests with the actual robot manipulator.

B. Grasping Dexterity

The manipulator of the Talon was shown to be accurately modeled in the grasping dexterity test. This test also analyzed the gripper of the Talon arm. The robot has a gripper with two fingers, and the model has these at the correct dimensions. The control of the manipulator and gripper have been correctly modeled.

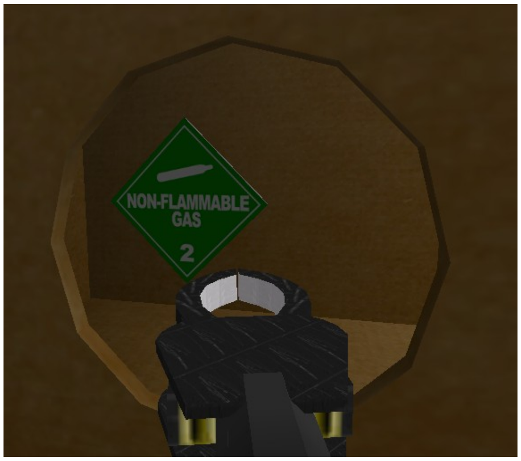


Fig. 12. Talon Model Successful Directed Perception Test

C. Stairs

At a Department of Homeland Security (DHS) workshop held in Las Vegas in 2005, the Talon robot was recorded as it climbed an open stairway with railings. The robot was able to use rocks at the base of the stairs to get on top of the first stair. Once on the stair, it was able to climb the remaining stairs with relative ease. In simulation, the robot had difficulty getting onto the first stair without a small obstacle. With that obstacle in place, the model was able to complete the test with ease[7]. The stairs used in the simulated test have a slope of exactly 40°. This is a slope close in value to that of the stairs on which the Talon was tested, which are estimated at 41°. Both tests were also performed on open stairs.

D. Step Field Pallet

The step field pallets were also useful in verifying the robot model. The robot can perform well on half step field pallets (medium mobility difficulty). The full step field pallets (difficult mobility) however proved challenging for the robot. The actual robot is able to eventually complete the difficult mobility test by reversing and reattempting at different angles, which is also the case for the model robot in USARSim. The model completed the medium mobility test with little trouble, and completed the difficult mobility test with some difficulty.

E. Mobility and Endurance (Zigzag and Figure 8)

Being based on the step field pallets, figure 8s and zigzags highlight much of the same abilities of the robot. Because the medium difficulty mobility is not challenging for the Talon, this test analyzes the endurance of the robot. The difficult tests focus on the mobility of the robot. The battery life of the Talon robot is near four hours at typical operational speed [15]. The battery life of any robot in USARSim is a configurable variable, which defaults to 20 minutes. The battery life is not calculated based on the use of the electric devices or energy consumption of the motors; however, this model simplification is acceptable because the difficulty of implementation outweighs the minimal benefits of battery accuracy. In addition to this, robots in USARSim cannot be damaged yet. Robot damages is being researched, and will be tested with these endurance tests once implemented.

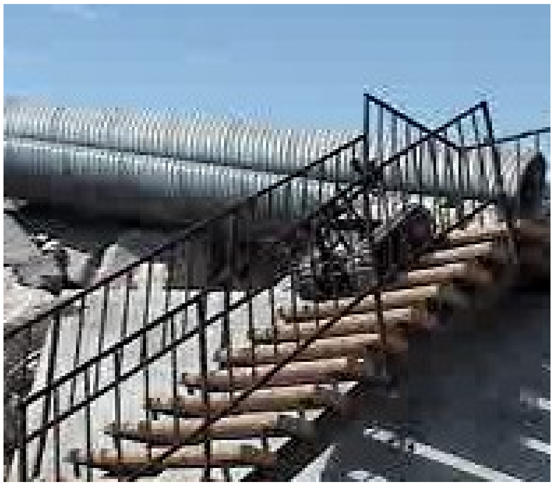


Fig. 13. Talon Robot Pass Stair Test at 2005 DHS Workshop, Las Vegas

VIII. CONCLUSION

This testing has revealed the test methods to be an excellent solution to the problem of determining and increasing simulation accuracy. The simplicity of the tests makes model fabrication and physical construction easier. The test methods created for the ASTM standard test specific characteristics of the robot, making them easy to use for modifying robot models. In using the test methods, a researcher is able to identify a specific problem, and can then improve the model accordingly. Developers can also use these tests to validate existing models, and show that the behavior is accurate to reality.

IX. FURTHER RESEARCH

The changes discussed on the Talon model will be implemented, including forcing the tires of the track to spin at the same angular velocity. NIST is currently investigating possible solutions to the issue. The release of the new Unreal Engine 3 may provide an answer.

Another area of future experimentation is the gripper behavior. Testing will be performed to determine the accuracy of the gripper strength. The arm must also be tested to determine the amount of weight it can lift. A simple test of lifting increasingly heavier weights with the actual Talon in a repeated manner will illustrate the behavior the model should mimic. Repeating the same test in simulation will allow for precise retuning of the model physics. In addition, other commercial platforms will be subjected to similar tests and have their models validated.

[7] Because the stairs of the test in Las Vegas are unavailable for friction experimentation, analysis of captured performance data was used to validate the simulated test method. Creating the test method as a static mesh produced results with strong correlation to the robot behavior observed at the DHS Workshop.

As different waves of first responder requirements are implemented in the robots and robot models, more test methods will need to be developed. The individual capabilities of the robot must be tested to ensure that they were implemented correctly.

ACKNOWLEDGEMENTS

The authors would like to thank the developers of USARSim, Benjamin Balaguer for assisting in Talon model modifications, and Tony Downs for conducting and assisting in testing with the Talon robot.

REFERENCES

[1] Messina, E., "Performance Standards for Urban Search and Rescue Robots," ASTM International Standardization News, August 2006.
[2] A. Jacoff, E. Messina, J. Evans, "Performance Evaluation of Autonomous Mobile Robots", Industrial Robot: An International Journal, Volume 29, Number 3, 2002, pp. 259-267.
[3] Epic games, "Unreal engine," 20 Jul 2007. www.epicgames.com, 2005.
[4] M. Lewis, J. Jacobson, "Game engines in scientific research," Communications of the ACM, Volume 45, Number 1, pp. 27–31, 2002.
[5] Gamebots, 20 Jul 2007. http://gamebots.sourceforge.net/.
[6] S. Carpin and M. Lewis and J. Wang and S. Balakirsky and C. Scrapper, "USARSim: a Robot Simulator for Research and Education", Proceedings of the IEEE 2007 International Conference on Robotics and Automation, October 2007.
[7] S. Carpin and J. Wang and M. Lewis and A. Birk and A. Jacoff, "High Fidelity Tools for Rescue Robotics: Results and Perspectives", Robocup 2005: Robot Soccer World Cup X, Springer, LNAI, Volume 4020, 2005, pp. 301-311.
[8] J. Wang and M. Lewis and S. Hughes and M. Koes and S. Carpin, "Validating USARSim for Use in HRI Research", Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting(HFES05), 2005, pp. 457-461.
[9] M. Zaratti and M. Fratarcangeli and L.Iocchi, "A 3D Simulator of Multiple Legged Robots based on USARSim", Robocup 2006: Robot Soccer World Cup X, Springer, LNAI, 2006.
[10] M. Balakirsky, C. Scrapper, S. Carpin, and M. Lewis, "USARSim: A RoboCup Virtual Urban search and Rescue Competition", Proceedings of SPIE, 2007.
[11] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper, "USARSim: A Robot Simulator for Research and Education", Proceedings of the IEEE International Conference on Robotics and Automation, 2007.
[12] "Unreal Developer Network Karma Reference." Unreal Developer Network. 19 Jul 2007 http://udn.epicgames.com/Two/KarmaReference.html.
[13] "Static Mesh" Wikipedia, the Free Encyclopedia, July 31, 2007, http://en.wikipedia.org/wiki/Static_Mesh.
[14] Busby, Jason, and Zak Parrish. Mastering Unreal Technology: The Art of Level Design. Indianapolis, Indiana: Sams Publishing, 2005.
[15] "Products & Talon Military Robots, EOD, SWORDS, and Hazmat Robots." Foster-Miller, Inc - QineticQ North America. July 13, 2007 http://www.foster-miller.com/lemming.htm.
[16] Jacoff, A., Messina, E., Weiss, B.A., Tadokoro, S., Nakagawa, Y., "Test Arenas and Performance Metrics for Urban Search and Rescue Robots," Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NE, October 27-31, 2003.
[17] Jacoff, A. and Messina E., "Urban Search and Rescue Robot Performance Standards: Progress Update," Proceedings of the 2007 SPIE Defense and Security Symposium Unmanned Systems Technology IX, Orlando, FL, April 2007.
[18] "teleMAX" telerob, July 23, 2007, http://www.telerob.com.

Design and Validation of a Whegs Robot in USARSim

B.K. Taylor
Case Western Reserve Univ.
Cleveland, OH, USA
brian.k.taylor@case.edu

S. Balakirsky, E. Messina
NIST
Gaithersburg, MD, USA
stephen.balakirsky@nist.gov,
elena.messina@nist.gov

R.D. Quinn
Case Western Reserve Univ.
Cleveland, OH, USA
roger.quinn@case.edu

Abstract— Simulation of robots and other vehicles in a virtual domain has multiple benefits. End users can employ the simulation as a training tool to increase their familiarity and skill with the vehicle without risking damage to the robot, potential bystanders, or the surrounding environment. Simulation allows researchers and developers to benchmark the robot's performance in a range of scenarios without needing to physically have the robot and or necessary environment(s) present. Beyond benchmarking current designs, researchers and developers can use the information gathered in the simulation to guide and generate new design concepts. USARSim (Urban Search and Rescue Simulation) is a high fidelity simulation tool that is being used to accomplish these goals within the realm of search and rescue. One particular family of robots that can benefit from simulation in the USARSim environment is the Whegs™[*] series of robots developed in the Biologically Inspired Robotics Laboratory at Case Western Reserve University. Whegs robots are highly mobile ground vehicles that use abstracted biological principles to achieve a robust level of terrestrial locomotion. This paper describes a Whegs robot model that was designed and added to USARSim's current array of robots. The model was configured to exhibit the same kind of behavioral characteristics found in the real Whegs vehicles. Once these traits were implemented, a preliminary validation study was performed to ensure that the robot interacted with its environment in the same way that the real-life robot would.

Keywords: *USARSim, Biologically Inspired Robotics, Whegs, Urban Search and Rescue, Simulation*

I. INTRODUCTION

A. Background on USARSim

Urban Search and Rescue Simulation (USARSim) is a high fidelity simulation tool that can be used to simulate robots in various environments [11]. USARSim is built on top of Epic Games' Unreal Tournament 2004[*] (UT2004) physics engine known as Unreal Engine 2.0. The Karma Physics Engine[*] [9] is utilized to simulate physics within the game. Unreal Script, the object oriented programming language for UT2004, is used to give robots functionality and to define how the robot will interact with its environment. Unreal Editor (UnrealEd) is used to create virtual worlds, or maps. It is also used to create 3D solid models (static meshes) that can be used to either construct a robot, or to construct obstacles and objects that are placed within a particular map.

The idea behind USARSim is as follows. A virtual robot is built by creating static meshes to represent its individual parts. The parts are connected to each other through a configuration file that specifies where and how parts are connected to each other (motors, hinges, ball-and-socket joints, etc). In addition to the robot, a map is created with obstacles that must be overcome, and objects and/or victims that need to be found. The physics engine handles the dynamics of how the robot should interact with the map that it is placed in. By adjusting parameters known as Karma Parameters [9,10], the performance of the robot in simulation can be changed. For example, changing the inertia tensor of a robot will affect its ability to rotate about particular body axes within a given world. For robots and maps, end users can select from the options available in a current release of USARSim [11], or design their own. Controller software is used to perform a range of tasks such as issuing simple drive commands, implementing autonomous features into the vehicle, and running multiple vehicles in a given environment [11,12]. This kind of setup allows an individual to build and simulate robots relatively quickly and inexpensively from both computational and monetary standpoints. USARSim currently has applications in end-user training for Urban Search and Rescue robots, and in the RoboCup Simulation League [2,3].

A disadvantage of USARSim is that the Karma Physics Engine is proprietary. This means that the exact mechanics behind how the engine uses the Karma Parameters cannot be obtained. Testing has been done to gain a better, more quantitative understanding of how the Karma Parameters affect the simulation, and how the parameters map to real-world quantities. For example, conversion factors between the simulation's length scales (Unreal Units and Karma Units) and real length scales (meters) have successfully been established and implemented in more recent releases of the software. However, there are still parameters

[*] Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

where this kind of understanding has not been reached. Ultimately, this means that iterative testing and comparisons must be done on a given real and virtual robot to determine the set of Karma Parameters that yields the most realistic performance of the virtual vehicle.

B. Background on Whegs

Whegs robots are highly mobile unmanned ground vehicles that were developed in the Biologically Inspired Robotics Laboratory at Case Western Reserve University. Their locomotion is based on abstracted biological principles observed in cockroach locomotion [1]. Unlike RHex which is a biologically inspired robot that predates Whegs [7], Whegs robots employ an appendage called a wheel-leg, which is made up of a hub with spokes equally spaced about the hub's central axis (Fig. 1).

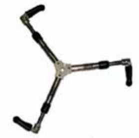


Fig. 1. A three spoke wheel-leg appendage

Most wheel-legs have three spokes. Rotating the wheel-legs about their central axes at a constant speed allows a given Whegs robot to move in the same way that a wheeled vehicle would be driven. In addition, the spokes allow the robot to obtain discontinuous footholds on irregular terrain, similar to legs [7]. Furthermore, the spokes also allow the wheel-leg to reach footholds that are taller than the wheel-leg radius. These wheel-leg features allow Whegs robots to be propelled in a similar manner to wheeled vehicles. They also enable Whegs robots to climb over and negotiate terrain that may be impassable to wheeled vehicles (Fig. 2).

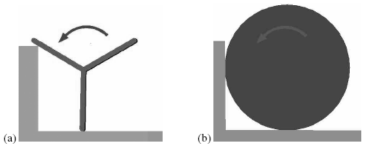


Fig. 2. (a) A wheel leg is able to obtain footholds on obstacles that are taller than the wheel-leg radius. (b) A wheel is unable to reach footholds of equal height

Cockroaches have six legs and typically walk in a tripod gait, meaning that the front and rear legs on one side move in phase with the middle leg on the opposite side. Contralateral pairs of legs move out of phase with each other (e.g. when the front left leg is in swing, the front right leg is in stance) [5]. When the animal comes to a large barrier, it moves its contralateral legs into phase to aid in surmounting the obstacle [6]. Similarly, Whegs robots employ six wheel-legs with contralateral pairs being placed out of phase such that the vehicle walks in a nominal tripod gait. Each axle features a compliant mechanism that allows the robot to passively bring its wheel-leg pairs into phase. This feature aids the robot in surmounting obstacles, and allows it to passively adapt its gait to changing and irregular terrain (Fig. 3).

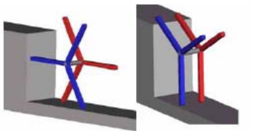


Fig. 3 Compliant mechanisms in the axles allow wheel-leg spokes to come into phase which aids in climbing obstacles

Also, cockroaches have a body flexion joint. They use the joint pitch the front of their bodies down to avoid high centering and to allow their front legs to reach the substrate [6]. More recent Whegs robots have been outfitted with body flexion joints or similar reasons. In addition, the joint also allows the vehicle to pitch its body upward to get a foothold on an obstacle during a climb (Fig. 4) [8].

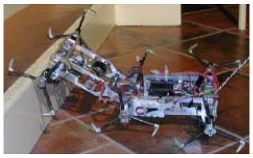


Fig. 4. Whegs robot with a body flexion joint surmounting a larger step than the robot's body height

In addition to Whegs robots, a series called Mini-Whegs™* has also been developed (Fig. 5).

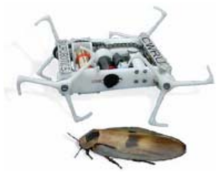


Fig. 5. Mini-Whegs and its relative size as compared to a cockroach

These are intended to be smaller, more compact versions of Whegs robots. They are on the order of 0.09 m long (9 cm), and have a top speed of about 10 bodylengths/second (0.9 m/s). Because of their small size, Mini-Whegs robots only possess four wheel-legs instead of six [4]. The four wheel-legs move in a diagonal gait. While some work has been done with implementing compliance into the axles of these robots, for simplicity, Mini-Whegs typically lack both torsional compliance and body flexion joints [4].

While different types of Whegs robots have been constructed and tested, there is formalized method for developers to gauge a robot's performance limits or test the viability of design ideas before construction begins. Also, the only current way to learn how to operate a Whegs robot is to drive a real robot. A Whegs simulation would allow robot designers to test their ideas before construction begins, allowing them to make design changes that will improve performance. The simulation can also be used by developers to test robots in environments that are not readily available, or potentially damaging. This would allow designers to gauge a given robot's performance limits. For robots in development, performance enhancing changes could then be implemented. In addition to design work, a Whegs robot simulation would allow end users to become skilled in operating Whegs vehicles in numerous environment(s) without having the robot or environment(s) physically present. This can reduce the risk of damaging the robot. If a simulated robot is incapacitated, the simulation can be restarted rather than having to repair or rebuild the vehicle.

In this paper, a virtual Whegs robot was created and given the same behavioral characteristics as a real robot. The virtual robot's performance was then benchmarked against the real robot. Section II describes the approach and methods used to impart functionality to the virtual robot and benchmark its performance. Section III describes the results obtained during testing. It also describes some of the problems that were encountered during testing and how these issues were resolved. The final section summarizes the work presented in this study and discusses future work.

II. Methods

To perform this study, a generic Whegs robot model was first created in USARSim by adding a "Whegs" class. This class and its base classes were given functionality to enable the virtual robot with the same behavioral characteristics that are found in a real Whegs vehicle. After the virtual robot had the necessary behaviors, it was run through several tests to gain an understanding of how particular Karma Parameters affected its performance. Once the effects of these parameters were known, the virtual and real robots were placed in test scenarios with the same conditions. The results of these tests were compared and used to make changes to the virtual robot's Karma Parameters to improve its performance.

For the purposes of this study, the virtual vehicle was modeled after a Mini-Whegs robot with torsional compliance. This was done in an effort to lay the ground work for creating any given Whegs vehicle while still maintaining a degree of simplicity during modeling and testing. As stated above, Mini-Whegs robots typically lack a body flexion joint and only use four wheel-legs. These features make Mini-Whegs robots easier to simulate because there are fewer features to control and less wheel-legs to monitor during testing. Torsional compliance, while not present on all Mini-Whegs vehicles, was not a feature present in USARSim. Because this feature is found on many of the Whegs vehicles, it was felt that successfully modeling and implementing it would aid in laying the fundamental groundwork necessary for building more specific and accurate Mini-Whegs and full size Whegs models.

A. Developing a Whegs™ Robot Model

Modeling a Whegs vehicle can be broken down into two main steps: creating the appropriate static meshes, and writing and modifying classes to give the virtual vehicle the same types of behavioral characteristics as the real vehicle. The static meshes were created using UnrealEd. For the purposes of simplicity, the chassis was modeled as a rectangular block. The wheel-legs were modeled as cylinders (the hub of the wheel-leg) with three rectangular blocks (the spokes of the wheel-leg) attached to them and placed 120° apart (Fig. 6).

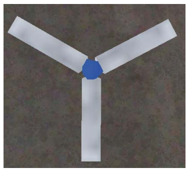


Fig. 6. Wheel-Leg appendage created in UnrealEd

Two separate wheel-leg meshes were made. One resembled a "Y" shape (Fig. 6) while the other was an inverted "Y". The two meshes were used as contralateral leg pairs to achieve proper wheel-leg phasing.

Incorporating the appropriate behavioral characteristics into the robot involved adding new functionality into USARSim. As mentioned above, there were no native USARSim features that allowed for passive torsional compliance. To solve this problem, a new class called "KDSpringy" was created. This class tells USARSim to make a hinge joint (KHinge) whose hinge type is set to a spring (KHingeType=Springy). Physically, this is like connecting two objects together with a torsional spring that is able to have stiffness and damping about a particular axis. The spring attempts to maintain an input angle (KDesiredAngle) between two objects placed in an Unreal map (Actors). In USARSim, this corresponds to maintaining a desired angle between the current part and its parent. Once the KDSpringy class was created and the appropriate base classes were modified, it was implemented in the following way.

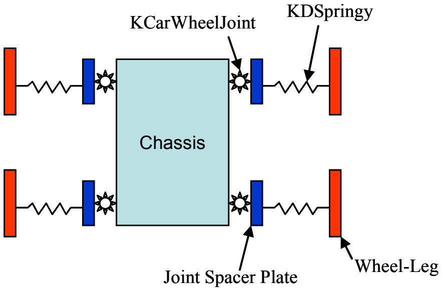


Fig. 7. Illustration of how passive torsional compliance is implemented

A static mesh that is used as a spacer plate was created. As can be seen in Fig. 7, the chassis was connected to the spacer plate via a KCarWheelJoint. A KCarWheelJoint is a joint that has a spin axis that is driven by a motor, and a steering axis that is driven by a controlled motor that attempts to achieve a specified orientation, similar to a servo. The spacer plate was connected to the wheel-leg via KDSpringy. This setup effectively made the spacer plate the actual "wheel" that drove the vehicle. However, because the wheel-leg's parent is the spacer plate, the two parts rotate together. Differences in their rotational speeds come from the reaction torques and forces that are experienced by the wheel-leg from the terrain, and from the parameter used to set the stiffness of KDSpringy. A large stiffness allows the spring to withstand large reaction torques before deflecting, thus allowing the wheel-leg and spacer plate to move at more closely matching speeds (this corresponds to the wheel-leg functioning like a normal wheel). A lower stiffness means that the spring is easier to displace and must be deflected to the point where it is able to exert a large enough torque to spin the wheel-leg. This enables the spacer plate to wind up the spring and build torque when a particular wheel-leg is unable to move, which allows the contralateral wheel-leg to come into phase with it. This is precisely how Whegs robots behave in reality when surmounting obstacles.

A disadvantage to this method is that UT2004 appears to only use tire properties for a tire (a KTire in UT2004) that is connected to a KCarWheelJoint. KTire properties allow the user to control the following properties of the tire: Rolling Friction, Lateral Friction, Rolling Slip, Lateral Slip, Minimum Slip, Slip Rate, Tire Softness, Tire Restitution, and Tire Adhesion. Even though the wheel-legs are defined as KTires, since they are connected via KDSpringy, it appears that they only have what are known as KActor properties. KActor properties allow the user to control the following parameters: KFriction and KRestitution. As can be seen, a KTire is the ideal case because more control is allowed over how the tire will interact with the environment. This problem can be rectified by altering the static mesh so that individual wheel-leg spokes are added to the hub as tires. However, this solution requires a larger number of parts and more class functionality to make the robot function properly. Also, this approach gave adverse preliminarily results, which are discussed in the next section. Because real Mini-Whegs robots do not use formal tires, the preliminary validation was performed with the wheel-legs as KActors. It was felt that this approach would provide a good first approximation of the appropriate set of Karma Parameters that would yield realistic virtual performance while narrowing the search space at the same time.

B) Virtual Test Maps and Validation Testing

After the virtual robot was developed, two maps were created to test the vehicle's performance. One of these maps was a large empty room to test the vehicle in walking and running while minimizing its chances of hitting a wall. The other map included basic obstacles such as: ramps, standard 2x4 boards (3.81 cm by 8.89 cm actual cross sectional dimensions) and textbooks for climbing, a straightaway for walking/running testing, and stairs and large drops for falling and impact testing. These worlds were used to compare the virtual robot's performance to that of the real vehicle. In this study, attention was focused on walking/running and basic climbing over textbooks. The following metrics were used to evaluate the virtual robot's performance:

- Top speed of about 0.9 m/s without significant end-over-end rotation (~25 rad/s wheel-leg drive speed)
- End-over-end rotation when attempting to climb up a wall at higher wheel-leg drive speeds
- The ability to surmount obstacles (textbooks in this study) that are 0.04 m tall in head on and oblique angle (30°) approaches (Fig. 8)

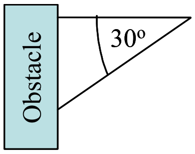


Fig. 8. Top view of head on and oblique approaches

Fig. 9 provides an illustration of how robots are currently validated

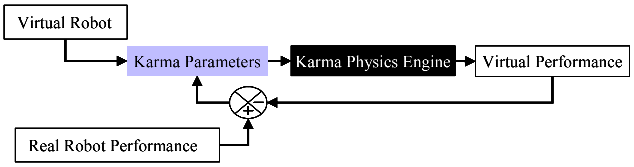


Fig. 9. Illustration of how robots are currently validated

As stated above, the Karma Physics Engine is proprietary, which means that the exact way in which the engine uses the Karma Parameters to affect the simulation cannot be directly obtained. In addition, the engine uses its own unit system (e.g. lengths are in Unreal Units or Karma Units depending on the context). The mapping between the Unreal Unit System and real world quantities is known for some parameters (e.g. length and time). However, conversions for other parameters (e.g. force and torque) are still under investigation. Due to the proprietary nature of the engine, to perform a validation, a robot is run through the engine with an initial set of Karma Parameters, as illustrated in Fig. 9. This yields the robot's virtual performance, which is then compared with real robot performance through the use of video data and any other relevant performance metrics for the robot in question. The information learned from the comparison is used to modify the Karma Parameters. After parameter modification, the virtual robot is run through the engine again for comparison with the real robot. This cycle is repeated until the virtual performance meets a desired level. For actual validation testing, this method was combined with the following procedure:

1) Baseline Run: First, the virtual robot was run through a range of drive speeds with an initial set of Karma Parameters. The goal of this run was to obtain performance data for an initial set of parameters for the purposes of comparison. In the open room, vehicles were run through the following wheel leg drive speeds: {0, 2, 10, 15, 20, 25, 30} rad/s.

2) Individual Karma Parameter Variation: After the baseline performance test, individual Karma Parameters were varied through a range of values while leaving all other parameters at their initial settings. For each value, the virtual robot was run through the same set of drive speeds used in the baseline run. The purpose of these runs was to obtain data that illustrated how each parameter affected the vehicle's performance.

3) Physical Reasoning: At this stage, physical reasoning was used to determine what conditions were required for the virtual robot to behave in a particular way in order to explain its performance and the effects of individual Karma Parameters.

4) Karma Parameter Search: At this point, with an understanding of the effects of different Karma Parameters, the method illustrated in Fig. 9 was used to improve the virtual robot's performance.

The virtual robot was compared to physical observation of a real Mini-Whegs robot. All simulation trials were recorded using FRAPS* [13] video capturing software. In addition, the vehicle's instantaneous velocity, position in the world, orientation with respect to the world coordinate frame, time, and speed change commands were logged to various files. The logged parameters were used to plot the vehicle's speed and velocity components against time. Velocities were reported in both world (fixed) and vehicle (moving) coordinates to help quantify the robot's behaviors. Steering was not used in any of the tests.

III. Results

A. Dimensions Used in the Simulation

Initial testing (phases 1-3 of the above described procedure), was done with a slightly larger vehicle. Phase 4 was performed with a smaller vehicle (Table 1).

		Initial Dimension	Final Dimension
Body	Length (m)	0.1143	0.09
	Width (m)	0.09144	0.068
	Height (m)	0.01905	0.02
Wheel-Legs	Diameter (m)	0.096	0.0762

Table 1. Initial and final dimensions used in the simulation

The latter dimensions were chosen because they more accurately reflect the size and performance basis of current Mini-Whegs robots. For example, the 10 bodylength/second speed listed above corresponds to a 0.09 m bodylength, so for this performance metric, the smaller vehicle size is more appropriate. The study could have been performed with the larger size vehicle since a real vehicle could be created that has larger dimensions. The dimensional change is only used here for convenience in comparing virtual and real performance.

B. Functionality for Maintaining Proper Wheel-Leg Phasing

During testing, it became apparent that new functionality would need to be added to the "Whegs" class to ensure that the virtual robot maintained proper wheel-leg phasing. Initially, when the vehicle spawned into a world, it would spawn properly with its wheel-legs out of phase, but then "fall" due to its mass such that the wheel-legs were in phase (Fig. 10).

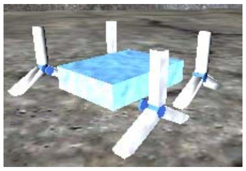


Fig. 10. Wheel-Legs are unable to maintain proper phasing without extra class functionality

Occasionally, all of the wheel-legs would stay out of phase upon spawning such that the robot could be tested. However, if the robot was not stopped with the correct orientation and speed, the wheel-legs would "fall" out of phase. This behavior appeared to be independent of the torsional stiffness that was provided, and even occurred when the wheel-legs were connected directly to the chassis via a KCarWheelJoint. This was problematic because real Whegs robots maintain proper phasing even when stopped. It was determined that this problem was due to the nature of the KCarWheelJoint class. This class rotates a given Actor about a spin axis by applying a torque to overcome external torques. If the motor applies no torque, then the actor will be rotated about the motor's spin axis by all other external torques. Physically, a KCarWheelJoint is analogous to having an axle that rotates a wheel mounted in a perfectly frictionless bearing and motor. Real Whegs robots have a drive train that connects each wheel leg to a single drive motor and torsional springs that are pretensioned which causes them to maintain proper phasing even when the motor is not running. To fix this problem, a member function was added to the "Whegs" class that forces the KCarWheelJoint to achieve zero angular velocity by using a preset torque value when the robot is stationary (drive speed = 0 rad/s). This solution appeared to solve the problem.

C. Wheel-Legs Behaving as KActors Vs. Tires

Initially, the wheel-legs were implemented as KTires. However, when attempts were made to run the robot, the wheel-legs would rotate but not cause the robot to translate, resulting in the wheel-legs spinning in place and the robot itself not making any forward progress. It was determined that because the wheel-legs were not connected to KCarWheelJoints, KActor properties were used instead of KTire properties. The default KActor friction value is zero, which led to the wheel-legs perfectly sliding on a given substrate. An attempt to rectify this problem was made by implementing the solution proposed above: making each spoke a KTire that is connected to the central hub via a KCarWheelJoint. Because of the problems experienced in maintaining proper wheel-leg phasing mentioned in the section above, a function was added to the "Whegs" class that forces the spokes to maintain their initial orientation relative to their parent hub. This solution resulted in the wheel-leg spokes drifting into altered positions over time, particularly at higher drive speeds. An attempt to remedy this problem was made by increasing the torque used to maintain the spoke orientation. This resulted in the vehicle going through seemingly nonphysical end-over-end rotation at higher wheel leg drive speeds (about 15 rad/s and higher) while still translating forward, and did not appear to remove the drifting problem. It was found that the only apparent way to influence the problem was to increase the vehicle's inertia tensor or angular velocity resistance (KAngularDamping) Karma Parameters. These parameters only seemed to slow down the rotation. They also had to be raised to levels much higher than any other vehicle in the USARSim, including vehicles that are more massive such as the Hummer.

The nonphysical nature of this behavior and its solution prompted performing the validation with the wheel-legs behaving as KActors instead of KTires. This approach led to behavior that appeared to be more physically relevant when compared to the real vehicle. Also, as stated above, because this approach offers a narrower parameter search and because the wheel-legs on Mini-Whegs vehicles are not formal tires, it was felt that using the wheel-legs as KActors would provide a reasonable approximation that would allow for relatively simple but effective preliminary validation.

D. Effects of Individual Karma Parameters

The robot's speed and velocity components were plotted in both world and vehicle coordinates. These plots were used as a tool to help examine the effects of individual Karma Parameters on the robot's performance. Example plots are shown in Fig. 11.

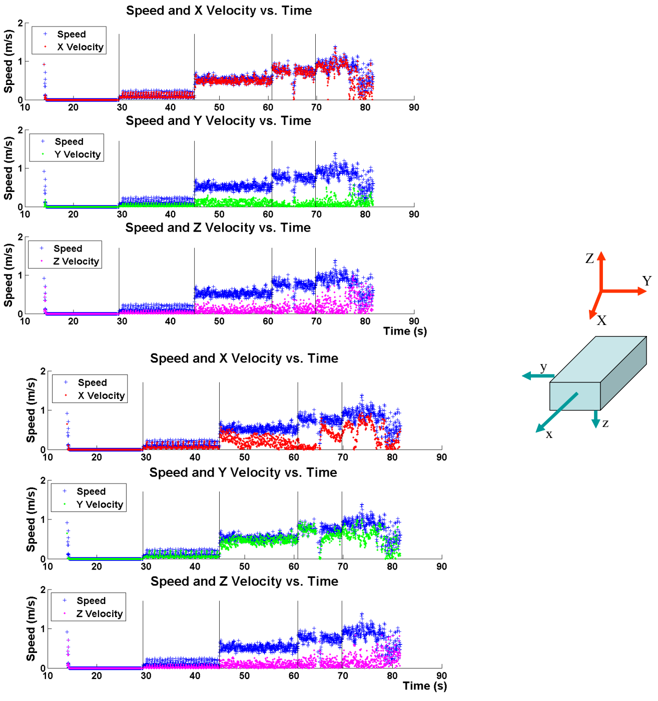


Fig. 11. Velocity in vehicle coordinates (top) and world coordinates (bottom). In vehicle coordinates, x is forward, y is starboard, and z is out the bottom of the vehicle. The absolute value of the velocity components is plotted here for comparison with the speed.

The vertical black lines are the times at which speed change commands were issued. The vehicle's initial speed (which occurs around 15 s in the plots shown above) is due to it spawning into the world. By looking at the body-fixed coordinate plot, it can be seen that the X-component of velocity in the vehicle coordinates is nearly identical to the speed, meaning that the vehicle is making forward progress and not translating to its left or right. In addition, there is a point around 65 s where the X-velocity and the speed both drop while the Z-velocity spikes. Upon comparison with the video data, it was observed that this was the location at which the robot flipped over, or end-over-ended. This kind of feature was present in all cases where the robot flipped over. The world coordinate plot illustrates the vehicle's tendency to move in a particular direction within the world. In this plot, the robot is initially heading mostly in the Y direction. However, after it flips over at around 65 s, it also begins to have motion in the X-direction as well. In addition to these kinds of observations, the data in the plots can be used to look at other phenomena such as the average speed of the vehicle for a given time interval and the variability of the data about the average.

The following Karma Parameters were singly modified while leaving all other parameters at their initial settings to determine their effects on the performance of the vehicle:

1) ChassisMass: This is the mass of the chassis. With the initial parameters that were set, it was found that altering this parameter did not appear to have a large impact on the overall performance of the vehicle in terms of being able to reach a top speed, or end-over-ending.

2) KMass: This is the mass of the Spacer Plates. This mass was varied to determine the effects of raising and lowering mass that is not coincident with the vehicle's center of mass. Initally, the mass of the chassis was very small, both compared to the spacerplates, and in an absolute sense, so this test also revealed how the vehicle's overall mass affected performance. When these masses were lowered to a value of 0.002 in the Unreal Unit System, the vehicle immediately began to end over end at higher drive speeds (10 rad/s and over). At mass values of 2 in the Unreal Unit System, the vehicle appeared to perform in a relatively predictable manner with only occasional end-over-end instances occurring at drive speeds between 25 rad/s and 30 rad/s.

3) KFriction: This is the friction present in the wheel legs. It was found that, as one would expect, higher values of friction (10 in this study) resulted in the wheel-legs not slipping on the substrate as much during walking. Visible slippage occurred with lower friction values (0.5 in the Unreal Unit System). Both of these friction values yielded roughly the same vehicle average speeds for a given drive speed. However, the variation about the average was much higher for the larger friction value. This was attributed to stronger braking forces in the step cycle. The wheel-legs provide both propulsive and braking forces, where braking occurs in the beginning of the stance phase, and propulsion occurs towards the end. Because the friction value is higher, both the brake and propulsion forces are increased. Therefore, when a wheel leg touches down, it is able to provide better traction to propel the vehicle, but also has a greater tendency to retard its motion initially.

4) KRestitution: This parameter is similar in concept to the coefficient of restitution used in collision analysis. A value of 1 corresponds to an elastic collision between two objects. Values less than one result in increasingly inelastic collisions. At a KRestitution value of 1 in the Unreal Unit System, as the vehicle's drive speed was increased, it appeared to have increasingly continuous elastic collisions with the ground while its forward speed appeared to reach a relatively constant value that became independent of the input drive speed. As a result, the vehicle's average speed at higher drive speeds seems to flat-line when compared to other tests. The average speed also had a great deal of variability for each drive speed.

5) Torsional Stiffness: The torsional stiffness of the rear wheel-legs was set to 250 in the Unreal Unit system for all runs. This value appeared to make the back wheel-legs rotate with the spacer plates under all circumstances. The front torsional stiffness of the front wheel-legs was adjusted to see how adjusting the stiffness affected their motion. As expected, higher values of stiffness led the wheel-legs to behave more like conventional wheels, where low stiffness values allowed the spring to "wind up" before rotating the wheel-legs. Excessively low values of stiffness cause wheel-legs to fall out of phase when the vehicle is spawned. At these low values, when a drive command was issued, the wheel-legs would not rotate at first. However, after the springs were deflected sufficiently, they would rotate forward to release the tension as one would expect.

E. Karma Parameter Search

After the effects of the above mentioned parameters were understood from the single parameter variations, testing was done to move the virtual robot towards matching real performance. Table 2 indicates the parameters that were changed.

	Karma Parameter (Unreal Unit System)	Initial Value	Current Value
Chassis	ChassisMass	0.00342	0.75
	MaxTorque	32000	50
	MotorTorque	2400	50
	KCOMOffset (X, Y, Z)	0	0.04464
	KCOMOffset (X, Y, Z)	0	0
	KCOMOffset (X, Y, Z)	0	0
Wheel Legs	Wheel-Leg Kfriction	1	0.75
	Wheel-Leg KRestitution	0	0.1
	Wheel-Leg Kmass	0.0008	0.08
Spacer Plate	Spacer Plate Kmass	1	0
	Spacer Plate KInertiaTensor(0) ~ Ixx	0.0035	0
	Spacer Plate KInertiaTensor(3) ~ Iyy	0.0066	0
	Spacer Plate KInertiaTensor(5) ~ Izz	0.0035	0

Table 2. Initial and Current Karma Parameter Values

Column 1 is a listing of the initial values of the Karma Parameters. Column 2 represents the current values that they have been adjusted to. As can be seen from Table 2, the following general changes were made. First, because spacer plates are not found on the real robot, their mass and inertia values were set to zero so that they would have no effect on the dynamic characteristics of the vehicle. Based on the results obtained from varying individual Karma Parameters, lowering the mass of the Spacer Plates caused severe end-over-ending of the vehicle. This prompted raising the ChassisMass property of the vehicle to 1 in the Unreal Unit System, which drastically reduced this problem. Based on testing of an actual Whegs robot on tile, it was observed that the wheel-legs slip during walking at higher speeds, similar to what can occur with lower values of the KFriction parameter. The vehicle also appeared to have a degree of elasticity with the ground, similar to when the KRestitution values were raised. Accordingly, the KFriction and KRestitution values were lowered and raised respectively. The center of mass of the vehicle (KCOMOffset) was not varied in the single parameter variation study. However, after examining a particular Mini-Whegs vehicle, it was found that many of the components such as the steering servo, steering mechanism, and battery are located towards the front of the vehicle. Also, the virtual vehicle still went into end-over-ending behavior more than was desired. Therefore, the KCOMOffset value was adjusted to bring the center of mass of the vehicle forward. This reduced the end-over-ending behavior slightly, but did not completely remove the problem. While the center of mass is not typically in the forward section of a Mini-Whegs vehicle, the change was made here to judge its impact on the performance. In addition to these parameters, the KCarWheelJoint motor torque and wheel-leg masses were also changed. The motor torque was ultimately lowered from its initial value of 2400 to 50 in the Unreal Unit System to give the vehicle a more realistic level of drive torque (for comparison, the Hummer uses a motor torque value of 2400). The wheel-leg masses were raised from 0.0008 to 0.08 (Unreal Unit System).

F. Performance Results

The parameter changes that were made appeared to move the virtual vehicle towards matching the performance of the real vehicle. With the final set of parameters given above, the robot was able to walk at near top speeds with occasional end-over-ending. It was also only able to end-over-end at walls with higher wheel-leg drive speeds (10 rad/s and up), which is normal Mini-Whegs behavior. In climbing tests, the robot was able to successfully surmount a 0.04 m obstacle with head-on and 30^{o} approaches.

IV. Discussion

Basic functionality modifications of existing USARSim base classes along with functionality implemented in the newly defined "Whegs" class appears to successfully replicate the general behaviors of torsional compliance and wheel-leg phasing found in Whegs robots. Karma Parameter modification based on physical reasoning and observation of real robots through videos and direct interaction appeared to result in improved, more realistic performance of the virtual robot in walking, running and basic climbing. In addition, the procedure used in the parameter modification yielded insight into how each individual parameter contributes to the overall performance of the vehicle. The procedure also yielded the velocity history of the vehicle in world and body coordinates, along with the speed of the vehicle, and the average speed for a given time increment. This data was used to determine if the vehicle is able to attain a particular drive speed, the degree to which the vehicle collides with the ground, and the vehicle's tendency to end-over-end.

While the accomplishments listed above are significant first steps towards creating an accurate representation of a Whegs robot in USARSim, there are many steps that can be taken to improve the virtual robot's performance. With respect to the real robot, high speed video capture methods can be used to obtain data and establish performance metrics that can also be measured within the simulation for more accurate benchmarking. In terms of the virtual robot, several steps can be taken including: making the wheel-legs function as actual tires, refining the behavioral characteristics of the "Whegs" class and investigating the use of more detailed and accurate static meshes. In terms of the Karma Parameters, the above tests can be conducted in more depth and expanded to better understand the effects of individual parameters on robot performance. More testing can also be done to better understand how individual Karma Parameters map to real world quantities. Also, while individual Karma Parameters can be varied, they are not necessarily independent, so the coupling between Karma Parameters needs to be understood. For the validation, a number of steps can be taken. Several map substrate surfaces can be made from KActors and tuned to match the performance of real surfaces such as concrete, tile, wood, etc. The virtual robot can then be tested and compared to the real robot on each of these surfaces, which would yield a better representation of the appropriate Karma Parameters for the robot. With more rigorous performance metrics defined, numerical methods could be employed to help determine how well the virtual robot matches the real robot's performance. All of these steps would lead to a more reliable and repeatable simulation.

ACKNOWLEDGEMENTS

We would like to thank Ben Balaguer and Richard Bachmann for their contributions in developing this work.

This research was performed under an appointment to the U.S. Department of Homeland Security (DHS) Scholarship and Fellowship Program, administrered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by Oak Ridge Associated Universities (ORAU) under DOE contract number DE-AC05-06OR23100. The opinions expressed here do not necessarily reflect the policies and views of DHS, DOE, or ORAU/ORISE.

REFERENCES

[1] Allen, T.J, Quinn, R.D., Bachmann, R.J., and Ritzmann, R.E. (2003) "Abstracted Biological Principles Applied with Reduced Actuation Improve Mobility of Legged Vehicles," IEEE International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas.

[2] Balakirsky S, Scrapper C, Carpin S and Lewis M. (2006) "USARSim: Providing a Framework for Multi-Robot Performance Evaluation," Performance Metrics for Intelligent Systems, Gaithersburg MD, USA., August 21-23, 2006.

[3] Balakirsky S, Scrapper C, Carpin S and Lewis M. "USARSim: A RoboCup Virtual Urban Search and Rescue Competition," Proceedings of SPIE, 2007.

[4] Morrey, J.M., Lambrecht, B., Horchler, A.D., Ritzmann, R.E., and Quinn, R.D. (2003) "Highly Mobile and Robust Small Quadruped Robots", IEEE International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas.

[5] Quinn, R.D., Kingsley, D.A., Offi, J.T. and Ritzmann, R.E., (2002), "Improved Mobility Through Abstracted Biological Principles," IEEE Int. Conf. On Intelligent Robots and Systems (IROS'02), Lausanne, Switzerland.

[6] Quinn, R.D., Nelson, G.M., Ritzmann, R.E., Bachmann, R.J., Kingsley, D.A., Offi, J.T. and Allen, T.J. (2003), "Parallel Complimentary Strategies For Implementing Biological Principles Into Mobile Robots," Int. Journal of Robotics Research, Vol. 22 (3) pp. 169-186.

[7] Saranli, U., Buehler, M. and Koditschek, D. (2001) "RHex A Simple and Highly Mobile Hexapod Robot". International Journal of Robotics Research, 20(7): 616-631.

[8 Schroer, R.T., Boggess, M.J., Bachmann, R.J., Quinn, R.D., and Ritzmann, R.E. (2004) "Comparing Cockroach and Whegs Robot Body Motions," IEEE Conference on Robotics and Automation (ICRA '04), New Orleans.

[9] "Unreal Developer Network Karma Reference." Unreal Developer Network. (9/25/2007) http://wiki.beyondunreal.com/wiki/

[10"Unreal Wiki: The Unreal Engine Documentation Site." (DATE HERE) http://wiki.beyondunreal.com/wiki/

[11] USARSim (9/25/2007) http://sourceforge.net/projects/usarsim

[12] MOAST (9/25/2007) http://sourceforge.net/projects/moast

[13] FRAPS (9/25/2007) http://www.fraps.com/

Maze Hypothesis Development in Assessing Robot Performance During Teleoperation

Salvatore Schipani
National Institute of
Standards and Technology
100 Bureau Drive, MS-8940
Gaithersburg, Md., U.S.A.
salvatore.schipani@nist.gov

Elena Messina
National Institute of
Standards and Technology
100 Bureau Drive, MS-8230
Gaithersburg, Md., U.S.A.
elena.messina@nist.gov

***Abstract*—** National Institute of Standards and Technology (NIST) personnel had the opportunity to assess 14 prospective Urban Search and Rescue (USAR) robots, for the purposes of developing performance standards which currently do not exist. During this exercise, a maze configuration – hypothesized as potentially valid test methodology – was assessed. Among the findings, resultant significant differences in completion and decision making times facilitated classifying platforms based on performance. Also revealed was that errors in navigation and encounters with walls correlated with times taken in making decisions… the longer it took to make a decision, the greater the chance this decision was incorrect. Results validated the hypothesis of a maze as beneficial in eliciting data necessary for human controlled robot performance assessment.

***Keywords*:** maze, metrics, performance standards; robotics; situation awareness; teleoperation

I. INTRODUCTION

Test performance standards for application-specific Urban Search and Rescue (USAR) robots providing valid replicable assessment measures do not exist, thus little or no guidance may be offered to local, state, or federal agencies regarding their utilization or procurement. In 2004, the Department of Homeland Security (DHS) Science and Technology (S&T) Directorate initiated an effort with the National Institute of Standards and Technology (NIST) to formulate comprehensive criteria related to the development, performance testing, and certification of available and anticipated robotic technologies, specifically directed toward application in USAR scenarios. To encourage collaboration between USAR responders and system developers, and in hopes of generating standards consensus among those interested, a third response robot evaluation exercise was conducted by NIST at the Montgomery County Fire Rescue Training Academy in Rockville, Maryland, particularly targeting the needs of DHS/Federal Emergency Management Agency (FEMA) USAR professionals. Operational standards deemed necessarily of concern included mobility, sensing, navigation, planning, integration into operational caches, and consideration of the human factor.

Individual characteristics of current production robots utilized for USAR vary. In light of recent national security concerns, this reality brings to the forefront a necessity for categorizing the operational capabilities of tools and methods used to placate concerns. Any attempt at the organization of such information must address the identifiable requirements of emergency response professionals, and offer recommendations for system attribute improvement as discovered. In August of 2006, NIST personnel had the opportunity to assess 14 robots with potential for application during USAR situations based on visual sensors, mobility, logistic cache packaging, radio communications, and human factors in operations. This document reports on one proposed measure of performance, a subset of the decision making process referred to as operator time to acquire situation awareness, when attempting to teleoperate a robot within a maze, a scenario hypothesized as valid test methodology given observed apparatus methods of control and assumed tasks.

A. Background

1) Maze Rationale:

Mazes derived directly from their descendents, ancient labyrinth designs. This symbol and its family of derivatives may be traced back over 3500 years, however origins remain a mystery. As opposed to a maze, labyrinths have no false pathways or dead ends, but rather consist of one single meandering way leading from entrance to center. Conversely, mazes may possess many paths, enticing or impairing anyone attempting to maneuver through. These have become accepted exercises in direction finding, providing paths to follow, some correct and others erroneous. As such, they are considered highly respectable tests of navigational skills, and attempted by many.

Correlations between maze performance and traditional psychometric measures of spatial ability have affirmed the relationship [1,2], especially as vestibular information from the inner ear as well as kinesthetic feedback from muscles has been shown to provide important cues regarding direction of heading and distance information [3,4]. The rationale becomes particularly acceptable subsequent to reviews of factor analytical studies for large spatial batteries yielding multiple spatial dimensions [5,6,7]. Optic flow also provides motion and movement cues necessary to navigate through environments, offering a visual analyses of motion which we have come to anticipate and rely on. Unfortunately, during teleoperation, such visual cues become the *only* aid presented [8,9], rendering tasks such as remote control especially difficult. Given that these cues are often disturbed during teleoperation due to issues in transmission, it should be expected that maze navigation become increasingly difficult.

2) Acquiring Situation Awareness:

Though several definitions of Situation Awareness (SA) are posed in literature [10,11,12], SA is normally defined in terms of goals with particular decision tasks directed to a specified effort [13,14]. One definition offered, encompassing the essence of what most researchers care to relate, is *"The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future"* [15]. Endsley and Garland [14] further define levels of SA as: *Level 1*, the perception of cues: *Level 2*, an extension of cue perception, including the integration of multiple pieces of information plus the determination of their relevance to goals. Here, meaning must be considered as subjective interpretation (awareness) and objective significance (situation) [16, page 3], so that at this level one is able to derive operational relevance and significance from prior data, and; *Level 3*, the ability to forecast future events. SA is normally depicted as an operator's internal state model within an environment [17,14], causing designers to consistently question how well particular systems support one's ability to acquire necessary information. This design concern is exaggerated in dynamic situations and under operational constraints, thus observing the acquisition and eventual degree of SA has become a frequently used measure of performance.

Time has been shown a critical affecting factor in acquiring both *Levels 2* (comprehension) and *3* (future event projection) SA [18,19,20]. This is particularly the case in teleremote operations, as operator SA must be derived from a combination of the environment and integrated system's displays, and then interpreted by the operator at afforded instances and in short intervals [14]. Here, sufficient information must be provided through a remote interface so as to compensate for cues once perceived directly [21], an unfortunate scenario commonly found deficient. The collection of whatever information presented is assumed a subset of that derived from the environment and internal system parameters, however only a portion may be displayed via existing (visual) interfaces. With the majority of teleoperated systems currently deployed, operators are given minimal control of which information may be collected other than that presented via the visual channel, and are often restricted in transmitting commands to request further knowledge arrived at in such ways as by the autonomous selection of directions of traverse or specifying areas of sensor coverage [22]. Such deficiencies in data acquisition not only lengthen the time required for information collection, but also inhibit assimilation.

In goal driven processing such as that which takes place during teleoperation, an operator actively seeks information required for attainment of the goal, during which the mental model is claimed as existing underlying knowledge therefore the basis for SA [23]. Smith and Handcock (1995) support this view of SA as behavior directed toward goal achievement, describing it as the *"…up-to-the minute comprehension of task relevant information"*. Referred to as cognition-in-action, Lave [24] claims *"SA fashions behavior in anticipation of the task-specific consequences of alternative actions"*. Over time, a pattern-recognition thus action-selection sequence becomes routine, developing to a level of response automaticity [25]. Such automaticity can positively affect SA by reducing demands on limited attentional resources, but only if proper information is retrieved, comprehended, and adequately assimilated. When one's goal is to eventually emplace a system (robot) at a specified location, an internal model of previously traversed terrain with appropriate continued or corrected model for subsequent route direction becomes essential. This has been shown difficult when using existing teleremote visual displays due to inadequate cueing for guidance, and lack of available space for displaying previous information, thus SA is compromised.

II. METHOD

A. Participants

Personnel operating robots during this exercise were engineering professionals representing their respective product. Each had extensive experience not only in robot operation, but also in development. Additionally, each vendor-operator was made aware that the performance of their product would be compared to competitors during the exercise, thus it behooved them to offer their best operator for the assessment. Personal observations substantiated the fact that each participant could be considered proficient in robot manipulation, thus the level of expertise was deemed a fixed factor. In all, 14 participants were involved, one each from all robot vendors appearing for the test.

B. Materials

1) Test Course:

In this particular maze (see Figure 1), there exists one possible solution with only a single main branch leading to correct termination, having an approximate solution length of 2,117.29 centimeters (833.58 inches) which consists of 21 wall segments equating to 21.17 meters (69.47 feet).

Traveling forward, the maze possess three left turns, three right turns, three straight-aways, two left curves, no right curves, two irregular curves, two ramps, four junctions, no crossroads, loops, or roundabout passages, and two dead-ended isolation points (designated points 1 and 2 in the Figure 1 diagram). Additionally, two route enticements were constructed at which light was visible hinting at clear passage however actually blocked, with only short possible deviation lengths within the two provided false passages of 115.57 and 346.71 centimeters (45.5 and 136.5 inches).

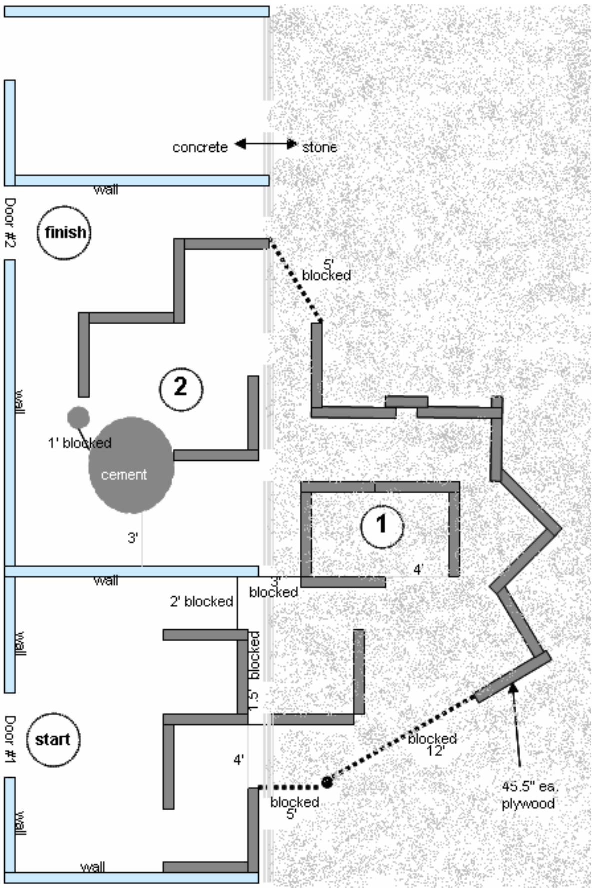


Figure 1. Original Maze Configuration (not to scale)

2) Robots:

Following are descriptions of the 14 participating robots, accompanied by individual dimensions (taken from the NIST draft publication "Response Robots – DHS/NIST Sponsored Evaluation Exercises" Pocket Guide, Version 2006.1). To provide for anonymity, each has been designated a number in place of name.

- Robot #1: Width 57.15 centimeters (22.5 inches), length 86.36 centimeters (34 inches), height 38.1 centimeters (15 inches), weight 52.16-63.5 kilograms (115-140 pounds), turning diameter 0 centimeters (0 inches) (skid-steer, tracks), maximum speed 8.369 kilometers per hour (5.2 miles per hour), non tethered (tether option), remote teleoperation control, sensor include black and white camera (optional biological, chemical, and temperature sensors), five degrees-of-freedom 132.08 centimeters (52 inches) horizontal reach end effector (*i.e.*, manipulator);
- Robot #2: Width 30.988 centimeters (12.2 inches), length 42.164 centimeters (16.6 inches), height 15.24 centimeters (6 inches), weight 6.35 kilograms (14 pounds), turning diameter 0 centimeters (0 inches) (skid-steer), maximum speed 2.286 meters per second (7.5 feet per second), non tethered, remote teleoperation control, sensor include black and white camera (with options for thermal, acoustic, infra-red, and visual wide-angle sensing), no end effector;
- Robot #3: Width 25.4 centimeters (10 inches), length 35.56 centimeters (14 inches), height 16.51 centimeters (6.5 inches), weight 6.35 kilograms (14 pounds), turning diameter 50.8 centimeters (20 inches), maximum speed, 1.829 meters per second (6 feet per second), non tethered, remote teleoperation control, sensors include color and infrared cameras, no end effector;
- Robot #4: (no data available);
- Robot #5: Width 55.88 centimeters (22 inches), length 68.58 centimeters (27 inches), height 63.5 centimeters (25 inches), weight 56.7 kilograms (125 pounds), tracked skid-steer turns on center, maximum speed 10.46 kilometers per hour (6.5 miles per hour), non tethered, eyes on and remote teleoperation control with way-point following and drive intent, sensors color video camera and laser range scanner, no end effector;
- Robot #6: Width 57.15 centimeters (22.5 inches), length 86.36 centimeters (34 inches), height 63.5 centimeters (25 inches), weight 52.16-63.5 kilograms (115-140 pounds), turning diameter 0 centimeters (0 inches) (skid-steer, tracks), maximum speed 8.369 kilometers per hour (5.2 miles per hour), non tethered (tether option), remote teleoperation control, sensor include black and white camera (optional biological, chemical, and temperature sensors), five degrees-of-freedom 132.08 centimeter (52 inch) horizontal reach end effector;
- Robot #7: (no data available);
- Robot #8: Width 34.29 centimeters (13.5 inches), length 52.07 centimeters (20.5 inches), height 30.48 centimeters (12 inches), weight 11.34 kilograms (25 pounds), turning diameter 0 centimeters (0 inches) (skid-steer, tracks), maximum speed 6.437 kilometers per hour (4 miles per hour), non tethered, no tether, eyes on and remote teleoperation control, sensor black and white camera, no end effector (i.e., manipulator);
- Robot #9: Width 53.34 centimeters (21 inches), length 76.2-86.36 centimeters (30-34 inches), height 30.48 centimeters (12 inches), weight 27.67 kilograms (61 pounds), turning diameter 0 centimeters (0 inches) (skid-steer, tracks), maximum speed 3.219 kilometers per hour (2 miles per hour), non tethered, fiber optic cable tether (for data, video, and audio), remote teleoperation control, sensors include black and white camera (optional biological, chemical, and radiological sensors), five degrees-of-freedom 111.76 centimeter (44 inch) end effector;
- Robot #10: Width 40.64 centimeters (16 inches), length 63.5 centimeters (25 inches), height 19.304 centimeters (7.6 inches), weight 11.34 kilograms (25 pounds), turns in place, maximum speed 1.341 meters per second (4.4 feet per second), non tethered, remote teleoperation and telemetry control, sensor black and white camera, end effector (i.e., manipulator) six degrees-of-freedom with 106.68 centimeter (42 inch) reach;
- Robot #11: Width 40.64 centimeters (16 inches), length 68.58 centimeters (27 inches), height 19.05 centimeters (7.5 inches),

weight 21.77 kilograms (48 pounds), turning diameter 0 centimeters (0 inches), maximum speed 8.047 kilometers per hour (5 miles per hour), non tethered, remote teleoperation control, sensor black and white camera on short non-extending boom, no end effector;

- Robot #12: Width 40.64 centimeters (16 inches), length 68.58 centimeters (27 inches), height 19.05 centimeters (7.5 inches), weight 21.77 kilograms (48 pounds), turning diameter 0 centimeters (0 inches), maximum speed 8.047 kilometers per hour (5 miles per hour), non tethered, remote teleoperation control, sensor black and white camera on three-rod extending boom, no end effector;
- Robot #13: Width 50.8 centimeters (20 inches), length 55.88 centimeters (22 inches), height 45.72 centimeters (18 inches), weight 6.804 kilograms (15 pounds), turning diameter 0 centimeters (0 inches), maximum speed 5.633 kilometers per hour (3.5 miles per hour), non tethered, remote teleoperation control, sensor black and white camera, no end effector;
- Robot #14: Width 27.432 centimeters (10.8 inches), length 42.672 centimeters (16.8 inches), height 13.97 centimeters (5.5 inches), weight 6.35-9.072 kilograms (14-20 pounds), turning diameter 0 centimeters (0 inches) (skid-steer, tracks), maximum speed 0.4572 meters per minute (1.5 foot per minute), 30.48 meter (100 foot) polyurethane multi-cord tether, remote teleoperation and eyes-on control, sensor black and white tilt camera, no end effector.

III. PROCEDURE

Participants were directed – upon the experimenter command "begin" – to teleoperate assigned robotic platforms traversing pathways through the unfamiliar maze, and do so within the shortest time possible. They were further instructed to operate carefully enough to limit or avoid encounters with path walls. Their informed consent to participate and to allow a video record made of their system was agreed upon prior to test initiation, at which time operator sightedness was screened. Participants were permitted to ask questions concerning test methods and purpose prior to testing, or at any time during the test. They were instructed that they were to complete four iterations, two in forward and two in reverse, until reaching their goals which were open doorways located at the beginning and end of the maze.

A. Data Collection

Time data collection was recorded in seconds and performed manually utilizing hand-held stop watches (one recording total maze traverse time, the second monitoring time spent in decision points), and on digital video in order that post-test evaluations of performance could be made. Video records were taken via hand held roving camera, with camera person consistently positioned behind the robot thus completely out of robot camera view to ensure that no visual cues were offered to operators.

IV. EXPERIMENTAL DESIGN

The experiment was treated as a 2 *x* 2 *x* 1 factorial, where two levels of traverse exist (forward and reverse), with two instances of dead ended isolation points, and this applied between the performance of 14 robotic platforms given one level of operator proficiency.

A. Dependent Measures

Dependent measures were averaged "maze completion times" traveling forward and reverse, averaged "decision making times" recorded at points specified within the maze, "errors" in direction of traverse when exiting aforementioned decision points, and observed "encounters" made with maze walls.

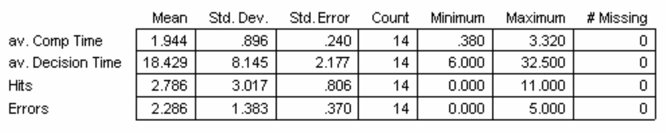
Times were recorded for total maze completion in each direction (separate forward and reverse recordings), and during instances at which robots entered into and lingered in designated dead-ended isolation points. For total completion time, recording began as test director instructed participants to "begin" each trial, and ended once the robot reached the step-sill of exit doors located at either end of the maze. Each participant completed two forward and two reverse iterations.

For instances in which participants entered a dead-ended isolation area (*i.e.*, decision eliciting 'traps'), total time spent within was recorded. Time data collection for this began when the most forward portion of a robot crossed a horizontal imaginary line at the entrance of the isolation area, and ended as the most forward portion again crossed this line exiting. This data was treated as the time necessary for participants to gain situation adequate awareness, sufficient for participants to realize that they had entered a dead end in the maze and to reach a decision on how to properly exit.

As participants exited dead-ended decision points, their direction of traverse was recorded for correctness. The accurate direction could be determined by experimenter observation as being the most obvious direction of course traverse within which one might successfully complete the maze. Finally, robot encounters with walls (*e.g.*, "hits") were recorded as each participant teleoperated through pathways.

V. RESULTS

Following (see Table 1) find descriptive statistics for averaged *Maze Completion Time*, *Decision Making Time*, wall *Hits* (encounters), and *Errors* in direction traversed.



	Mean	Std. Dev.	Std. Error	Count	Minimum	Maximum	# Missing
av. Comp Time	1.944	.896	.240	14	.380	3.320	0
av. Decision Time	18.429	8.145	2.177	14	6.000	32.500	0
Hits	2.786	3.017	.806	14	0.000	11.000	0
Errors	2.286	1.383	.370	14	0.000	5.000	0

Table 1. Descriptive Statistics
(Times in seconds, Hits & Errors in unit segments)

Figure 2 presents maze completion times, showing robots 2, 8, 9, 10 and 12 displaying lowest times to complete the maze (averaging 1.14 minutes, or 68.4 seconds), and robots 5, 6, 7 the highest (averaging 3.23 minutes, or 193.8 seconds).

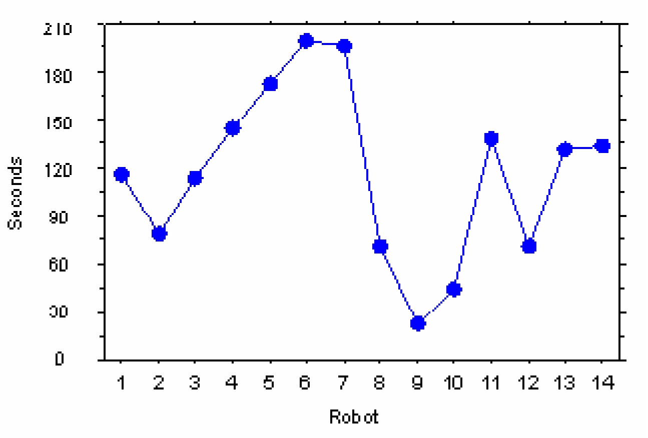


Figure 2. Maze Completion Times

Robots 1, 8 and 9 displayed lowest decision making times (*i.e.,* Situation Awareness gaining time) averaging 6.93 seconds, and robots 6, 11, and 13 the highest averaging 31 seconds (see Figure 3).

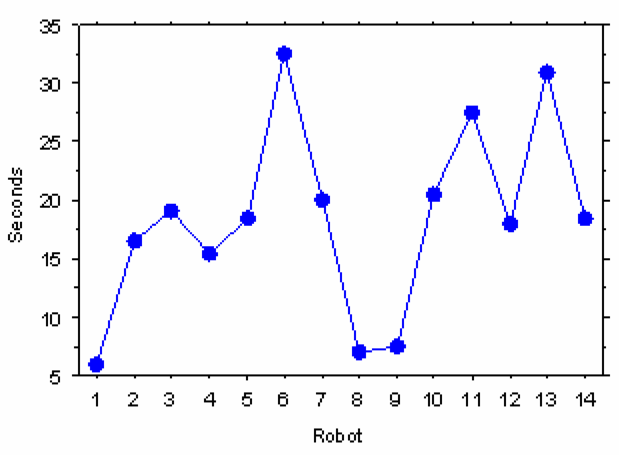


Figure 3. Decision Making Times

No statistically significant difference found among robots for average maze completion times (*p* = 0.68), the most frequently attained ranging from 2.14 to 2.48 minutes (128.4-148.8 seconds). It may be assumed that – being a first attempt – the current maze configuration did not provide sufficient distance to evoke performance anticipated. Future maze investigations employing increased areas of traverse should resolve this issue. However, three categories may be delineated from the data when observing performance groupings which ranged from slightly greater or less than 1.0, on average 2.2, and slightly greater or less than 3.0 minutes (60, 132, and 180 seconds respectively) (see Figure 4). There was a significant difference found between *forward* and *reverse* times to complete the maze (*p* = 0.003). Times in reverse were shorter, obviously an indication that operators were becoming familiar with the test course.

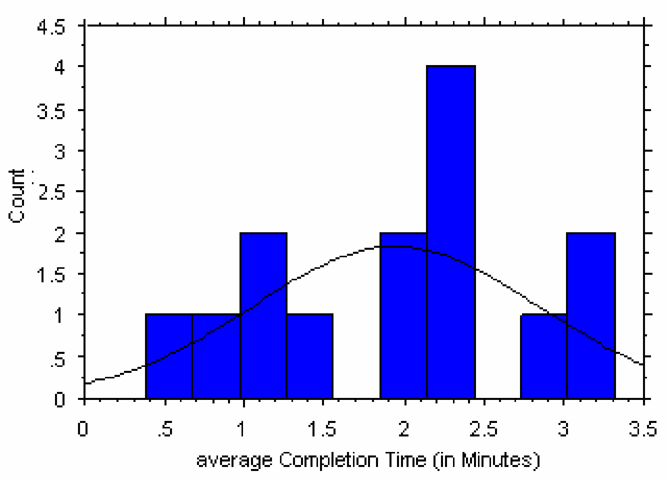


Figure 4. Histogram of Distributed Completion Times

There was a significant difference found among robots concerning averaged decision making times (*p* = 0.001), as individual attributes of particular platforms apparently aided or hindered performance during the challenge. There was not a significant difference found between times to decide at isolation area 1 versus 2 (*p* = 0.891), revealing the two similar in nature. The most frequently attained decision making times ranged from 16.6 to 19.25 seconds. Here again, robots could be grouped per three categories of performance of from slightly greater or less than 7.5, averaged 18.5, or slightly greater or less than 30 seconds (see Figure 5).

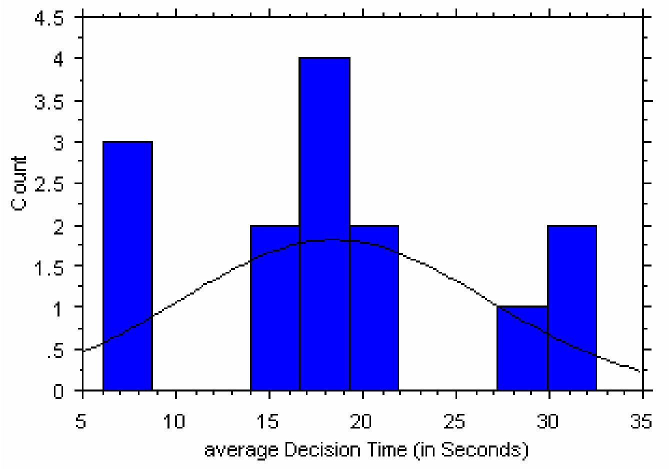


Figure 5. Histogram of Distributed Decision Times

A significant difference was found among robots concerning hits (wall encounters) (*p* = 0.001). In reviewing video recordings, it would appear as if particular robots acted out-of-control due to inferior or transmission lagged control response, no or poor methods of halting forward movement, or poor camera views provided the operator.

There was a significant difference found among robots concerning errors (*p* = 0.048). Errors were also found correlated with increased times spent in making decisions (r = 0.67). This would appear to support the notion that the longer it took to make a decision as to which direction to move next, the more this decision (the direction of traverse selected) was found incorrect. No significant correlations were observed between averaged completion times and decision making times, revealing these entities distinct (r = 0.543). However, averaged wall hits data correlated highly with errors made in correct direction of traverse (r = 0.864), suggesting confusion in the selection of subsequent travel direction due to post-collision trauma.

For comparative purposes, individual performance is displayed in Table 2.

Robot	Comp. Time	Decision Time	Errors	Hits
1	average	best	best	best
2	average	average	average	best
3	average	average	average	best
4	average	average	average	best
5	poor	average	average	best
6	poor	poor	poor	poor
7	poor	average	poor	poor
8	best	best	best	best
9	best	best	best	best
10	best	average	average	best
11	average	poor	poor	poor
12	best	average	average	best
13	average	average	poor	best
14	average	average	average	best

Table 2. Performance as a function of Dependent Measures

VI. DISCUSSION

At present, performance standards for Urban Search and Rescue (USAR) designated robots are nonexistent, thus little guidance may be offered to local, state, or federal agencies regarding their purchase or use. A precursor to successful search and rescue operations if employing a robot is the ability to teleoperate the system satisfactorily, attaining directional cures from the remote visual display as possible. When one is driving, vestibular information and kinesthetic feedback provide additional cues regarding direction. However, during teleoperation, the *only* cues available are those presented visually, yet sufficient information must be attained via a remote interface in order to compensate thus discern most advantageous pathways. Intensifying this effort, situation awareness in such circumstances must be attempted while on-the-move, which becomes defined in terms of goal achievement with time the critical factor affecting acquisition. This document reports on one scenario hypothesized as valid methodology for assessing performance of such platforms, a maze test configuration employed as a navigation exercise.

Data collected included time to complete the maze, and also that necessary for gaining situation awareness when entrapped in either of two predestinated dead-ended isolation points. Data also included recordings of maze wall encounters, and errors made in direction of traverse. Digital video recordings were taken to enable *post hoc* analyses. Participants were directed to teleoperate their assigned platforms through the maze in the shortest time possible, while avoiding encounters with walls. Fourteen robots, potential candidates for deployment in USAR scenarios, were involved. Participants operating the robots were engineering professionals representing their respective product, each possessing extensive experience both in operation and platform development. Results revealed significant differences in time to gain situation awareness (*p* = 0.001), encounters with walls (*p* = 0.001), and errors made in direction of traverse (*p* = 0.048). Also uncovered was that increased times spent in making decisions correlated with erroneous subsequently selected directions of traverse (r = 0.67), supporting the notion that the longer it took to make a navigational decision the more this decision could be found incorrect. Finally, encounters with walls correlated highly with errors made in direction of traverse (r = 0.864), revealing confusion as a result of post-collision trauma.

Given results of the current exercise, utilization of a maze test approach for evaluating robot teleoperation appears rational, as the scenario elicited data sufficient to examine performance as intended. Forthcoming endeavors are expected to include increased maze distances and complexity, to ensure that appropriate pragmatic assessments may be made.

Anticipations are to submit the maze hypothesis to tests of validity and reliability in the near future. Generally accepted validity determinations involve criterion-oriented procedures such as predictive and concurrent, or are else-wise considered either content or construct [26]. For the test method in question, a predictive approach to validation appears most logical, as criterion-oriented validity "*involves the acceptance of a set of operations as an adequate definition of whatever is to be measured*." [27]. This will be attempted per performance criterion found necessary via repeated investigation, as well as by exploiting guidance offered from emergency response professionals. Reliability assessments should establish whether this type examination measures consistently. Concurrently, appropriate levels of maze complexity will be evaluated, and mathematical formulas aiding in maze construction developed for use by those not capable of testing at a NIST designated arena. Subsequently, results will be submitted through appropriate committee of the American Society for Testing and Materials (ASTM) to attain consensus as a national standard, as NIST personnel explore supplementary measurement methods deemed essential.

VII. REFERENCES

[22] Adams, M.J., Tenney, Y.J., & Pew, R.W. (1995). Situation awareness and the cognitive management of complex systems. *Human Factors*, *37*(1).

[26] Anastasi, A. (1950). The concept of validity in the interpretation of test scores. *Educational Psychological Measurement*, *10*, 67-78.

[27] Bechtoldt, H. P. (1951). Selection. In S.S. Stevens (Ed.), *Handbook of Experimental Psychology*. New York: Wiley, 1237-1267.

[5] Borich, G.D. & Bauman, P.M. (1972). Convergent and discriminant validation for the French and Guilford-Zimmerman spatial orientation and spatial visualization factors. *Educational and Psychological Measurement*, *32*, 1029-1033.

[3] Chapuis, N. & Scardigli, P. (1993). Shortcut ability in hamsters (Mesocricetus auratus): The role of environmental and kinesthetic information. *Animal Learning and Behavior*, *21*, 255-265.

[19] Cohen, M.S., Freeman, J.T., & Wolf, S. (1996). Metacognition in time-stressed decision making: Recognizing, critiquing, and correcting. *Human Factors*, *38*(2), 206-219.

[12] Dominguez, C. (1994). Can SA be defined? In, M.Vidulich, C.Domingues, E.Vogel, & G.McMillan (Eds.), *Situation awareness: Papers and annotated bibliography* (AL/CF-TR-1994-0085), 5-15. Wright-Patterson AFB, OH.: Armstrong Laboratory.

[6] Ekstrom, R.B., French, J., Harman, H.H., & Dermen, D. (1976). *Kit of Factor Referenced Cognitive Tests*. Princeton, MJ: Educational Testing Services.

[15] Endsley, M.R. (1988). Design and evaluation for situation awareness enhancement. In, *Proceedings of the Human Factors Society 32nd Annual Meeting* (97-101). Santa Monica, CA.: Human Factors Society.

[14] Endsley, M.R. & Garland, D.J. (Eds.), (2000). *Situation Awareness Analysis and Measurement*. Mahwah, N.J.: Lawrence Erlbaum Associates.

[18] Endsley, M.R., Farley, T.C., Jones, W.M., Midkiff, A.H., & Hansman, R.J. (1998). *Situation awareness information requirements for commercial airline pilots* (ICAT-98-1). Cambridge, MA.: Massachusetts Institute of Technology International Center for Air Transportation.

[20] Endsley, M.R. & Robertson, M. (1996). Team situation awareness in aviation maintenance. In, *Proceedings of the 40th Annual Meeting of the Human Factors and Ergonomics Society* (1077-1081). Santa Monica. CA.: Human Factors and Ergonomics Society.

[4] Etinne, A.S., Teroni, E., Maurer, R., Portenier, V, & Saucy, F. (1985). Short distance homing in a small mammal: The role of exteroceptive cues and path integration. *Experientia*, *41*, 122-125.

[16] Flach, J.M. (1995). Situation awareness: Proceed with caution. *Human Factors*, *37*(1), 149-157.

[10] Fracker, M.L. (1988). A theory of situation assessment: Implications for measuring situation awareness. In, *Proceedings of the Human Factors Society 32nd Annual Meeting* (102-106). Santa Monica, CA.: Human Factors Society.

[1] Gaulin, S.J.C & Fitzgerald, R.W. 1986). Sex differences in spatial ability: An evolutionary hypothesis and test. *American Naturalist*, *127*, 74-88.

[17] Gibson, J., Orasanu, J., Villeda, E., & Nygren, T.E. (1977). Loss of situation awareness: Causes and consequences. In, R.S. Jensen & R.L.A. (Eds.), *Proceedings of the 8th International Symposium on Aviation Psychology* (pp. 1417-1421). Columbus, OH.: The Ohio State University.

[25] Logan, G.D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*(4), 492-527.

[24] Lave, J. (1988). *Cognition in practice: Mind, mathematics and culture in everyday life*. Cambridge: Cambridge University Press.

[21] Marshak, W.P., Kuperman, G., Ramsey, E.G., & Wilson, D. (1987). Situational awareness in map displays. In, *Proceedings of the Human Factors Society 31st Annual Meeting* (533-535). Santa Monica, CA.: Human Factors Society.

[7] Michael, W.B., Zimmerman, W.S., & Guilford, J.P. (1951). An investigation of the nature of the spatial relations and visualization factors in two high school samples. *Educational and Psychological Measurement*, *11*, 561-577.

[8] Mittlestaedt, H. (1983). The role of multimodal convergence in homing by path integration. In *Multimodal Convergences in Sensory Systems*, E.Horn (Ed.). New York: Gustav Fischer Verlag, 197-212.

[23] Mogford, R.H. (1977). Mental models and situation awareness in air traffic control. *International Journal of Aviation Psychology*, *7*(4), 331-342.

[11] Sarter, N.B. & Woods, D.D. (1991). Situation awareness: A critical but ill-defined phenomenon. The *International Journal of Aviation Psychology*, *1*(1), 45-57.

[2] Silverman, I. & Eals, M. (1992). Sex differences in spatial abilities: Evolutionary theory and data. In the *Adapted Mind*, J..H. Barkow, L.Cosmides, and J.Tooby (Eds.). New York: Oxford, 533-549.

[13] Smith, K. & Hancock, P.A. (1995). Situation awareness is adaptive, externally directed consciousness. *Human Factors*, *37*(1), 137-148.

[9] Teroni, E, Portenier, V, & Etienne, A.S. (1987). Spatial orientation of the golden hamster in conditions of conflicting location-based and route-based information. *Behavioral Ecology and Sociobiology*, *20*, 389-397.

Human System Performance Metrics for Evaluation of Mixed-Initiative Heterogeneous Autonomous Systems

Lisa Billman
ARINC Engineering Services
lbillman@arinc.com

Marc Steinberg
Naval Air Systems Command/Office of Naval Research
marc.steinberg@navy.mil

Abstract-This paper describes a set of human system performance metrics and their implementation and use in a series of operator experiments for mixed initiative control of multiple heterogeneous unmanned systems. The focus of the work is on technologies that support the control of five to ten air, sea, and undersea vehicles with a common human interface. The individual systems have significant differences both physically and with regards to their on-board level and type of autonomy. This includes some ability of the operator to modify the autonomy levels relative to particular types of autonomous decision-making. This paper will describe the set of metrics and experience in applying them including implementation factors and their utility. Finally, it will describe some lessons learned.

Keywords: *UAV, USV, UUV, Mixed-Initiative, Human System Performance Metrics*

I. INTRODUCTION

The Naval Intelligent Autonomy program is developing and demonstrating autonomous control and human interface technologies that support management of five to ten heterogeneous unmanned systems by a single operator [1-2]. Accomplishing this requires significant increases in autonomous control and greatly reducing the need for human intervention in the system as compared with many current systems that require one or more dedicated and skilled operators to control even a single system. However, this does not mean that a desired goal of the work is to eliminate the need for human interaction with the system. The types of applications being looked at are long, complex missions with many interdependencies that will sometimes require significant human collaboration with the autonomous systems in order to coordinate unmanned systems planning and execution with changing situations in a dynamic operational space. Further, the difficulty of the types of missions being examined are such that they cannot currently be solved completely with autonomy and will require making use of human strengths such as tactical understanding, judgment, and decision-making while also minimizing the impact of human weaknesses [3]. This can be a particular problem because there may be significant differences between how human operators and highly advanced autonomous systems conceptualize planning and execution. In addition, designers of autonomous systems are unlikely to have a full understanding of how users in the field will want to utilize these systems and there will be times when unforeseen problems will arise. As a result, it will be important for operators to be able to interact with these systems at a variety of different levels within different control loops. Thus, there is a strong need to determine how best to design the entire autonomous system in a way that supports the role of the human in the system and not just assume that this can be solved with a good user interface design.

In the past, many approaches to autonomous systems metrics that relate to the operator have assumed that the ultimate goal of autonomy is to get the operator out of the loop. For this program, it is critical to examine the performance of the total system including the human in-the-loop as well as examine what factors are impacting on the ability of the human to effectively collaborate with the automation. The particular type of system being examined has some complex features that can make it difficult to evaluate. First, it deals with highly heterogeneous vehicles including air, sea surface, and undersea vehicles. These vehicles have significant physical differences, operate in very different environments, and have different types of on-board autonomous control systems. Second, there are significant differences in communications with each type of platform. This can range from a high altitude Unmanned Air Vehicle (UAV) that may have relatively good communications with an operator to an Unmanned Undersea Vehicle (UUV) or small UAV that may have extended periods without communications or relatively low bandwidth communications. The vehicles also may be widely distributed geographically throughout the course of the mission rather than all operating within close proximity to each other. Finally, this program is focusing on vehicles that can operate with a fairly high degree of autonomy. Operators can adjust the levels of autonomy for individual vehicles or classes of vehicles so that they respond to different types of contingencies in different modes such as operation by consent, operation by exception, and fully autonomous operation. One consequence of this is it may be important to examine factors that are difficult to measure such as the operator's trust and mental model of the autonomous system.

This paper presents one approach to a set of metrics that can be used to evaluate human system performance for very complex autonomous systems and also describes experience in applying these metrics in a series of naval operator evaluations [4]. It is important to note that these are not meant to be solely human performance metrics, or measures of hardware/software performance. Rather, these metrics are intended to assess human-in-the-loop system performance, which includes hardware, software, liveware, and the environment. Metrics were chosen based on factors such as objectivity, repeatability, real-world validity, appropriate level of fidelity, and unmanned system/human agent independence and intuitiveness. This is not intended to be a comprehensive list of all possible metrics for human/autonomous system performance. Some more general background on metrics for Human-Robotic Interaction can be found in refs. 5-6. There are currently a number of programs examining human control of multiple unmanned systems at higher levels of autonomy. Each program has tended to develop their own approach towards metrics and test approaches as suited to their problem. Ultimately, it will be important that common sets of metrics are developed that can be utilized for these types of systems and support comparisons across different programs.

II. SYSTEM DESCRIPTION

This section will provide an overview of the Intelligent Autonomy architecture, the main components, and how an operator would interact with them. The different components are integrated via a publish/subscribe approach with a set of common Extensible Markup Language (XML) schema that allows components from different academic, industry, and government performers to interact. Additional details about the overall Intelligent Autonomy (IA) system can be found in refs. 1-2.

Initially, an operator would begin tasking the system by specifying high-level mission tasks, constraints, and priorities through a Mixed-Initiative Interaction Module (MIIM). There are several different human interface concepts that will be evaluated as part of the MIIM [4, 7-8]. Tasks range from simple data collection to more complex tasks such as searching or maintaining coverage over a region. There are a large number of constraints available including no-go zones, no communication zones, no surfacing zones for UUV's, and hard time and precedence constraints. Because many of the missions are fairly complex, the operator can choose to use a Case-Based Reasoning (CBR) component that identifies previous mission plans that could be used like a template as a starting point [9]. The operator defines high-level features of the mission and the CBR component ranks the past mission plans that are most relevant to that set of criteria. Alternatively, the operator can choose to specify the mission completely manually. The operator then goes through a risk management process to define what types of risks are acceptable for the system to take [10]. For each type of risk, the operator defines the severity of risk and possible risk mitigation approaches. Risk exposure is managed hierarchically. For example, the operator can specify that all systems should avoid a particular risk in general, but then also specify a particular vehicle or mission task for which it would be acceptable to be exposed to that risk. Next, the operator can specify the level of autonomy of the unmanned systems relative to various contingencies [11]. This can apply to all vehicles or to specific individual or classes of vehicles. The choice of levels includes fully autonomous dynamic replanning, management by exception with a customizable time delay, and management by consent.

After completing mission specification, the operator sends the mission specification to a multi-vehicle planning system. This will allocate tasks to vehicles for all tasks other than those that have already been designated to a particular vehicle by the operator. The planning system will order and schedule tasks for each vehicle and provide inputs to detailed route and payload planners that may vary by vehicle classes. In order to ensure that solutions are both computationally feasible and operationally acceptable, the planning problem is decomposed and constrained in different ways. Several different approaches towards optimization have been examined under the program including Mixed Integer Linear Programming, a market-based approach, and a Contract Net Protocol approach. Some mission plans will have secondary tasking that is not mandatory to complete. After completion of planning for the primary mission, the system will attempt to provide as much coverage as possible of secondary tasks without violating any of the constraints of the primary mission tasks. This is optimized using a receding horizon approach to support multi-vehicle coverage of an area with both path deconfliction and appropriate vehicle trajectories for sensing [12]. Individual vehicle plans are then combined into a single mission plan that is provided to the operator. The operator is provided with a variety of options to visualize and analyze the mission plan including geographic, timeline, task allocation, and risk assessment displays and animation of the mission [7-8, 10-11].

After the plan has been generated and the operator approves the mission, the autonomous systems begin execution. During the course of the mission the MIIM provides a variety of options for monitoring the mission including different team and individual vehicle mission visualization approaches and an alert management system [7-8,13]. While the interface is focused around a map-based display, there are also display options that show timelines, communication networks, vehicle status, and teaming issues. If necessary, the operator can work on developing new plans to re-task the system or modify the plan by tabbing to separate windows for new plan development. Some of the vehicles operate without communications with the operator for significant periods of time and have on-board mapping, sensor processing and sensor fusion capabilities that they can utilize to update their understanding of the environment and other entities [14]. This information is used as part of a replan assessement component on-board both the operator's control station and those vehicles that have a replanning capability. This determines if new data will impact on the vehicles capabilities or their ability to carry out the mission. If there is an impact, the replanning component alerts the operator or triggers a fully autonomous replan depending on the levels of autonomy specified by the operator and whether or not the vehicle is currently in communications. When not in contact, it will sometimes be necessary for the vehicle to make a decision about changing its mission plan without operator assistance based on rules of engagement that the operator has specified. This has included some examination of the ability of unmanned systems to reallocate tasks between vehicles fully autonomously in a way that is robust to communications limitations [15-16]. In some significant events, the system will provide the operator with choices about what the vehicle should do next in a mixed-initiative way, which may ultimately lead to a replan.

III. METRICS

Table 1 contains the list of candidate metrics that was compiled by a Human Factors Working Group (HFWG) under the Intelligent Autonomy program. The HFWG consists of government, academic and contractor personnel supporting the IA program who have an interest in the user interface design including engineers with experience in autonomous vehicles, human-factors engineers, and psychologists.

IV. OPERATOR EVALUATIONS

Several evaluations of different user interfaces and autonomous systems components have been conducted to elicit unmanned vehicle operator feedback on the systems, and to evaluate the human performance metrics from the IA Metrics Toolset [4]. Scenario-based user evaluations were employed as the experimental paradigm to identify, refine and validate IA metrics. During each evaluation, sets of proposed metrics were employed and evaluated. Once the data were collected and analyzed, the utility of the metrics was analyzed. An end state objective is to provide a toolset that can be useful on this and other similar programs. There was significant focus on three areas of human systems integration for the evaluations: usability, appropriateness of automation levels, and system/mental model compatibility. An assessment of the usability of the systems was accomplished through traditional heuristic evaluation techniques, including the use of Likert scales to rate various aspects of usability. The appropriateness of the levels and kinds of automation was assessed by collecting situational awareness scores from the operator during the simulation. Finally, the software interface was assessed to determine the correlation between the users' mental model and the design and functioning of the software using several mapping techniques.

The process for the evaluations consisted of the following steps:

- Signing of the informed consent and receiving the pre-briefing on evaluation
- Explanation of Intelligent Autonomy Program
- Human Factors Operator Workload Drivers Briefing
- Evaluation Guidelines
    i. Evaluation Method and Metrics
    ii. Heuristic Feedback
    iii. Graphical User Interface (GUI) Guidelines
- Demonstration of software package by contractor
    i. System Overview Brief
    ii. Simulation/Training
- Freeplay with software
- Performance with Scripted Scenarios Questionnaire completion
- NASA-TLX (task load index) completion
- Crew Debriefing

Three sets of active duty and retired Navy UAV and UUV operators worked in teams during the training and free play time. During the actual evaluation, operators worked alone, and were supervised by a member of the evaluation team. Operators were given the background scenario information and required to task the UxVs to accomplish various reconnaissance and surveillance tasks such as to search sections of a coastline. The evaluation team collected data during the evaluation, recorded observations, provided the Situation Awareness Global Assessment Technique (SAGAT) probes, recorded mental workload ratings, and answered any general questions for the operators

5. EVALUATION AND USE OF METRICS

The following describes some of the major metrics that were employed and evaluated for their utility during these evaluations. A brief description of the metric, intended implementation and utility is included below.

Table 1: Candidate Metrics for the IA Toolset.

Metric	**Performance Parameter**	**Human Factors Concern/Intent**
Cognitive workload	Modified Cooper-Harper scale rating	Subjective cognitive workload estimates
Command frequency	# Commands/event	Objective operator cognitive workload
Communication	# Interactions/agent/hour	Level of collaboration / communication efficiency
Decision accuracy	% Correct decisions	Human-system performance (Defining decision points and appropriateness of outcome will be difficult.)
Error complexity	# Steps used to correct error	Extent mission goals are diverted or derailed due to human error
Error frequency	# Errors/hour	Human-system performance
Error impact	Time to correct error	Extent mission goals are diverted or derailed due to human error
Error recovery	% Errors corrected	Human-system's ability to recover from human error
Planning efficiency	# Commands/event/time of event	How well does the operator's mental model of the planning tool match the actual algorithm?
Replanning	Time to resolve forced change	Human-system adaptability (i.e. ability to respond to system failure, environmental influence, new direction, etc.)
Reaction Time	Time to respond to stimulus	Saliency of important stimuli (ex. availability of target images)
Situational Awareness	% Accuracy as measured by SALIANT or SAGAT or SART scale rating	Ability for operator/team to understand and communicate past and present events or states and predict future ones. Is the information available for the operator to maintain 1)general SA and 2) state or modality of specific system elements (i.e. vehicle,
Mental Model	Correlation between system state and operator mental model	How well does the operator's temporal and spatial mental model of the current system state match reality?
Task time	Time to complete mission/task	Time to complete task can be used to measure various aspects of human-system performance (efficiency)
Automation adaptability	Time to complete mission/task	Function allocation of automation. Mission performance as influenced by level of autonomy (chosen or forced)
Temporal workload	Time to complete mission/task	Objective operator cognitive workload
Tasks accomplished	% Missions/tasks completed	Use of pre-defined relevant milestones necessary for completing a mission measure the productivity of the system through the quality of information and interfaces
Trust	Lee & Moray trust scale rating	Subjective measure of trust/faith, perceived predictability, and perceived dependability of the automated agent
Usability	Likert scale rating	Subjective ratings for comfort, ease of use, consistency, etc. (elements of usability)
Training	Time in training before achieving proficiency (as defined by another metric/combination of metrics, such as tasks accomplished and task time)	How much training is required to meet other human performance objectives?
System effectiveness	Probability of identifying or classifying target	Objective joint agent-agent measure of effectiveness
False alarm	# incorrect classifications/# classifications	Objective joint agent-agent measure of effectiveness

Planning Time

- Metric Definition: The degree to which the automated planning tool supports the planning process as a measure of time.
- Performance Parameter: Time from presentation of mission goals to start of execution of mission.
- Utility: Response times are generally most useful as a baseline for comparisons of multiple systems, or assessment of design enhancements. Response times can also be useful if the evaluator has knowledge of time allotted to perform a task during actual operations. For example, if it takes x minutes to replan a route with system A, and the naval system requirement is to be able to replan a route in y minutes, then the data could be used to determine whether system A is in compliance with Navy requirements. Therefore, the use of response time measures requires extensive knowledge regarding the task and the operational requirements, typically based on interviews with subject matter experts and review of operational concept documentation.

Task Time

- Metric Definition: Time to complete task can be used to measure various aspects of human-system performance (efficiency).
- Performance Parameter: Time to complete the mission, task, or task segment. Task time for three events was recorded: defining the mission (planning time), re-plan by inserting a constraint such as a no-fly zone and re-plan by prosecuting a target.
- Utility: Task reaction time events measured are generally not very useful in isolation, but can be useful for comparisons relative to other similar designs or to measure improvements. However, it is important to ensure that this metric is applied to time critical tasks.

Situation Awareness

- Metric Definition: Ability for operator/team to understand and communicate past and present events or states and predict future ones. This can address if the information is available for the operator to maintain both general SA and the state or modality of specific system elements (e.g., vehicle, sensors, environment, battlespace, mission, level of autonomy, decision framework).
- Performance Parameters:
    - The Situation Awareness Global Assessment Technique (SAGAT) provides an objective measure of situation awareness by directly comparing operators' reported SA to reality. With this technique, a human-in-the-loop simulation is frozen at randomly selected times, the simulation is suspended, and the system displays are blanked while the operators quickly answers questions about their current understanding of the situation. Operators' perceptions are then compared to the real situation (based on information drawn from the computer or from subject matter experts who answer the SAGAT queries while looking at the displays). Comparing the data in this manner provides an objective, unbiased assessment of SA [17].
- Utility: Overall, the SAGAT measure appears to provide useful information regarding operator situation awareness, and the scores tend to correlate with the operators' and human factors analysts' opinions regarding the display. SAGAT data will only be useful, though, if appropriate probes are generated. When using SAGAT, the HSI analyst must develop a thorough understanding of the task and the information the operator requires to accomplish that task. This will likely require the evaluator to perform a task or work analysis, or at least conduct extensive interviews with subject matter experts. Therefore, the use of the SAGAT requires intimate knowledge of the operator's task and may be cumbersome to implement. Furthermore, once an HSI analyst has conducted the task analysis, they can recommend design changes for a system prior to conducting a human-in-the-loop evaluation.

Workload

- Metric Definition: Subjective workload estimates.
- Performance Parameters:
    - An automated version of the NASA-TLX was used to measure workload. NASA-TLX is a subjective workload assessment tool. NASA-TLX is a multi-dimensional rating procedure that derives an overall workload score based on a weighted average of ratings on six subscales. These subscales include Mental Demands, Physical Demands, Temporal Demands, Own Performance, Effort and Frustration. It can be used to assess workload in various human-machine environments such as aircraft cockpits; command, control, and communication (C3) workstations; supervisory and process control environments; simulations and laboratory tests [18].
    - Workload ratings were also obtained coincident with the administration of each set

of SAGAT probes through use of a modified Cooper-Harper Workload rating scale.

- Utility: NASA-TLX is a multi-dimensional measurement tool and includes six subscales to provide a more detailed analysis of the operator's workload. Those subscales analyses were not conducted for this evaluation, but could be examined if more detailed information regarding the source of the workload is required. The Modified Cooper-Harper Workload Rating Scale is a flow chart that leads the operator through a series of questions that result in a numeric rating of the workload associated with the task. Although the scale is one-dimensional, the outcome may be more objective and has a quantifiable meaning (e.g., "3" indicates "fair, some mildly unpleasant deficiencies - Minimal sailor effort required for desired performance"). For the purposes of this type of operator evaluation, it may be better to use a measure such as the Modified Cooper-Harper which provides a more general measure workload, but allows the evaluator to know exactly what the operator meant by the score. A comparison of the Modified Cooper-Harper scale vs. NASA-TLX workload metrics was conducted during a recent evaluation. As expected, similar ratings were obtained between the two methods.

Usability

- Metric Definition: Subjective ratings for comfort, ease of use, consistency, etc. (elements of usability).
- Performance Parameters:
  - A Likert scale was used to assess various software system capabilities and aspects of the user interface.
  - Open ended questions were presented to the operators.
  - Usability was also measured using the System Usability Scale (SUS)
- Utility: The System Usability Scale literature advises not to use individual question scores in the analysis, but to only use the composite score. There is limited meaning to the composite score since it does not identify the factors that were positive or negative in determining the score. In addition to the SUS, we employed an additional questionnaire to address areas of specific interest. Subjective questionnaires provide some of the most useful information in the assessment of a user interface. This method of data collection allows the operator to focus their comments on the areas of most importance to them. Furthermore, it provides the opportunity for the operators to make recommendations and suggest enhancements, which is the ultimate goal of conducting an evaluation. However, there were some situations in which operators would focus on relatively minor aspects of the interface that could be easily changed as opposed to the more advanced technologies that were being studied.

Mental Model Mapping

- Metric Definition: The degree to which the operator's temporal and spatial mental model of the current system state matches reality.
- Performance Parameters:
- A number of techniques can be employed that attempt to correlate between system state and operator mental model.
  - Mental model mapping required operators to recreate a visual picture of the display when the screen was blanked for the SAGAT probes.
  - A second approach presented screen shots to the operators and asked them to label icons and explain the purpose of various features identified on the screen shots.
- Utility: The mental model mapping was difficult to score and did not seem to provide a good indication of the operator's understanding of the system state. The second approach provided more detail regarding the operator's understanding of the features on the display, but did not successfully address the underlying degree of understanding regarding the system state.

6. CONCLUSIONS

The metrics described in this paper have been useful in understanding important aspects of the cooperation and coordination of human/system interaction and collaboration. However, additional work will be required to examine other metrics and continue refining the implementation of the ones already employed. SAGAT has been relatively successful as a measure of situation awareness, but the probes will need to continue to be refined to improve relevance to the operational tasks. Both NASA TLX and Cooper-Harper Rating (CHR) were also relatively successful as a measure of workload. While TLX provides a broader workload scope, Cooper-Harper provides faster and more specific design feedback, such as defining the severity of the workload problem. The use of task times and response times to compare systems or versions of systems was also helpful, but could be improved with future software versions including embedded recording of times within the software packages. Additional refinement of the time metrics will require refining the tasks selected for measurement to ensure that they are operationally relevant, and that the operators are aware that time to complete is an important aspect of that task. Some areas that require more significant metric development are measures of the operator's mental model and trust of the autonomous system. To date, the mental model metrics have focused on the use of the map to measure the operator’s spatial mental model of the task environment and understanding of interface features. Another possible approach is to explore the temporal aspects of the task perhaps by requesting that the operators complete a simplified version of a timeline display or to test the operator on their understanding of how the autonomy will react in specific contingencies.

ACKNOWLEDGEMENTS

The authors would like to acknowledge all of the intelligent autonomy team members who helped develop and apply the set of intelligent autonomy metrics including Cris Cristina, Naval Surface Warfare Center, Missy Cummings, MIT, Karen Harper, CRA, John Lee, University of Iowa, Mike Linegang, formerly of Aptima, Vera Zaychik Moffitt, Lockheed-Martin ATL, Danielle Moulds, NAVAIR, Tom Moulds, DTI, and Norman Warner, NAVAIR. The work described in this paper was funded by the Office of Naval Research.

REFERENCES

[1] Steinberg, M., “"Human Interface and Autonomy Technologies for Multiple Unmanned Air, Sea, and Undersea Vehicles," in proceedings of AUVSI Unmanned Systems North America, 2007, to appear.

[2] Steinberg, M. “Intelligent Autonomy for Unmanned Naval Vehicles,” in proceedings of the SPIE Unmanned Systems Technology Conference, 2006.

[3] Mitchell, P., Cummings, M., Sheridan, T., "Human Supervisory Control Issues in Network Centric Warfare" (HAL2004-01) ,September 2004.

[4] Billman, L., Cristina, C., and Balmer, D., "Operator Feedback On The Lockheed Martin Intelligent Control And Autonomous Replanning Of Unmanned Systems (ICARUS) And Draper Labs/Charles River Risk-Aware Mixed-Initiative Dynamic Replanning (RMDR),″ CSS/TR-06/XX, to appear.

[5] Steinfeld, A., Fong, Kaber, T., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M."Common Metrics for Human-Robot Interaction," in proceedings of 2006 Human-Robot Interaction Conference, ACM, March, 2006.

[6] Goodrich, M., Olsen, D. R.: “Seven Principles of Efficient Human-Robot Interaction” in proceedings of International Conference on Systems, Man, and Cybernetics, IEEE, 2003.

[7] Kilgore, R., Harper, K., Cummings, M., and Nehme, C., "Mission Planning and Monitoring for Heterogeneous Unmanned Vehicle Teams: A Human-Centered Perspective" in proceedings of the AIAA Infotech @Aerospace 2007 Conference, 2007, to appear.

[8] Lingang, M., Stoner, H., Patterson, M., Seppelt, B., Hoffman, J., Crittendon, Z., Lee, J.,“Human-Automation Collaboration in Dynamic Mission Planning: A Challenge Requiring an Ecological Approach,” Human Factors and Ergonomics Society Annual Meeting, 2006.

[9] Ulam, P, Endo, Y., Wagner, A., Arkin, R.C., “Integrated Mission Specification and Task Allocation for Robot Teams - Design and Implementation", in proceedings of ICRA, 2007.

[10] Page, L., Nervegna, M., DiBiaso, D., “Unmanned System Risk Management,” in proceedings of AUVSI Unmanned Systems North America, 2007.

[11] Nervegna, M., Ricard, M., “Risk-Aware Mixed-Initiative Dynamic Replanning (RMDR) Program Update,” in proceedings of AUVSI Unmanned Systems North America, 2006.

[12] Ahmadzadeh, A., Buchman, G., Cheng, P., Jadbabaie, A., Keller, J., Kumar, V., Pappas, G., “Cooperative Control of UAVs for Search and Coverage,” in proceedings of AUVSI Unmanned Systems North America, 2007.

[13] J. Franke, V. Zaychik, T. Spura, E. Accettullo, “Inverting the Operator/Vehicle Ratio: Approaches to Next Generation UAV Command and Control,” in proceedings of AUVSI Unmanned Systems North America, 2005.

[14] Snyder F., Morris D., Haley P., Collins R., “Maritime Image Understanding” in proceedings of SPIE Philadelphia Conference, 2004.

[15] M. Godwin, S. Spry, Hedrick, K., "Distributed Collaboration with Limited Communication using Mission State Estimates," in proceedings of the American Control *Conference*, Minneapolis, USA, June 2006.

[16] Ryan, A. et al "Decentralized Control of Unmanned Aerial Vehicle Sensing Missions," in proceedings of the American Controls Conference, 2007.

[17] Endsley, M., "Measurement of Situation Awareness in Dynamic Systems." *Human Factors*," 37(1), 65-84, 1995.

[18] Hart, SG, and Staveland, LE, "Development of the NASA-TLX (task load index): Results of empirical and theoretical research,"1988.

Concepts of Operations for Robot-Assisted Emergency Response and Implications for Human-robot Interaction

Jean Scholtz[1]
Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352
jean.scholtz@pnl.gov

Brian Antonishek, Brian Stanton, and Craig Schlenoff
National Institute of Standards and Technology (NIST)
100 Bureau Drive
Gaithersburg, MD, 20899, USA
{brian.antonishek, brian.stanton, craig.schlenoff}@nist.gov

Abstract—In this paper we discuss a field study at Disaster City, Texas in March 2006. First Responders and robot developers tried out various concepts of operations in a number of disaster scenarios. Observations, video data, and questionnaire data were analyzed and based on these results, we propose some guidelines as well as some future research areas for human-robot interaction. In addition to the guidelines proposed as a result of our observations in this study, we include design implications from other literature, both laboratory and field studies.

Keywords: Human-robot interaction, rescue robots.

I. INTRODUCTION

The exercise at Disaster City is one of a series in a National Institute of Standards and Technology (NIST) program sponsored by the Department of Homeland Security (DHS). The goal of this program is to develop metrics and evaluation methodologies for Urban Search and Rescue (USAR) robots. In initial workshops with the first responder community, NIST developed a number of requirements for USAR robots [http://www.isd.mel.nist.gov/US&R_Robot_Standards/ accessed August 31, 2006]. These requirements were prioritized and[1] several work items are now being developed with the ASTM standards group for emergency response [ASTM E.54.08, http://www.astm.org accessed August 31, 2006]. In order to refine the requirements initially developed, NIST is running a number of "responder meets robots" exercises.

II. DISASTER CITY

Disaster City is a Texas Task Force One (TX-TF1) training facility located at Texas A&M University, College Station, Texas [http://www.teex.com/teex.cfm?pageid=USARprog&area=USAR&templateid=1117 accessed August 31, 2006]. It is part of the Texas Engineering Extension Service (TEEX) at Texas A&M. The TX-TF1 training site features full-sized collapsible structures, including a strip mall, office building, industrial complex, assembly hall/theater, single family dwelling, train derailments, and three rubble piles.

The event took place over three days. There were "scenarios" scheduled for 4- three hour blocks. These scenarios were used to familiarize the responders with the capabilities of the various robots[2]. Scenarios took place on two rubble piles, in

[1] This research was conducted while Dr. Scholtz was at NIST.

[2] Vendors supplied robots for the technology exercise. The mention of these robots in this paper does not constitute an endorsement by the National Institute of Standards and Technology. The robots are described only to help readers understand the capabilities of the different robots.

the strip mall, on the passenger and hazmat trains, in the collapsed house, and in the single family dwelling.

The final three hour block of time was used as a mock incident response. First Responder teams were assigned to one of four scenarios: single family dwelling, collapsed house, passenger train, and rubble pile. In the Data Analysis Section, we explain how these were selected.

Figures 1 – 4 show each of the venues. In addition, a brief description of each type of disaster is given.

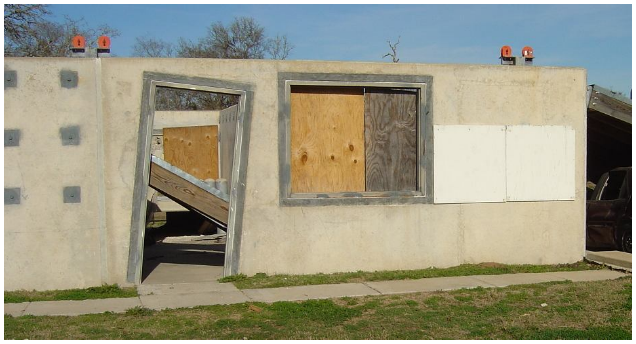


Figure 1. Single family dwelling.

The single family dwelling is partially collapsed due to an earthquake. The main entrances are compromised. Responders must enter through either a leaning collapse or through a 24” triangle breach. There is also a basement that is accessible from the outside down some steep stairs. The maze of rooms needs to be mapped and searched for victims.

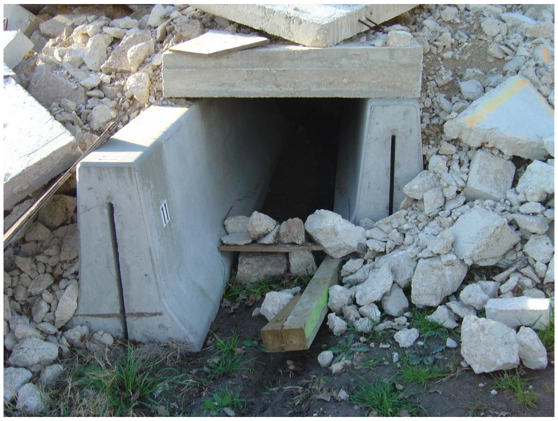


Figure 2. Rubble Pile.

The rubble pile is a fully collapsed structure with subterranean voids. There are some entrances supported loosely by concrete barriers. There are confined dimensions and problematic rubble that will hamper searching.

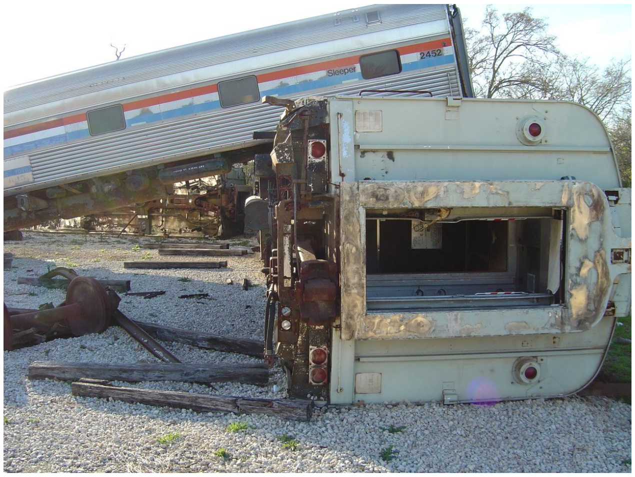


Figure 3. The Passenger Train.

The passenger train was hit by the industrial hazmat tanker cars. The sleeper car is evaluated and has curtained alcoves on each side of a narrow aisle that should be searched. The crew car is lying on its side and also needs to be search. The mailroom in this car needs to be searched but is too small for a responder in a level A suit to enter.

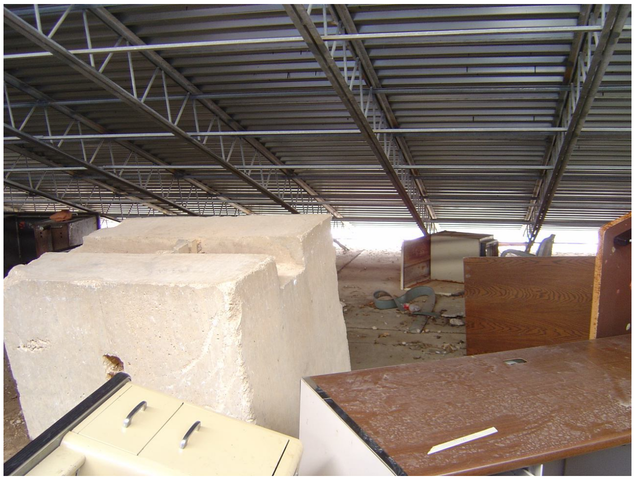


Figure 4. The House of Pancakes viewed from inside.

The house of pancakes is a partially collapsed building with the roof almost in contact with the ground on the only accessible side. Robots must enter through the confined space under the metal roof or through a breach. There is a maze of obstacles and debris which will hamper search.

III. ROBOTS

We used both air and ground robots in the initial scenarios. However, because of safety concerns, the grounds had to be cleared when using the aerial vehicles so they were not incorporated into the final mock incident responses. A number of diverse ground robots were used. These included robots with manipulators, extreme mobility robots, and robots that could be thrown or otherwise launched into an area the responders needed to investigate. Some robots had wheels while others had treads. Some robots had the ability to change shape (See figures 5 a and b). Figures 6a-6e show the diversity of ground robots. The robots used in the scenarios were all teleoperated. One constraint in selecting robots for various scenarios was that the bandwidth they operated on had to be compatible. Of course, this was in addition to the physical constraints imposed by the scenario.

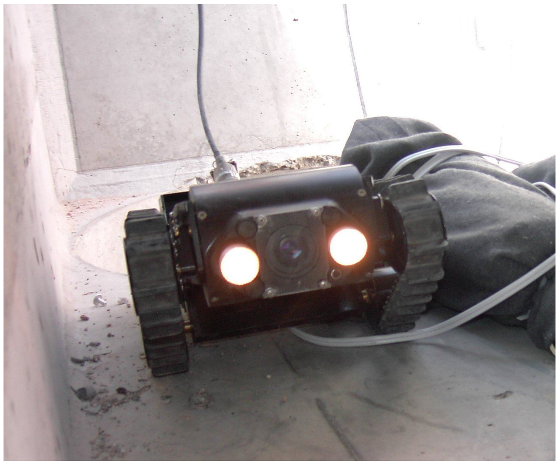


Figure 5a. Shape-shifting robot in lower configuration.

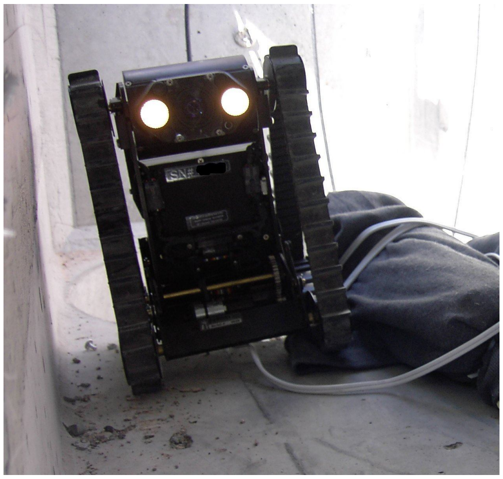


Figure 5b. Shape-shifting robot in raised configuration.

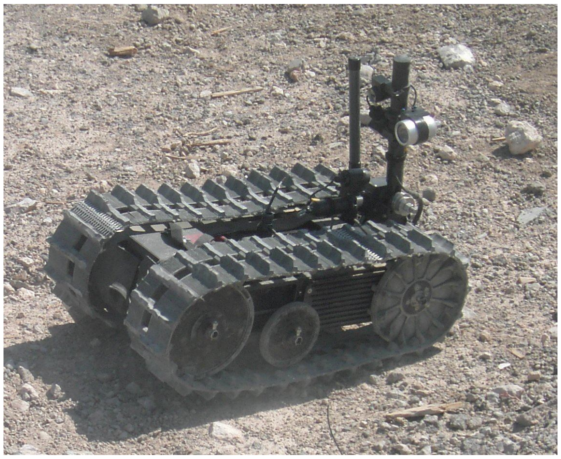


Figure 6a. A robot which navigates using tracks.

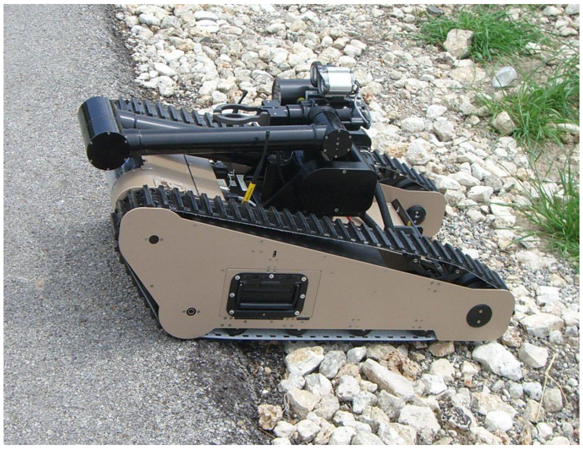


Figure 6b. A wedge-shaped track robot with manipulator.

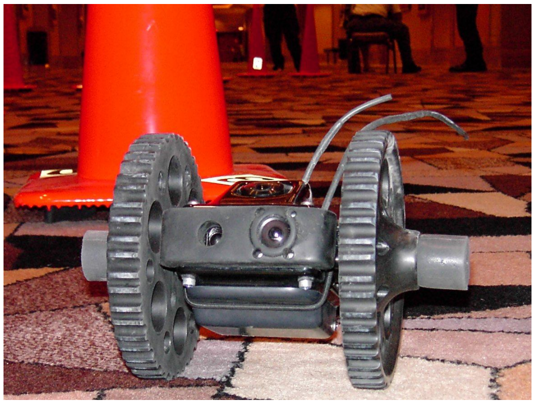


Figure 6c. A small "throwable" robot.

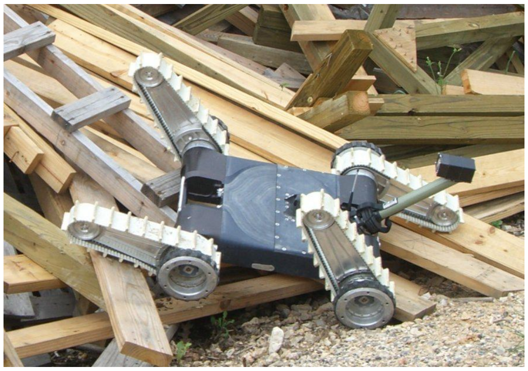


Figure 6d. A robot with articulators.

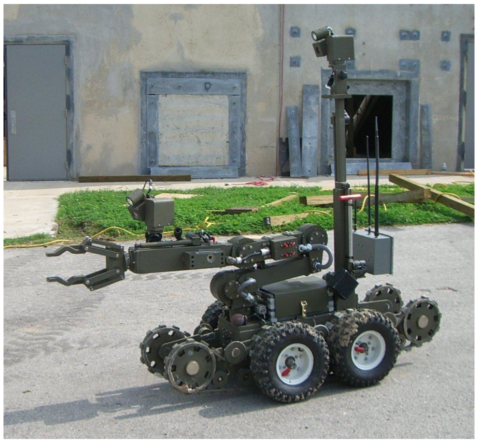


Figure 6e. A wheeled robot with articulators.

IV. DATA COLLECTION AND ANALYSIS

For each scenario NIST personnel took video data and made observations. In addition, we collected questionnaires from the responders concerning the representativeness of the scenario and the team performance. Figure 7 shows the questionnaire used. Responders were asked to rate each question on a scale of 1 to 7, where 1 was the low end of the scale and 7 was the high end. In general responders gave different ratings to different robots (if there were multiple ones involved in the scenario) for questions 4 and 5.

1. **How representative was the scenario of a possible US&R event?**
2. **Concept of operations used in scenario?**
3. **Assessment of responder team performance**
4. **Capabilities of robot**
5. **Utility of robot in scenario**
6. **Length of time needed to accomplish the scenario**
7. **Overall performance of scenario (responders and robot)**

Figure 7. Questionnaire used to assess the different venues during the first three days

These questionnaires were collected from each member of a responder team during the first three blocks of the exercise.

Table 1 shows the results from these questionnaires

	Passenger Train	**Rubble Pile**	**Strip Mall**	**Train**	**Dwelling**	**Rubble Wood Pile**	**Pancakes**
Representative Scenario	5.38	5.69	5	6	5.5	5.5	5.86
Representative Operations	5.29	6	4.58	5.38	5	5	6
Team Performance	4.67	4.4	5.14	5.25	4.75	4.67	5.17
Bot Capabilities	4.13	3.19	5	5.5	4	5	4.5
Scenario Utility	3.29	3.5	4.86	5.75	4	5	5
Time Required	3.86		4.14	4.43	5.75	4.75	5.5
Robot/Responder Performance	4	3.93	5.29	4.1	3.75	5	4.92
Operator Interface	3.67	4.67	4.57	5.75	5.67	5	5

The scenarios and the operations performed were rated as 5 or over with the exception of the strip mall. That venue was not used in our final portion of the exercise. The Hazmat train was not used as well as that required the use of an aerial vehicle. The four venues selected for use in the final portion of the exercise were the passenger train, the rubble pile, the dwelling and the house of pancakes. All of these were highly rated as representative of situations responders would encounter.

The robot/ responder performance and the operator interfaces for the robots were not as highly rated. There are several reasons for this. First of all, in many cases, the operator interface needs to be improved. One goal of this analysis is to examine the operator interface, not just for usability, but in the concept of operations. The performance of the robots and responders is also due to differences in expectations of responders and the actual capabilities of the robots. Again, using the robots and developing concepts of operations based on a better understanding of capabilities is essential to improving the robot/responder team performance.

V. CONCEPTS OF OPERATIONS

The most interesting data came from observations of emerging concepts of operations from the various venues. We describe these four scenarios in the following paragraphs.

A. Single Family Dwelling

The responder team used three robots primarily in this situation. They used a large robot with a manipulator arm, which we designate as robot A for this document, a smaller shape changing robot, which we designate as robot B, and for a portion of the time they employed a small throwable robot which we designate as robot C. The responders were setup in a tented area with power supplied by generators in front of the single family dwelling as there were no Hazmat concerns. In addition to the three robots, a search dog was also used. The robots were operated by the robot developers under the direction of the responders.

The team leader had the operator of the large robot drive the robot around the building. He watched the video and constructed a map of the exterior of the dwelling based on this information (Figure 8). This also allowed him to determine the entrances to the dwelling. After the exterior had been traversed, the team leader sent the larger robot into the dwelling through the partially collapsed entrance. The smaller, shape changing robot was sent into the basement of the dwelling using the stairs. The two operators were sitting close to each other under the tented area with the team leader watching the video from both. He used this to map out the inside area and to determine that the area was safe enough to send in a dog. A possible victim was identified by the smaller robot in the basement. A dog was sent in to verify this.

There was an issue when robot A was unable to get into a suspected space in the upper floor of the building. According to the map the responder constructed there was an additional space that had not yet been searched. However, there were obstacles (collapsed walls and debris) that prevented the robot from entering this space. Both robot A and B were moved out of the dwelling and the larger robot, robot A, used the manipulator arm to grip the smaller robot, robot B, and move it into the building, assisted by members of the response team. Once it had moved back into the area, the operator was able to place robot B on top of the collapsed wall which allowed the robot B to penetrate farther into the building. In this operation, the two operators moved close together and used cameras from both robots to do the placement.

Several other cooperative efforts were seen. In one instance, robot A dropped robot C through a hole in the main floor. The operator of robot C used both his camera and the camera of the robot A to maneuver through the basement area.

B. Rubble Pile

During the rubble pile scenario a responder operating a larger robot, which we designate as robot D, worked in conjunction with the rescue dog handlers. Using robot D, the responder circumnavigated the rubble pile, accessing possible entry points. The responder identified the existence of a victim using the microphone on the robot. The robot was also equipped with a speaker so the responder and the victim could communicate. This communication enabled the responder to narrow the search area by asking the victim if they could "see the robot". When the victim responded that the robot was in view, the dog handler then sent in the rescue dog to pinpoint the victim's location. Figure 9 shows what the rubble pile looked like.

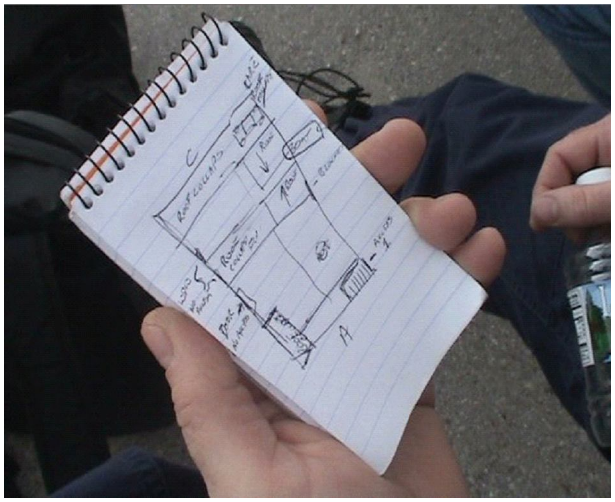


Figure 8. The map created by the responder.

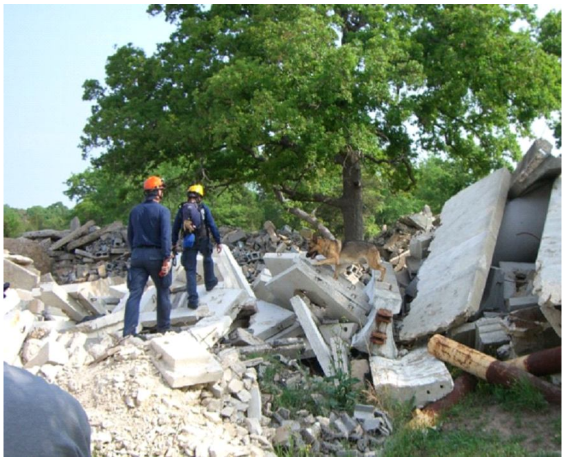


Figure 9. Responders searching the rubble pile

C. Passenger Train Wreck

Two robots were used in this scenario. Each robot was run by an operator under the direction of a First Responder. The responder asked the robot operators to clear the train and look for any signs of life on the slanted wrecked train. The first robot, which we designate as robot E, started at the entrance on the ground and began to clear the train looking for survivors / victims. The other robot, which we designate as robot F, started at the back of an upended train car and worked its way toward the front. A responder dropped robot F in a side window and stayed there to do tether management. The operators were located outside of separate sections of the trains and communicated over the hand-held radios to each other. Figures 10 and 11 show two setups at the passenger train.

Robot F eventually got a piece of cloth wrapped around a tread and was stuck. The operators decided to use robot E's arm/claw to grab and try to remove the cloth from the robot F's tread. Robot E's operator managed to grab the cloth with its manipulator but was unable to free the cloth and instead, dragged robot F a short distance. A second strategy was developed in which robot E stayed stationary and operator for robot F attempted to drive away from robot E to free the cloth. This strategy was successful.

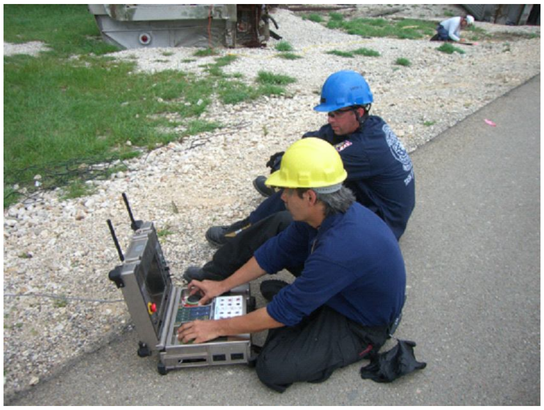


Figure 10. Responders find a place to setup the OCU to search the passenger train

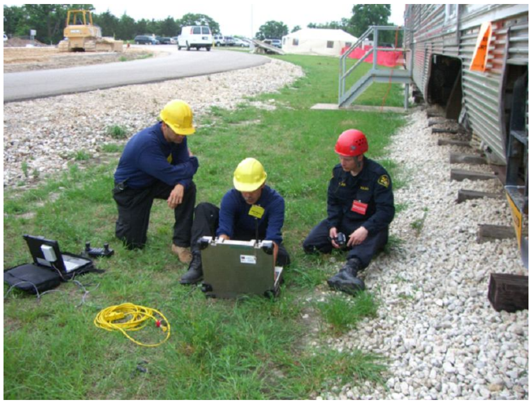


Figure 11. Another group of responders setting up to search the passenger train

In the second part of this scenario, the team was searching a train car that was lying on its side. The same two robots were used, again being driven by their operators under supervision of the First Responder. The responder asked to have them clear the train from opposite ends. This time the operators were set up next to one another. The robots eventually met up with each other in the center of the dark train and used each other's lighting to help see a larger area than they would have been able to see by themselves.

There was another interesting operator event at the trains. The operator for robot E was quite tired after concentrating so heavily and another operator offered to replace him. While turning over control, the original operator gave a verbal description of where he thought the robot was currently positioned in the train and drew an imaginary path on the operator control unit (OCU) using his finger to describe the center hall layout.

D. House of Pancakes

The House of Pancakes scenario focused heavily on three robots, which we designate as robots G, H, I, in conjunction with a rescue dog handler. In the scenario, the House of Pancakes was meant to represent a recently collapsed building. The scenario started with the responders tele-operating robot H around the outside of the house to look for the presence of survivors and to determine the best opening to enter the house. An open doorway was found and robot H was navigated through that doorway. Robot H was assumed to have biohazard sensors on it that could detect hazardous gases in the environment. Once robot H traversed all accessible areas of the house, robot H (conceptually) responded that the environment was safe, the rescue dogs entered the site to smell for survivors. In the scenario, there was one survivor near the back of the house which the dog quickly detected.

In parallel with this, robot G, with robot J in its grippers, was tele-operated to drive up on the collapsed roof of the house. Figure 12 shows robot G carrying robot J. The purpose of this part of the scenario was to have robot G drop robot J into a breach near the uppermost portion of the roof to allow it to look around the remaining upper stories of the building to see if any survivors could be detected.

A small piece of plywood (about 1 m by 1 m) was placed near the bottom of the collapsed roof to allow robot G to drive up onto the roof. Once robot G drove up on the plywood and reached the uppermost portion of the roof, the operator aligned robot G with the breach and extended its manipulator to be directly over the breach. The pincher in the manipulator was then released and robot J was dropped into the breach. For this scenario, robot J was not functional (it broke earlier in the week), so the scenario ended here. If robot J was functional, it would have been used to navigate around the upper stories of the building to find survivors.

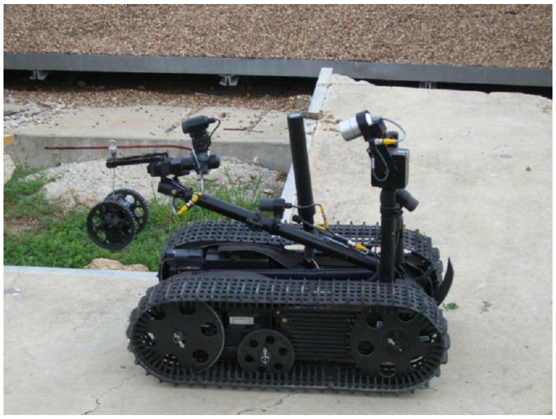


Figure 12. Robot G with robot J in its gripper.

VI. IMPLICATIONS FOR DESIGN

Based on the emerging concept of operations we can determine some priorities for design – and likewise we can also determine some items that are not as likely to affect design.

In the single family dwelling we did not find a need for operators to be on the move. Moreover, since there was no Hazmat danger, there was not a need for operators to wear protective gear. However, in the train scenario we found that the operators worked outside, sitting on the ground. Therefore, things such as lighting conditions played a big part in being able to see the OCU. Moreover, being able to comfortably set up operations in less than ideal conditions has to be considered when designing the OCU hardware.

The team lead was busy trying to update a map sketched on his field notebook with information given him by the two operators of the robots. A shared electronic notebook might be a good addition when working with teams of robots. Assume that the team lead could sketch in the initial external map as the perimeter is being mapped out. If this were done on a tablet PC, for example, and then used as a shared file both robot operators could add information to it as they searched the building. The team lead could have access to this on the tablet PC and could add information and annotations as well. The notion of maps surfaces again in the train scenario when operators change shifts. Having explicit information for the incoming operator to understand where the robot is and what has already been searched is valuable.

The use of videos from two robots when doing a cooperative task was accomplished by having the operators sit close and leaning over to see the other's OCU. While there is a need to have hardened cases for the OCU, it might be feasible to have hardened display units that could be attached to several OCUs if it is feasible that robots might cooperate. Then the video from one robot could be broadcast to several additional display units. For example, when one operator is picking up or setting down a smaller robot, both operators need to have a good view of what is happening so that the smaller robot can be correctly placed and can move as necessary to enter a void or start up a steep slope. In these scenarios responders positioned the smaller robot in the grippers of the larger robot outside of the buildings. This might not always be the case so it is essential to provide good video to the operators to position both robots to ensure that the smaller robot is not damaged during this operation. Releasing the smaller robot was a delicate operation in many cases. In the case of the robot J, this was not an issue. But in the single family dwelling, for example, it was necessary to place the smaller robot on a rather steep incline. Therefore the smaller robot had to be position so that it could immediately start moving up the incline rather than sliding backwards. This necessitated ensuring the camera view was on the smaller robot while releasing the grippers. The operators had to be closely coordinated to carryout their actions (releasing and starting to move the smaller robot) at the same time.

The team also made use of sharing resources of the robots. In the train scenario using two lights (one on each robot), rather than just a single light helped to speed the search of the train.

Communications need to be provided. In the scenarios we saw communications between robot operators, between robot operators and victims, and between responders and the robot operators. Teams communicated to share robots. Granted that this was due to limitations of the number of robots available but we assume that this will most likely be the case in the future. This would allow teams to know what robots are available should they find a need for a particular capability. In the House of Pancakes and in the Single Family dwelling we saw responders use two robots in parallel. There is a need for communications between the responders in these two efforts. As the goal is to quickly locate victims and to determine how much of the site has been covered, a way to fuse information coming back from both efforts should be provided.

We did not simulate a command and control center in this exercise. This would add another level of communications. Some questions would be whether the raw data such as video footage or sensor data would be available directly to the command and control center on demand. Would it be sufficient to have a dynamically updated map showing where teams are working and where the robotic resources are? Assuming that multiple robots are being used in a scenario, what type of fusion of information should be done and transmitted to command and control? Would it be sufficient to know which robots were currently in use?

A number of awareness issues should also be considered [4]. Knowing which teams are using which robots at any point in time is essential both for command and control and for the responder teams. Responder teams might want to know if the robots being used are "on task", that is actually searching or if there is some sort of robot help situation in progress.

VII. DESIGN IMPLICATIONS FROM THE LITERATURE

Murphy and her team have done much work on field studies that can be added to this analysis. For example, Burke et al. found that a good percentage of operators' time in US&R missions was consumed with gathering information about the state of the robot and that state of the environment [2]. This time was significantly greater than the time they spent navigating. They also found that operators had difficulty incorporating their small view (through the robot camera) into the overall picture. Displaying dynamically constructed maps of the overall area and the actual search areas of the various teams might help with overall situation awareness.

Burke and Murphy found similar issues in another field study when over 50% of the robot operator communications dealt with situation awareness concerns [1].

Murphy also found that two humans working together are nine times more likely to find a victim than one operator alone. This was not directly incorporated into our scenarios especially when there was more than one robot involved. The First Responder moved between robots and did look at the video but there as not a concern attempt to dedicate another responder to watching the robot video. If there are multiple robots involved, must a dedicated responder watch the video sent back from each robot? Or would it be feasible for a responder to watch video from several robots, assuming it could be viewed on a single display [5].

Drury et al. formulated a framework for awareness in human-robot interactions [4]. As noted in this framework there is a need for human-human awareness, robot-human awareness, robot-robot awareness, and humans' overall mission awareness. As the robots in our field study were tele-operated we did not see instances of robot-human awareness. The robot-robot awareness was also mediated by the human operators due to tele-operation control.

Yanco et al. studied awareness issues in USAR contests [6]. In this environment they were able to identify issues with the operator control unit, such as having to fuse information from multiple windows and lacking information about the area directly around the robot. [3] contains guidelines for presentation of information to the operator. While the contests are good tests of individual robot capabilities, there is no notion of a concept of operations.

VIII. CONCLUSIONS

We have described a multi-day field exercise culminating in an opportunity for responders to respond to a mock incident. In doing this, they selected robots appropriate for the venue and a concept of operations evolved. We observed the mock incident responses and noted how the robots, robot operators, and responders interacted. From this we were able to identify a number of issues that should be considered for human-robot interaction design. Some of the issues identified apply to individual robot OCUs. Other issues are concerned with the fusion of information to provide an overall assessment to the commanders.

These designs will need to be tested in the laboratory for effectiveness and usability but testing them in field exercises is essential to identify design requirements at a higher level. It is also interesting to compare these evolving concepts of operation to task analyses to determine if and how the strategies used by responders change as new technology is placed in use [7].

As a final note observations here led to discussion with some of the responders concerning metrics for evaluating the effectiveness of human-robot teams. Responders are concerned with how much of the disaster area is covered in how much time. Robots can contribute to this by coverage a good portion of this without putting the responders at risk. A proposed metric to use for judging the effectiveness of teams of humans and robots would be the amount of coverage/ time accomplished with only a robot. This addresses both the effectiveness and efficiency of the team along with the objective of minimizing the time responders are at risk.

IX. REFERENCES

[1] Burke, J.L, and Murphy, R. R. (2004). Situation Awareness and Task Performance in Robot-Assisted Technical Search: Bujold Goes to Bridgeport (No. CRASAR-TR2004-23). Tampa, FL:Center for Robot-Assisted Search and Rescue.

[2] Burke, J., L., Murphy, R.R., Coovert, M.D., and Riddle, D. L. (2004) Moonlight in Miami: A field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. Human-Computer Interaction, 19 (1-2), 85-116.

[3] Drury,J., L., Hestand, D., Yanco, H.A., and Scholtz, J. (2004). Design Guidelines for Improved Human Robot Interaction, CHI 2004 Poster

[4] Drury, J.L, Scholtz, J.C and Yanco, H. A.(2003). Awareness in Human-Robot Interactions. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Washington, DC, October 2003.

[5] Murphy, R.R. & Burke, J.L. Up from the rubble: Lessons learned about human-robot interaction from search and rescue. Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society, Orlando, FL, September 2005

[6] Yanco, H.A. , Drury, J.L., and Scholtz, J.C. (2004) Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition. Human-Computer Interaction, 2004

[7] Burke, J.L. RSVP: An investigation of the effects of remote shared visual presence on team process and performance in urban search & rescue teams Dept. of Psychology, University of South Florida, Tampa, FL, 2006, 140.

Multimodal Displays to Enhance Human Robot Interaction On-the-Move

Ellen Haas, Ph.D.
U.S. Army Research Laboratory
APG, Maryland,USA
ehaas@arl.army.mil

Chris Stachowiak
U.S. Army Research Laboratory
APG, Maryland, USA
cstachow@arl.army.mil

Abstract— The U.S. Army is exploring the use of advanced technologies such as tactile and spatial (3-D) audio displays to enhance Soldier performance in human-robot interaction (HRI) tasks. A field study was conducted at the U.S. Army Research Laboratory (ARL) in 2006 to determine the extent to which the integration of spatial auditory and tactile displays affects soldier situation awareness in a simulated UV HRI target search task performed in a moving HMMWV. Participants were 12 civilian males ranging in age from 18 to 46 years, with a mean age of 32 years. Participants performed a target search task, in which they searched for one target symbol among 50 non-target symbols displayed on an 18-inch diagonal computer monitor (a 30° field of view (FOV) visual display). Participants received audio and tactile cues to indicate on which third of a computer screen the target symbol was located. The independent variables were display modality, signal azimuth, participant age, and HMMWV movement condition. Display modalities were visual displays with supplemental cues in three display modalities; spatial audio, tactile, and combined spatial audio + tactile. The dependent variables were participant response time and accuracy, as well as the participant's subjective workload rating of display modality effectiveness. Accuracy data indicated that participants located over 99% of the targets correctly. Display modality was significant in terms of participant workload ratings, but was not significant for response time. Response time data indicated that no one display modality provided the shortest response time to all age groups, for all terrains. Workload with auditory + tactile displays was rated lowest of the three display modalities, which may have been because the combination audio + tactile display incorporated cues from both the audio and tactile modalities, an advantage in an environment with strong auditory and tactile distractors. The discrepancy between the workload and the performance data indicate that a greater understanding is needed of the role of each modality in on-the-move operations. Future research will deal with multimodal directional cues that can inform Soldiers of important HRI events 360° around of their field of view.

Keywords: *audio,tactile,multimodal,displays*

I. Introduction

The U.S. Army is exploring the use of advanced technologies such as tactile and spatial (3-D) audio displays to enhance Soldier performance in human-robot interaction (HRI) tasks, including monitoring and/or supervisory control of one or more autonomous or semi-autonomous unmanned vehicles (UVs). Particularly important Army HRI tasks include using a UV operator control unit visual display to search for important objects such as targets, and maintaining spatial situation awareness in their environments and around the UV(s). The Soldier has traditionally relied on the visual modality for UV monitoring and supervisory control, but the battlefield provides many conditions that challenge the visual modality, increase operator workload, and hinder situation awareness. Visually challenging conditions include weather, darkness, dust, and noise. Operator workload can be amplified or challenged by cognitively demanding tasks such as individual control of one or more robots by direct control or teleoperation, robot sensor control and interpretation, and air or ground space management. Soldiers responsible for managing UVs may also encounter difficulty when they must maintain their situation awareness of the battlefield environment, of friendly and enemy battlefield entities, and manage robot situation awareness at the same time.

Battlefield challenges may also arise from new demands for Soldier mobility; some Army system concepts propose that robot control operations take place in highly mobile vehicles such as High-Mobility Multipurpose Wheeled Vehicles (HMMWV) in order to enhance robotic command and control function and survivability [1]. In mobile environments, vehicle vibration and jolt may tax Soldier visual performance [2] and visual search [3], making cues in other modalities valuable.

Researchers have shown that spatial audio and tactile cues can be useful by themselves or in combination (as multimodal displays) to supplement visual displays, reduce HRI task difficulty [4] and create a greater sense of operator immersion in robotic tasks [5] over conditions with visual displays alone. Supplementary audio, tactile, or combined audio and tactile cueing have been shown to provide shorter response time than visual cueing alone [6,7]. The purpose of this paper is to describe advanced display technologies that might be useful supplementing visual displays, in highly mobile environments, and to describe a field-study which explored the extent to which vehicle (HMMWV) operations affect user performance with audio and tactile displays. Spatial audio displays, tactile displays, and combination spatial audio + tactile displays are described below.

A. Spatial Audio Displays

With spatial audio displays, also known as 3D audio, a listener using earphones perceives *spatialized* sounds that appear to originate at different azimuths, elevations, and distances from locations outside the head. Because each sound is presented in different spatial locations that are meaningful to the listener, the sounds can provide tracking information regarding object position, velocity, and trajectory beyond the field of view [8,9,10].

Spatial audio displays can communicate events, using sound coming from a number of directional sound sources. For example, radio communications from a commander can sound like they originate from the Soldier's front, a hazardous agent warning signal may come from the Soldier's right, and a signal indicating the position of a remote robot may be heard from the general direction and elevation of that robot. Or, if microphone arrays are installed on the robot, the Soldier can hear the ambient sound transmitted near the robot, which permits the Soldier to hear the environment local to the robot. For a robotic unmanned ground vehicle (UGV), the transmitted sound can convey information such as the motor speed, and the type of surface the wheels or tracks are in contact with. Research has shown that spatial audio cues are useful in human-robot interface target search tasks. Spatial audio displays have been shown to increase user situational awareness in target search of unmanned aerial vehicle (UAV) displays [11] and with target search tasks using narrow field-of-view visual displays the size of a computer monitor [6]. Because they provide positional cues, spatial auditory display cues can also enhance 360-degree situation awareness in applications without a visual display [8].

B. Tactile Displays

Tactile displays use pressure or vibration stimulators, also known as tactors, which interact with the skin [12]. One common example is the vibration function on cell phones. Tactors can be worn individually, or in groups on the user's skin, usually on the hand (in gloves), on the arm, leg, or abdomen (in belts), or on the torso (in a vest). One or more tactors can be used to convey information such as warnings or alerts, by vibrating in patterns consisting of different rhythms or frequencies. A group of tactors positioned in an array can be used to signal directional flow or movement (i.e., by simulating movement to the left or right), which can be useful for applications such as navigational displays. Research has shown that tactile displays have been used to successfully provide safety warning information, and communicate information regarding orientation and direction as well as user position and velocity. Calhoun, Fontejon, Draper, Ruff, and Guilfoos [13] found that tactile displays can significantly improve detection of errors in UAV teleoperation control tasks and can serve as an effective cueing mechanism.

C. Auditory and Tactile Displays

Researchers have explored the use of audio and tactile cues separately and in simultaneous combination as HRI displays for teleoperation as well as for other applications. Gunn, Nelson, Bolia, Warm, Schumsky, and Corcoran [14], and Gunn, Warm, Nelson, Bolia, Schumsky, and Corcoran [15] used multimodal displays to communicate threats in a UAV target acquisition visual search task. They found that spatial (3D) audio and tactile cues used separately enhanced target acquisition performance over no cueing. Chou, Wusheng, Wang and Tianmiao [16] designed a multimodal interface for internet-based teleoperation in which live video images, audio, and tactile force feedback information were combined and presented simultaneously. They found that presenting simultaneous multimodal information reduced operator mental workload relative to no feedback at all.

D. Research Focus

A field study was conducted at the U.S. Army Research Laboratory (ARL) in 2006 to determine the extent to which the integration of spatial auditory and tactile displays affects soldier situation awareness in a simulated UV HRI target search task performed in a moving HMMWV. The objective of the study was to determine whether tactile and 3D audio technologies could effectively convey information in moving vehicle environments that contain relatively high levels of vibration and jolt. A second objective was to examine the extent to which vehicle (HMMWV) operations affect user performance with multimodal cues. Although vehicle vibration has been shown to degrade visual performance, it may also degrade user perception of tactile cues due to the user's need to tense the torso muscles to steady themselves on rough (i.e., cross-country) terrain. It is possible that vehicle noise peaks during travel might mask even well-designed spatial auditory signals. Research was needed to explore to what extent HMMWV vibration and noise affects the integration of audio and tactile cues used to provide information in localization and visual search in HRI target search tasks. This is an important U.S. Army issue because Army system concepts propose that Tactical Operation Centers be emplaced in highly mobile vehicles in order to enhance robotic command and control function and survivability [1].

II. METHOD

A. Participants

Participants were 12 civilian males. All had a hearing level (HL, the decibel level over threshold at which they can hear a test stimulus) as determined by an audiometer-administered hearing test, corresponding to the Army hearing profile H2 or better; an average of no more than 30 dB HL, no individual level greater than 35 HL at 50, 1000, and 2000 Hz, and no level greater than 55 HL at 4000 Hz [17]. Participants also had at least 20:40 vision as determined by a Snellen vision test. Participants ranged in age from 18 to 46 years, with a mean age of 32 years. Participants performed a target search task, in which they searched for one target symbol among 50 non-target symbols displayed on an 18-inch diagonal computer monitor (a 30° field of view (FOV) visual display).

B. Apparatus

Participants received audio and tactile cues as supplements to visual cues, to indicate on which third of the screen the target symbol was located. Audio cues were pre-recorded spatial audio sound files, consisting of the words, “target, target” spoken by a female voice. Tactors were eight tactile sensors developed by Dr. Lynette Jones at the Massachusetts Institute of Technology (MIT) under the ARL Advanced Decision Architecture Collaborative Technology Alliance (CTA) [18]. The tactors were incorporated into a canvas belt that the participant wore on his torso.

C. Variables

The independent variables used were display modality, signal azimuth, and HMMWV movement condition. Participant age was also used as an independent variable, because age had been found to affect response time to display cues in previous experiments. Signal azimuth was defined as the azimuth location of the target symbol on the display, which ranged from -15° to +15°, with 0° being the center of the screen, encompassing the entire 30° field of view of the computer monitor. Display modalities were visual displays with three different modalities of supplementary cues: spatial audio, tactile, and spatial audio + tactile. All display cues were presented 0°, -90°, and +90° (straight ahead, at the subject’s left, and at the subject’s right, respectively). The three HMMWV movement conditions were vehicle at a stop with engine idling, vehicle traveling over gravel road at approximately 12 mph, and vehicle traveling over cross-country terrain at approximately 12 mph.

The dependent variables were participant response time and response accuracy. The participant’s subjective workload rating of display modality was also obtained; participants were asked to rate the workload for each display modality on a one-to-ten scale, with one representing very low workload, and ten representing very high workload. All measures were obtained after the participant traveled over each terrain.

D. Procedure

For the target search task, the computer monitor initially showed a visual display consisting of a topographic map with a red box in the center. After 1.5 seconds the red box disappeared and the computer screen showed the topographic map along with 50 U.S. Army map symbols and one target symbol, with a vertical green cursor now located at the center of the screen. At the same instant, the participant experienced the target cue condition that described the location of the visual target. The participant was instructed to use the alert as a guide to visually locate the target symbol. When the participant thought that he knew where the target symbol was located, he used the knob controller to move the green vertical cursor line on the computer screen as quickly as possible to the location of the symbol he thought was the target. When the participant moved the line on top of the map symbol that he had chosen, he pushed down the knob to indicate that he had located the target.

When the knob had been depressed, the symbology disappeared and the screen with the red box reappeared to refocus the participant’s gaze upon the center of the screen. Then, 1.5 seconds later, the red box disappeared and the next trial began. The green line was relocated at the center of the screen at the beginning of each trial.

For the experimental trials in each movement condition, the participant performed 18 experimental trials (6 trials with each of the 3 display modalities). The 18 targets in the display search task appeared at the different azimuth angle locations between ± 15° azimuth, including 0, without repetition, for each display modality. Six targets appeared on the left side of the screen, between -15° and -6° azimuth. Six targets appeared in the center of the screen, between -5° and +5° azimuth. Six targets appeared on the right side of the screen, between +6° and +15° azimuth. The location of the target symbol and the order of target symbol appearance were random, without replacement. The 18 targets also appeared at random vertical locations within 26°elevation so that half the targets appeared between 0° and 13° elevation, and half appeared between -13° and 0° elevation. At the end of the final display search trial, the movement condition ended. Each movement condition lasted from 20 to 30 minutes.

After each movement condition, the participant provided an on-the-spot single workload rating estimate [19,20] in which they verbally assigned a number between 1 and 10 to describe the workload associated with each display, with 1 being very low workload and 10 being very high workload.

III. RESULTS

A. Response Time and Accuracy

Accuracy data indicated that participants located over 99% of the targets correctly (there were no differences as a function of any condition or signal modality), indicating that response accuracy was not a function of display modality or condition. A multivariate analysis of variance (MANOVA) for the response time data, which used Wilk’s criterion $\underline{U}$ as the test statistic for within-subjects data and $\underline{F}$ for between-subjects data. Effects which were shown to be significant were explored through the use of selected LSD post-hoc tests. The results indicated that main effects of movement condition ($\underline{U}$ = .192, $\underline{p}$ = 0.001) and age ($\underline{F}$ = 3186.661, $\underline{p}$ = 0.001) were significant, as were the two-way movement condition x display modality interaction ($\underline{U}$ = 0.257, $\underline{p}$ = 0.05), and the three-way movement condition x age x display modality interaction (U = 0.084, p = 0.001). There were no other significant main effects or interactions. Figures 1 through 3 shows the three-way interaction, by showing the movement condition x display modality interaction for each of the three age groups.

Post-hoc tests indicated that for all display modalities, participants in their 20s generally had shorter response times than participants in their 40s. As can be seen in Figures 1 through 3, no one display modality provided the shortest response time to all age groups, for all terrains. During engine idle, the only display modality that provided shorter response times within one age group was the tactile display, which generated a significantly shorter mean response time for participants in their 30s. On gravel terrain, the only display modality that provided significantly shorter response times within one age group was the audio + tactile display, for participants in their 40s.

Travel over cross-country terrain provided several differences between display modalities, within and across each age group, with no one display modality standing out as providing the greatest benefit. Participants in their 20s showed the shortest response times with audio, and audio + tactile displays. Participants in their 30s had significantly shorter response times with audio + tactile displays. However, participants in their 40s showed significantly shorter response times with audio and tactile displays. As can be seen, cross-country terrain provided a great deal of variability between age groups and display modalities. This may be due to the small number of participants in their 30s and 40s; there were twice as many participants in their 20s (6 participants) as there were in their 30s and 40s (3 participants in each age group). The relatively small number of older subjects may have been one source of variability on the demanding cross-country movement condition. Future studies should include larger numbers of participants in their 30s and 40s to duplicate U.S. Army demographics, especially since the maximum recruit enlistment age rose in 2007, from 34 years to 39 years for the U.S. Army National Guard and the Reserve [21]. The significant three-way display modality x age x movement condition interaction precludes the interpretation of the display modality x movement interaction as well as the main effects of age and movement condition.

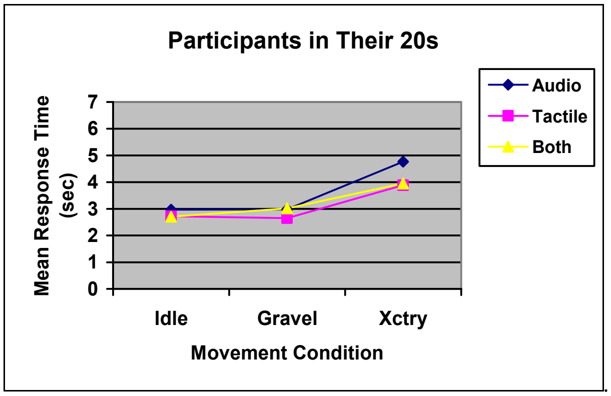


Figure 1. Mean Response Time for Movement Condition x Display Modality for Participants in their 20s

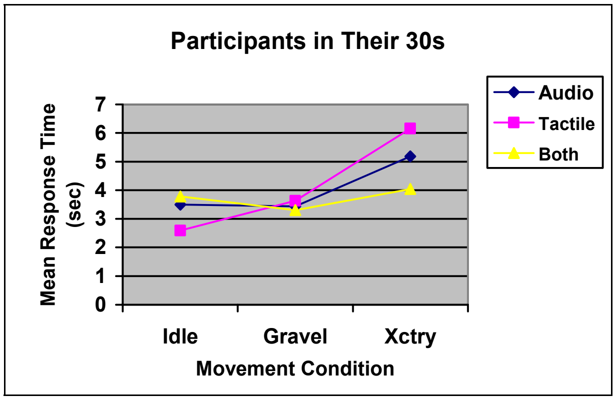


Figure 2 Mean Response Movement Condition x Display Modality for Participants in their 30s

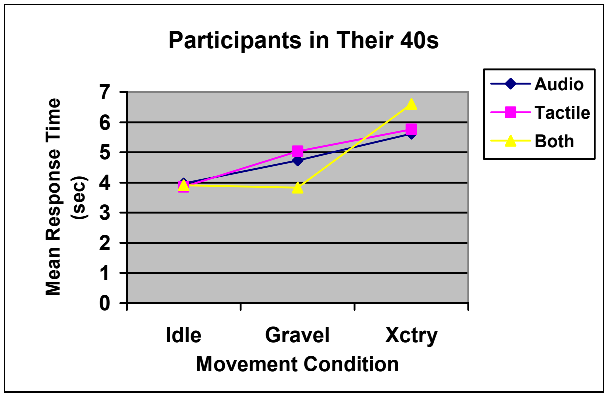


Figure 3. Mean response time for Movement Condition x Display Modality for Participants in their 40s

B. Workload

An analysis of variance (ANOVA) indicated that for workload data, main effects of display modality and movement condition were significant ($p \leq 0.05$). There were no other significant main effects or interactions. A least significant differences post-hoc test was performed for significant effects.

Figure 4 contains mean workload ratings for the different display modalities. Post-hoc testing indicated that participants rated combination tactile + audio displays as having a significantly lower workload than audio and the tactile displays used separately. There were no other significant differences. One reason for the significantly lower auditory + tactile workload rating may have been that the combination audio + tactile display incorporated cues from both audio and tactile modalities, allowing one display modality to provide cues because the combination is more powerful in an environment with strong auditory and tactile distractors.

Figure 5 contains mean workload ratings for the different movement conditions. Least significant difference post-hoc testing indicated that participants' workload ratings were significantly greater after the cross-country condition, than for the engine idle and gravel conditions. There were no other significant differences. The workload ratings mirror the response time data; cross-country terrain had higher levels of noise and vibration, which had an impact on participant workload ratings.

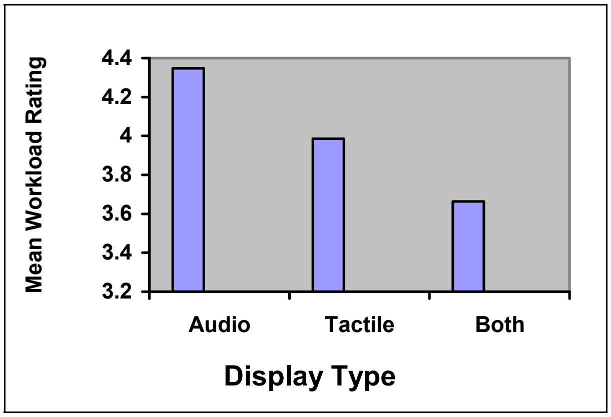


Figure 4. Mean Workload Ratings for Display Modalities

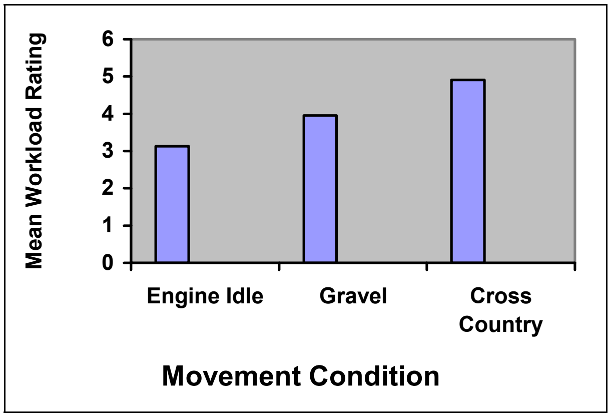


Figure 5. Mean Workload Ratings for Movement Conditions.

IV. CONCLUSION

Alone or together, advanced technologies such as tactile and spatial (3-D) audio displays can enhance user performance in HRI target search tasks. Previous research had shown that supplementary audio, tactile, or combined audio and tactile display modalities have been shown to provide shorter response time than visual cueing alone [7]. The results of the current study indicated that for target tracking response time and accuracy on a narrow field-of-view visual display, the audio, tactile, and audio + tactile displays performed equally well as supplements to a visual display on most terrains.

Data also indicated that performance time and accuracy with the tactile display were not limited by movement or vibration on the gravel or cross-country terrain. Results show that for the target search task, tactor output was not masked by participant contact with the seat back during vehicle bumps on the gravel or cross-country terrain.

The results indicated that display modality was significant in terms of participant workload ratings (perceived workload), but was not significant for response time. One reason for the significantly lower auditory + tactile workload rating may have been that the combination audio + tactile display incorporated cues from both the audio and tactile modalities, an advantage in an environment with strong auditory and tactile distractors. However, user perception was not reflected in actual user performance.

The discrepancy between the workload and the performance data indicate that a greater understanding of the role of each modality in on-the-move operations is needed. The audio display modality may have been limited by the lack of headtracker and individualized spatial audio algorithms, which may have made them a bit more difficult to localize (determine from which direction the cue originated), and thus deliver a perception of greater workload. Future localization display research is important and relevant not just to target tracking, but because directional cues can inform Soldiers of important HRI events 360° around of their field of view. ARL research in the next fiscal year will explore the integration of audio with tactile cues in HRI displays, to communicate multiple levels of information.

ACKNOWLEDGEMENT

The authors would like to thank Dave Ostrowski, James Gombash, Krishna Pillalamarri, and Theodoric Feng for their assistance in the conduct of this study.

REFERENCES

[1] Emmerman, P.J., Grills, J.P., and Movva, U.Y. (2000). Challenges to Agentization of the Battlefield. Proceedings of the International Command and Control Research and Technology Symposium. U.S. Department of Defense.

[2] Ishitake, T., Ando, H., Miyazaki, Y., and Matoba, F. (1988). Changes of visual performance induced by exposure to whole-body vibration. Kurume Medical Journal, 45(1), 59-62.

[3] Griffin, M.J., and Lewis, C.H. (1978). A review of the effects of vibration on visual acuity and continuous manual control, part I: Visual acuity. Journal of Sound and Vibration, 56: 383-413.

[4] Lathan, C.E., and Tracey, M. (2002). The effects of operator spatial perception and sensory feedback on human-robot teleoperation performance. Presence: Teleoperators and Virtual Environments, 11(4), 368-377.

[5] Burdea, G., Richard, P., and Coiffet, P. (1996). Multimodality virtual reality: Input-output devices, systems integration, and human factors. International Journal of Human-computer Interaction: Special Issues of Human-Virtual Environment Interaction, 8(1), 5-24.

[6] Haas, E.C., Pillalamarri, K., Stachowiak, C., and Lattin, M. (2005). Audio cues to assist visual search in robotic system operator control unit displays. U.S. Army Research Laboratory Technical Report ARL-TR-3632, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD.

[7] Haas, E.C., Stachowiak, C., Pillalamarri, K., and Lattin, M. (2006). Integrating audio and tactile displays for guiding visual search in robotic system OCU displays. Unpublished manuscript, U.S. Army Research Laboratory, Aberdeen Proving ground, MD.

[8] Perrott, D.R., Sadralodabai, T., Saberi, K., and Strybel, T.Z. (1991). Aurally aided visual search in the central visual field; effects of visual load and visual enhancement of the target. Human Factors, 33(4), 389-400.

[9] Elias, B. (1996). The effects of spatial auditory preview on dynamic visual search performance. Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting, 1227-1231.

[10] Fujawa, G.E., and Strybel, T.Z. (1997). The effects of cue informativeness and signal amplitude on auditory spatial facilitation of visual performance. Proceedings of the Human Factors and Ergonomics Society 41st Annual Conference, pp. 556-560.

[11] Simpson, B. D., Bolia, R. S., and Draper, M. H. (2004). Spatial auditory display concepts supporting situation awareness for operators of unmanned aerial vehicles. In D. A. Vincenzi, M. Mouloua, and P. A. Hancock (Eds.), Vol. I.- Human performance, Situational Awareness, and Automation: Current Research and Trends (pp. 61-65). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

[12] Gemperle, F., Ota, N., and Siewiorek, D. (2001). Design of a wearable tactile display. Proceedings of the 5th IEEE International Symposium on Wearable Computers, 5-12.

[13] Calhoun, G., Fontejon, J., Draper, M., Ruff, H., and Guilfoos, B. (2004). Tactile versus aural redundant alert cues for UAV control applications. Proceedings of the Human Factors and Ergonomics Society 48th Meeting (pp. 137-141).

[14] Gunn, D.V., Nelson, W.T., Bolia, R.S., Warm, J.S., Schumsky, D.A., and Corcoran, K.J. (2002). Target Acquisition with UAVs: Vigilance Displays and Advanced Cueing Interfaces. Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting, 1541-1545.

[15] Gunn, D.V., Warm, J.S., Nelson, W.T., Bolia, R.S., Schumsky, D.A., and Corcoran, K.J. (2005). Target acquisition with UAVs: Vigilance displays and advance cueing interfaces. Human Factors, 47(3), 488-497.

[16] Chou, Wusheng, Wang and Tianmiao (2001). The design of multimodal human-machine interface for teleoperation. Proceedings of the IEEE International Conference of systems, Man and Cybernetics, Volume 5, 2187-3192.

[17] U.S. Army (1991). Hearing Conservation. U.S. Army Pamphlet 40-501, Washington, D.C.: Department of Defense.

[18] Lockyer, B. (2004). Operation manual for the MIT wireless tactile control unit. Cambridge, Massachusetts: Massachusetts Institute of Technology.

[19] Hart, S.A., and Bortolussi, M.R. (1984). Pilot errors as a source of workload. Human Factors, 25(5), 575-556.

[20] Moray, N., Dessouky, M.I., Kijowski, B.A., and Adapathya, R.S. (1991). Strategic behavior, workload, and performance in task scheduling. Human Factors, 33, (6), 607-629.

[21] Garamone, J. (2007). Army reserve components boost enlistment age limit. U.S. Department of Defense American Forces Press Service News Articles, July 23, 2007, http://www.defenselink.mil/news/newsarticle.aspx?id=31126.

Autonomy (What's it Good for?)

J. P. Gunderson
Gamma Two, Inc.
1733 York Street
Denver, CO, USA
jgunders@gamma-two.com

L. F. Gunderson
Gamma Two, Inc.
1733 York Street
Denver, CO, USA
lgunders@gamma-two.com

Abstract**—** This paper focuses on the impacts of mission requirements and environmental conditions on the decision to incorporate autonomous components into a cognitive system. The costs of autonomy are discussed, along with the benefits, and the interrelationships between the environmental context and task requirements are explored with respect to the application of autonomous systems. We address two fundamental questions: "Given the environment and the design goals of the intelligent system, can autonomy be enabled? Second, if it can be, should it be?"

Keywords**:** *artificial intelligence, autonomy, machine cognition, reification*

I. Introduction

The decision to include autonomous capability in any deployed system is complex. It extends beyond the purely technical issues into economic, social, and safety concerns. While there can be clear benefits to deploying an autonomous cognitive system, there are also costs in many potential areas. There are costs associated with the design, development, and testing of an autonomous system; and there are additional costs associated with the loss of control (either actual or perceived) that occur with the granting of autonomous behavior. This paper presents some of these costs, and discusses the cost benefit analysis required by the decision to incorporate autonomy into any system. In specific, we look at the environmental drivers that push towards autonomy, the capabilities needed to support autonomous behavior, and the effects of the interaction of intelligence and autonomy on the ability of a deployed intelligent system to achieve mission goals.

Before discussing the relative merits of adding autonomy to an intelligent system, it would be good to define what these characteristics are. Recent papers have focused on the differences between intelligence, ability, and autonomy (see [3] for an analysis).

A. Intelligence

Defining intelligence, even in the restricted domain of intelligent systems, can be problematic. We use a simple working definition, derived, in part, from work by Albus and Meystel[5]. Albus and Meystel define intelligence (after expanding a few subordinate terms) in the following manner: *Intelligence is the ability of a system to act in a manner that increases the probability of successfully achieving the system's goals in an uncertain environment.*

We differ from their definition in only one respect, and that is to separate the development of the actions from their execution. Our working definition is:

> Intelligence is the ability to formulate one or more action sequences which can increase the probability of successfully achieving the system's goals in an uncertain environment.

The purpose of the change is to separate the cognitive capabilities of the intelligent system from the execution capabilities.

B. Autonomy

Given that one has an intelligent system, what then is an Autonomous Intelligent System? There is a tendency to equate 'autonomous' with 'hands-off operation.' As long as there is no human with a joy-stick controlling the system, it must be autonomous. However, this would suggest that any system that follows a hard-coded routine is autonomous. As Luc Steels[13] suggests, such systems may be automatic, but they are not autonomous. We believe that autonomy is more than simply playing back a script.

The dictionary definition of autonomy is the ability to self-govern. This can be defined as the ability to choose one's own course. This ability entails two conditions, first that there exist options, and second that the choice between these options is determined by the system itself. This results in the following definition:

> Autonomy is the capability of system to select between multiple possible action sequences to achieve the system's goals, based on the current situation and internally defined criteria.

In a dynamic world, with complex goals, it is normally the case that many possible solutions to any problem exist. However, it may be that in certain highly constrained domains only one course of action will achieve a goal. In this case, autonomy may hold vacuously.

The key feature of this definition is the system's ability to select between multiple courses of action. This is in a dynamic tension with intelligence. Intelligence is frequently related to the concept of finding not only *a* solution to a problem, but finding *the best* solution. However, autonomy suggests that the choice between solutions is not imposed from the outside.

Consider a simple system that plays music. If this system is given a hard-coded play-list, where each song is played in a specific order, there is clearly no autonomy. Now imagine the same system, which is programmed to play the least recently played song. Again, there is no autonomy since there is only one song that is the least recently played.

Next imagine that the software plays the least *frequently* played song. Now it is likely, that several songs will have been played the same number of times, and so the system would have to select between a number of possible courses of action that would each achieve the system's goal. Now the system begins to exhibit autonomous behavior.

C. Factors affecting Autonomy

Using these definitions of intelligence and autonomy, the question becomes "Can autonomy add value to an intelligent system?" and "Should autonomy be applied in a specific case?" The rest of this paper looks at what capabilities must be present in the intelligent system to enable autonomy, and what characteristics of the domain, the environment, and the tasks affect the decision to implement an autonomous solution.

II. Benefits of Autonomy

Increasingly, intelligent and cognitive systems are being deployed into real world environments to achieve task specific goals. The domains into which these systems are deployed range from complex financial applications to military robotic platforms to deep space probes. In each of these cases, there are demands that the system achieve the mission goals reliably in complex and uncertain environments. Before we can answer the question of what benefits autonomy can provide, we must look at the current state of deployed systems.

A. Person in the loop

Most of these systems are tightly controlled 'person in the loop' systems. In the case of unmanned vehicles, this coupling may require multiple humans to operate a single deployed system. In many cases the primary function of the human in the loop is to process the operational data into information (is that a truck parked under the trees?) and deciding what to do about it (ignore that truck for now). However, there are limited humans available for these jobs and they can quickly become fatigued. As a result, there is increasing demand to reduce the tight coupling between the human controller and the deployed system. This reduction is typically accomplished by off-loading the more routine operations onto the deployed system.

In a teleoperated system, much of the value that the human brings to the system is the ability to recognize new and unexpected situations and to respond appropriately to these situations. In these situations, simply enabling autonomy has little value, unless the intelligent system can recognize the salient aspects of the environment, and formulate appropriate responses.

When an intelligent system is deployed into in domain which is neither completely known, nor completely predictable it approaches certainty that the system will encounter situations that its designers did not envision. The system has no prior knowledge of the situation, or how to achieve it goals from this state, so it must depend on its human component to first make sense of the situation, and then determine the correct actions to apply.

The second significant capability that the human brings to a teleoperated system is the ability to choose between several different options. A ground vehicle, tasked with the goal of reaching a point on the other side of river, might have several options. It could drive south several kilometers to a bridge, it could drive north a shorter distance to a ford, or it could attempt to cross the river at its current location. This is a complex, multi-criteria decision problem which is affected by many dynamic conditions. While it could be possible to formulate a decision theoretic description of the problem and (subject to the underlying assumptions) produce an optimal solution, this is the kind of problem for which the human can produce a satisficing solution and implement that solution on the fly.

B. Autonomy Can Free These Limited Resources

Humans bring a powerful arsenal of skills and capabilities to a teleoperated system. However, skilled people are in short supply, are expensive assets to risk, and become fatigued quickly. The demands to replace the human in the loop with an intelligent, autonomous system are increasing. However, there are significant limitations to our current ability to provide viable autonomous solutions to these problems.

The value of an autonomous, intelligent system lies in its ability to achieve its goals without relying on a human agent to tell it what to do. To achieve this in a given domain it must be capable of providing for itself those functions that the human provides in a teleoperated system. As discussed above those two major functions appear to be a) the ability to interpret the features of the dynamic domain, and b) the ability to choose between multiple options and select one that will increase the probability of achieving the goals.

III. Necessary Capabilities to support Autonomy

A. Decision Making

As described above, the most obvious capability required by an autonomous system is the ability to choose between multiple options. This topic area has been part of artificial intelligence research since its inception, indeed it might be argued that the study of artificial intelligence has been the study of how to generate problem solutions and choose between them[10][11].

Autonomous decision making can be extremely simple, although it is challenging for a system to make good decisions. A common approach is to rank candidate decisions into groups of 'roughly equivalent' classes, where the ranking criteria are derived from the need to increase the probability of achieving the system goals. The system now looks at the class with the highest probability, and selects at random from within that class. This provides the benefits of a quick decision heuristic, which an procedure that provides a good probability of meeting the system's goals. However, this decision mechanism is dependent on the autonomous system having a good representation of the current state of the domain, and an ability to interpret those objects and events for which it has limited prior knowledge.

As an example, consider the recent series of DARPA Grand Challenges. In these events, autonomous ground vehicles were given a goal of completing a course defined by a series of waypoints. The vehicles had to handle all of the second by second driving decisions, based on models of the world, sensor data providing the current situation, and a set of 'road rules' (for example, the vehicle shall not deviate from the course by more than x meters).

Imagine one such vehicle traveling across the desert, the laser range finders providing a view of the obstacles ahead, the stereo cameras providing data about the edges of the roadway, and either GPS or inertial systems providing information about the vehicle's current location, and the locations of the waypoints. Suddenly, the sensors indicate a roughly one meter obstacle in the center of the roadway ahead. The intelligent system can quickly generate a set of responses:

1. Pull off the roadway and pass the obstacle on the side;
2. Drive over the obstacle; or
3. Sit and wait, perhaps the obstacle will go away.

Pulling off the road increases the risk of failing to achieve system goals (either by getting stuck in the sand, or by violating the road rules), hitting a rock is equally risky, but running over a tumbleweed has little risk. In this situation, the ability to make a 'good' autonomous decision is clearly dependent on having a good representation of the current state of the domain.

From this example, it is clear that the ability to produce a representation of the domain is a key capability for autonomy. Since any system that requires autonomy is going to run into situations for which it has no prior programming, it must be capable of generating a representation of this new situation. If it cannot form some representation of the situation, it cannot predict the future states of the domain. Without this representation, it cannot generate the possible actions that might be used to achieve the system goals.

In addition, if it cannot predict the possible results of the choices, it would be reduced to selecting between them blindly, and thus would have no effective ability to increase the probability of achieving its goals. Therefore any system deployed into a risky, unpredictable world must have the ability to perform a bidirectional mapping between a representation of the domain, and the signatures and affordances of the real world objects and those representations. This capability has been defined as reification.

B. Reification

In previous work we have defined reification as the bi-directional mapping of representation (symbolic data) to sensor data[4]. It is the process that allows humans to look out the window and see, not a roughly rectangular blob of red surrounded by black (which is transected by yellow lines), but a sports car in a parking lot. In addition reification allows us to utilize that sports car to stop and pickup milk on the drive home – thus achieving one of the system's goals.

Historically there have been two approaches to enabling intelligent systems to situate themselves in the world. These are represented by artificial intelligence researchers who work top-down from deliberative symbol manipulation and those who bottom-up from control systems in robots. The general consensus has been that as the two ends work towards the middle, the gulf will narrow and narrow until it disappears. However, recent research has suggested that the gulf may not be bridgeable by work from either side; rather it may require a specific research approach that is different from either the sensor-based or the symbolic domains.

From the point of view of the deliberative approach, a symbol manipulation system is developed, and it is outside the scope of the symbol system to recognize the physical and perceptual characteristics that define the thing referred to by the symbol. From the viewpoint of the embedded systems approach, the crucial task is the recognition of physical and perceptual cues, while mapping those cues onto a symbol system is outside the scope of the research.

Underlying both these beliefs is the assumption that once the core research was addressed, it would just be a matter of pushing the research frontier towards the opposing viewpoint until they met. If one continues the bottom-up (or top-down) approach long enough, eventually one gets to the top (or bottom) and the complete problem is solved (See Figure 1.A).

Both the top-down and the bottom-up approaches have made great strides towards the complete solution. However, there seems to be a gap that neither has been able to cross. It is clear that both the sensor/effector-to-symbol pathway and the symbol-to-sensor/effector pathway are necessary to support deployed intelligent systems. It is this bi-directional pathway that constitutes reification.

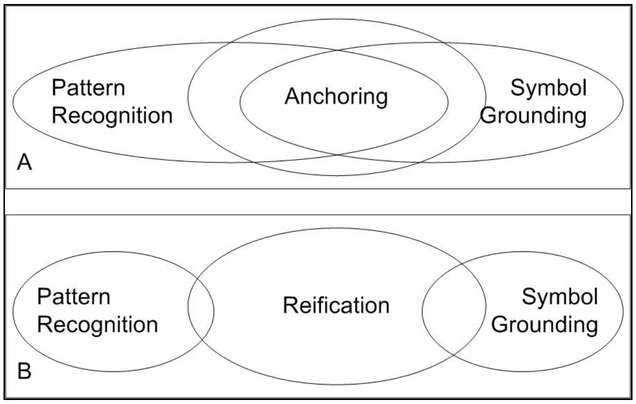


Figure 1 - Possible relationships of Pattern Recognition, Symbol Grounding, and Reification. In A, the problem of anchoring symbols to sensor/action patterns should be approachable by either top-down or bottom-up improvements. However, in B the problem cannot be solved by either top-down or bottom-up approaches, since there is no area of overlap. Rather, a third approach is required; one that solves the reification problem first, which then provides the bridge between symbol and sensors.

We have argued that this reification capability is necessary for any autonomous system that is deployed into the real world, where it will encounter situations which prevent it from achieving its goals, and for which it has to intelligently produce a collection of possible solutions, and autonomously select from these possible courses of action the one that it will undertake.

IV. Costs of developing autonomy

Adding autonomy to an intelligent system increases the complexity of the software. Adding any new capability will increase the costs associated with the design, development and testing of software, but when there is significant coupling between the existing software and the new capabilities, the "ripple effect" can radically increase the resulting complexity of the software[9]. The question arises whether adding autonomy causes such ripple effects.

A. Coupling between Autonomy and Intelligence

From the working definition presented in section I, it is clear that there are a number of commonalities between intelligent performance and autonomous performance. Intelligence is defined as the ability to generate actions that will improve the probability of the system achieving its goals. Autonomy is defined as the ability to select from these actions. At a basic level, a system can be autonomous, if it simply selects at random from the available options, and, in this case, there would be little coupling between the intelligence and the autonomy. However, implicit in the concept of increasing the probability of achieving the system's goals is the notion of making a 'good' selection from the available options. This suggests that there are several system wide aspects (in the Aspect-Oriented Programming sense[1]) which increase the amount of coupling between the base functionality (an intelligent system) and the added capabilities of an autonomous system.

B. Coupling between Autonomy and Reification

In addition to the coupling between intelligence and autonomy, there is also a coupling with the reification process. The autonomous selection of an appropriate response depends not only on the system's goals, and capabilities, but on the situation in which the system is making this decision. As a result the selection of a 'good' response is made in the context of how the current situation is interpreted, which means that the reification system must also provide information to the software supporting autonomy. This is another cross-cutting aspect of autonomy, which will increase the costs of developing the autonomy capability.

C. Testing an autonomous system

While the increased coupling between the system modules will result in synergistic effects in the cost of developing the software, this can pale in comparison to the added costs in testing that are required by an autonomous system. Once a system is autonomous, it becomes, in many ways, non-deterministic. After all, it was the very facts that the environment is uncertain and that we could not pre-program the correct responses to all situations that motivated the decision to add the capability of autonomous behavior to the system.

Under these conditions we can do preliminary testing of the system by putting it into pre-analyzed situations, for which we believe that we know the correct responses. However, this will only tell us that in the situations for which there is no need of autonomy, the autonomy hasn't broken anything. This testing does not tell us if the system will perform correctly in the very situations where we need to it perform.

Frequently, the only effective way to test the autonomy is to use simulation techniques (Monte Carlo, Bootstrapping, etc.) to generate a wide range of 'typical situations' and then to perform a statistical analysis of the performance. This, in turn, requires the development of high resolution, high fidelity simulators of the domain and the environment. The increased costs of designing, developing, and testing the simulations, as well as the supporting code for the gathering and statistical analysis of the simulation testing will add significantly the overall costs of adding reliable autonomy to an otherwise intelligent system.

V. Factors Affecting the Autonomy Decision

Given that autonomy is a viable solution, many of the operations we would like to off-load require constant low-level decisions to be implemented in response to changing environmental states. There appear to be two independent criteria that influence the decision to enable autonomous operations on a deployed system: the predictability of the environment, and the risk associated with autonomous operations

A. Predictability

The predictability of the environment is a critical and complex aspect of the decision to enable autonomy. We are using predictability in the modeling sense; specifically: to what degree can the future state of the domain be accurately predicted from the current state. Fundamentally, if the environment is completely predictable, then every possible state that the deployed intelligent system can encounter can be evaluated in advance, and (in theory, see Schoppers[12]and Ginsburg[2]) 'correct' responses for each of these can be provided to the system.

Predictability can be compromised in several different ways. In a dynamic environment with significant exogenous events (for example a battlefield scenario), the future state of the environment can only be predicted statistically, if at all. However, even in environments where there is 'complete knowledge' (e.g., chess) the enormous complexity of the state space makes it impossible to produce a universal plan. Typically, complete predictability can only be reached in small and simple problem domain, or for problem domains that are totally engineered – such as factory floor and laboratory automation systems.

If the environment is not predictable for either reason, it becomes difficult to avoid some level of both intelligence and autonomy. Since it is almost certain that some situation will be encountered for which the deployed intelligent system has no pre-defined, correct response, the system will have to generate one or more possible responses if it is going to achieve its goals. This will require intelligence. If there are multiple possible responses, the system will have to choose between them. This will require autonomy.

This leads to an indirect relationship between the predictability of the domain and the need for autonomy in the system. The greater the predictability, the less autonomy is needed.

B. Risk

Risk is another key area that can affect the decision to enable autonomous behavior. Since autonomy entails the system making its own 'choices', there is always the possibility that it will make a choice with which the humans would disagree. This leads to the question what happens if the 'wrong' decision is made?

In previous work we have explored autonomy in the domain of music selection[5]. We developed a music playback system that autonomously selects music to minimize the rejection rate. The Personal DJ has the responsibility of selecting music to play, and if it picks music we do not like, there is little risk. We hit the reject button, and life goes on.

In other contexts the risk of a wrong decision increases. Such a risk can be financial, for example if the Personal DJ system is enabled to purchase new music the risk is increased significantly. Other risks include damage or loss of property, for example, during the DARPA Grand Challenges several autonomous vehicles left the roadway, struck obstacles, or had other accidents. These risks include property damage, and in the upcoming urban challenge, the risks could include harming humans. In the future, an autonomous convoy that 'decides' to take a different route, thus delaying a delivery of critically needed medical supplies, has taken a risk which could result in deaths. In a non-deterministic environment any choice can turn out badly, resulting in after-the-fact analyses into 'what went wrong'.

This gives us another indirect relationship between the potential risks associated with deploying an autonomous system and the characteristics of the domain and the tasks. The greater the risk that a bad decision involves, the less likely that autonomy should be enabled.

C. Trust and loss of control

In addition to the domain and environment, there are social issues associated with the decision to enable autonomy. Whenever an autonomous system is deployed it is in a social space where there are people who are affected by the system. Even an autonomous system deployed as a planetary rover on a moon of Jupiter (such as the DEPTHX project[8]), while it may not encounter humans while it is exploring, many humans will be affected by it performance, and jobs and careers may be impacted by its autonomous behavior.

Ultimately, the decision to deploy an autonomous system comes down to the question "How comfortable am I that this system will achieve its goals without causing unacceptable harm?" This question is the same whether the autonomous system is a robotic device or a human sent out to do a mission. We are, in general, more used to making this decision when it is a human. But the same question is addressed every time we delegate a responsibility to any type of machine. Even something as simple as the introduction of antilock brakes (a primitive autonomous system) required several decades to gain acceptance[6].

The problem is increased with autonomous, intelligent systems, since in many cases there may be honest disagreement, even among the human experts, as to what the correct course of action is in a given situation. The deployed autonomous system will choose some course of action, which, due to the non-deterministic nature of the world, might succeed or fail. Regardless of the outcome, there is likely to be some expert who will second guess the machine.

The issue of trust is, in part, correlated to the risk in the environment. Clearly, if there is significant risk the need to trust the system must increase. However, it extends beyond the actual risk into perceived risk. Individuals, businesses, and societies vary in their need for control, and the essence of an autonomous system is that one gives up some level of control. Even if the risk is effectively zero, there is still a loss of control when a goal is delegated to an autonomous system. In some cases this loss of control may be the driving force in the decision to implement autonomy in an intelligent system – regardless of the ability of the system to achieve its goals.

VI. Autonomy: What is it Good For?

The domain, the intelligent system, and the mission all interact to determine whether autonomy is viable in a given system. If it is not viable, it becomes ineffective to attempt to 'off-load' autonomous operations from the human to the intelligent system. We have categorized the domain into a broad spectrum based on the predictability of the environment. With low predictability there is more value that an autonomous system can provide, with high predictability, there is less need for autonomy. In the same manner, the risks associated with a 'bad' autonomous decision can be broken down into two broad categories, low and high risk: with low risk, there are ample opportunities for deploying autonomous systems, with high risk domains and missions, the decision should be approached cautiously.

These results are shown in Table 1, below.

Table 1

High level partitioning of task and environment space with respect to autonomy.

	Low Predictability	**High Predictability**
Low Risk	Autonomy encouraged	**Autonomy optional**
High Risk	**Task dependent autonomy**	**Autonomy discouraged**

A. Low Risk, Low Predictability

This category is a perfect ground for autonomous systems. Since the environment is not easily predictable, it is very challenging to develop a non-autonomous system that will perform well. Currently, this category is populated by teleoperated systems which tie up humans to act as information processors, and decision makers. These are humans who are frequently highly skilled, and are easily fatigued. While the presumption is that they are in control of the machine components of the system, in reality they are often nothing more than biological sensor suites. In these cases the use of autonomy would be of great value.

B. High Risk, Low Predictability

In this category the decision to implement autonomous systems is more problematic. There are tasks such as planetary exploration beyond Mars, where the option of teleoperated control is simply infeasible, and the risks are primarily financial. In this situation, deploying autonomous exploration systems is reasonable. However this category also includes battlefield deployment of unmanned combat air vehicles, where the risk to human life resulting from autonomous activity probably far exceeds any gain. So we assign this 'task dependent autonomy,' where the decision to implement autonomy is controlled by the specific level and type of risk, and the possible gains.

C. Low Risk, High Predictability

In domains where there is high predictability the added value of an autonomous system is low. Since the environment is well defined, it is frequently more cost effective to forego autonomous solutions. However, there are likely to be niches where an autonomous system may be more effective than building an automated system. In addition, since the risks are lower, this is an excellent class of domains for research and development of autonomous systems technology.

D. High Risk, High Predictability

This type of system includes areas such as controlling refineries, power plants, and other domains where an error can have grave consequences, but the domain is well studied and fairly predictable. In this category autonomy is not really advised. The risks are high, and the marginal benefits of an autonomous system are low.

VII. Conclusion

In this paper we have presented working definitions of intelligence and autonomy that reduce the confounding effects of these terms. Based on these definitions, we have explored the requirements needed to support autonomy, the conditions in which autonomy provides benefits to the system, and have looked at the inherent costs of enabling autonomy in an intelligent system. From this we have proposed a simple classification schema that can be used to evaluate the question "In this domain, with these tasks, is it effective to develop an autonomous solutions to this problem?"

What is an intelligent autonomous system?

It is a system that has the ability to develop new responses to situations it encounters in a dynamic and uncertain world, and to choose between those responses to increase the probability of achieving its goals. It is intelligent if it can encounter new situations and develop responses to achieve its goals, it is autonomous if it can select between those responses without requiring instruction from other systems. By this definition a human, teleoperating a remote control vehicle is an intelligent, autonomous system. However, the focus of this work is to look at the decision to replace the human controlling the vehicle with a cybernetic system of some configuration.

What capabilities are necessary to enable autonomy?

Before autonomy can be considered, there are several capabilities that must exist in the intelligent system. The system must be capable of producing multiple possible responses to the situations it encounters. If the system can only produce one possible response, there is no need for autonomy. Second, the system must have a mechanism that allows it to select between these responses in a manner that will increase the probability of achieving the system's goals. Finally, if autonomy is required the system will encounter situations which involve novel states. In order to produce appropriate responses, and select from these effectively, the system must be capable of reifying these novel situations.

When is autonomy beneficial?

It is beneficial in situations where the ability to pre-program known, 'good' responses is limited due to the environment's constraints. These constraints can be due to lack of knowledge about the environment, or due to significant uncertainty about the state that might be encountered. Whenever it is not feasible to pre-analyze the domain, autonomy can bring significant benefits to the system.

What are the contra-indications for enabling autonomy?

Clearly, if the environment is well defined, well behaved, and can be modeled, there is limited value to adding autonomy. Having an autonomous telephone dialer is probably of little value. If the risks associated with failure are high, the decision to enable autonomy must be carefully thought out.

References

[1] T. Elrad, R. E. Filman, and A. Bader, "Aspect Oriented Programming", Communications of the ACM, vol. 44 No. 10, pp. 29 – 32, 2001.

[2] M. L. Ginsburg, "Universal Planning: An (Almost) Universally Bad Idea," AI Magazine, vol. 10 no. 4, pp.40 – 44, 1989.

[3] J. P. Gunderson and L. F. Gunderson, "Intelligence ≠ Autonomy ≠ Capability", Proceedings of Performance Metrics for Intelligent Systems, 2004.

[4] J. P. Gunderson and L. F. Gunderson, "Reification: What is it and Why Should I Care?",Proceedings of Performance Metrics for Intelligent Systems, 2006.

[5] L.F. Gunderson, T. Kilgore, and J. P. Gunderson, "Living with a Personal Disk Jockey – the Start of the Journey", Persistent Assistants: Living and Working with AI, American Association for Artificial Intelligence Technical Report SS-05-05, 2005, pp. 34-38.

[6] T.A Horan. and R. T. Barnes, "Public acceptance of automated highways: results from national focus groups," Proceedings of the Vehicle Navigation and Information Systems Conference, 1995, pp. 382 – 389.

[7] A. M. Meystel and J. S. Albus, Intelligent Systems Architecture, Design, and Control. New York: John Wiley and Sons, 2002, ch.1, pg. 3.

[8] NASA DEPTHX project: http://astrobiology.arc.nasa.gov/news/expandnews.cfm?id=10644

[9] R. S. Pressman, Software Engineering A Beginners Guide. New York: McGraw Hill, 1988, ch. 4, pp. 106-108.

[10] E. Rich and K. Knight, Artificial Intelligence 2nd edition. New York: McGraw Hill, 1991, ch.1, pg. 3.

[11] A. Samuel, "Some studies in machine learning using the game of checkers," IBM Journal of Research and Development, vol. 3(3), pp. 210-229, 1959.

[12] M. J. Schoppers, "Universal Plans for Reactive Robots in Unpredictable Environments," Proceedings of the tenth Joint Conference on Artificial Intelligence, vol. 2, pp. 1039 – 1046, 1987.

[13] L. Steels, "When are robots intelligent autonomous agents?", Journal of Robotics and Autonomous Systems, vol. 15, pp. 3-9, 1995.

Definitions and Measures of Intelligence in Deep Blue and The Army XUV

John M. Evans, Ph.D.
John M Evans LLC
1 Reservoir Road
Newtown, CT 06470
john_evans@snet.net

Abstract—Three different definitions of intelligence are reviewed, using Deep Blue as the basis for comparison, and a discussion of chess rating points as a metric of performance is presented. It is argued that Deep Blue showed intelligent behavior and passed a form of the Turing Test. Further applications of similar search techniques in the Army XUV are shown to generate behaviors that show elements of intelligence in autonomous systems.

Keywords: *definitions of intelligence, measures of intelligence, computer chess, Deep Blue, autonomous vehicles, path planning, Army XUV*

I. Deep Blue

A remarkable milestone in computer science was achieved in 1997 when IBM's computer, Deep Blue, beat the World Champion, Garry Kasparov, at chess [1,2]. This was the culmination of fifty years of work on what was considered a problem that "penetrated to the core of human intellectual endeavor."[3] This event ranks with the Wright Brothers first flight and with the achievement of sustained fission in the Manhattan Project: early success with what became (or in this case will become) world-changing technologies.

It is interesting to note that Hsu, the designer of the VLSI chips used in Deep Blue, characterizes the matches with Kasparov not as man versus machine but rather as man as performer (Kasparov) versus man as toolmaker (Hsu and the IBM team). [1] Deep Blue was a remarkable tool, quite successful at the specialized task for which it was designed.

II. Definitions of Intelligence

Was Deep Blue "intelligent"? Artificial Intelligence researchers generally say no, that Deep Blue's success depended on special purpose chips that were designed only for evaluating chess moves. Philosophers and psychologists generally say no, that there was no self-awareness, no consciousness, no real understanding in Deep Blue. However, the English mathematician Alan Turing in 1950 proposed an operational definition of "intelligence" that basically said that if a person interacting with another unknown entity could not distinguish between a computer and a person, then that entity would have to be considered intelligent. [4,5]

The Turing Test defines "intelligence" in terms of black box functionality of a machine in comparison with a human in human/machine interaction. Searle, with his famous Chinese Room argument, redefines intelligence in terms of understanding. [6, 7] Hawkins, in his book On Intelligence, defines intelligence in terms of predictive ability. [8] These three definitions are not necessarily incompatible, although Searle was specifically attacking the Turing Test definition.

Deep Blue had all of these facets of intelligence. Kasparov felt he was not playing with a machine but with an independent intelligence, an entity with an independent mind: "Now for the first time we are playing not with a computer, but with something that has its own intelligence."[9] He made this statement after winning game 2 of the first match in Philadelphia in 1996. In a less charitable mood during the losing rematch in New York in 1997, he accused IBM of cheating, of having a person directing the game. [10] From Kasparov's standpoint, Deep Blue passed a version of the Turing Test.

III. The Structure of Deep Blue

Deep Blue also had predictive capability (Hawkin's definition of intelligence) and embodied understanding of the game of chess (Searle's definition). To understand this we must delve into how Deep Blue operated.

The basic principles of playing chess with a computer were laid out by Turing in England and Shannon in the U.S. by 1950 [11, 12]. Since then there has been refinement and the addition of bells and whistles, but the main point is that increasing computer power makes it possible to look farther ahead in a game and that in turn leads to greater skill and therefore greater perceived "intelligence".

The chess game for a computer is divided into three parts. [1,2] The first part is the opening book, a sequence of scripted moves that have been played out many times in the past. This is essentially table lookup.

The second part of the game uses search techniques to evaluate different possible moves. At each level of search a quantitative value is calculated based on material and board positions and the search is selectively deepened along the most promising lines. The weights given to different pieces and different board positions were developed with the help of grandmasters and it is in these evaluation functions that chess knowledge is embedded in Deep Blue. Deep Blue calculated an average of fourteen plies (half-moves) but in some cases went to twenty-ply or even thirty-ply deep evaluations in examining possible lines of play. [1,2]

The final part of the game is the endgame. Deep Blue had all four piece endgames stored in main memory and all five piece and some six piece endgames stored on disk. At this point it is again a table lookup strategy.

In the first and last part of the game, understanding of the game of chess is embedded in the "book", the tables, which were created by human grandmasters. In the middle part of the game, understanding of chess is embedded in the evaluation functions, which again were set by grandmasters working with the chip designers and the programmers. In terms of Searle's argument, Deep Blue did not understand what was going on when it executed the evaluation functions, but it did ***embody*** understanding of the game of chess, and hence, from Kasparov's viewpoint, it seemed to possess intelligence.

From the standpoint of Hawkins definition, Deep Blue was exhibiting intelligent behavior by being able to predict the future results of its actions. This is the essence of cost-based search.

Cost based search is not how humans play chess at the highest levels. [13] Instead, we exploit the massive parallel processing capability of the human mind together with the basic ability of the neocortex to recognize and store patterns and sequences of patterns (8) and play mostly on the basis of pattern recognition. Functional MRI tests show that chess experts activate primarily the parietal cortex, where spatial patterns are stored, while novices activate primarily the temporal cortex, where information on individual pieces and their capabilities are stored. Either approach, pattern recognition with massively parallel processing or search with a Von Neumann architecture computer, obviously works.

IV. Performance Metric for Chess

In the case of chess there is an established performance metric, Elo's chess rating points system [14], adopted by FIDE (Fédération Internationale des Échecs) as the official international rating system for chess players. This is a statistical rating based on head-to-head competitions: a player taking 3 out of 4 points in a four game match is considered 200 rating points above the losing player. Newborn has developed experimental data based on running real chess matches through computer chess programs. His data indicate that increasing the depth of search by one level gives an increase of at least 100 rating points in skill level. [15] It is then just a matter of applying sufficient computing power to reach a level of skill beyond that of any human. Deep Blue was able to exercise the equivalent of two to three trillion instructions per second (using hundreds of special purpose VLSI chips on a cluster computer with 36 processors) which allowed it to examine two hundred million board positions per second and it played in New York at a rating level of over 2800, on a par with Kasparov [1,2].

The following data on ratings for specific chess playing computers and computer programs is from Newborn [2].

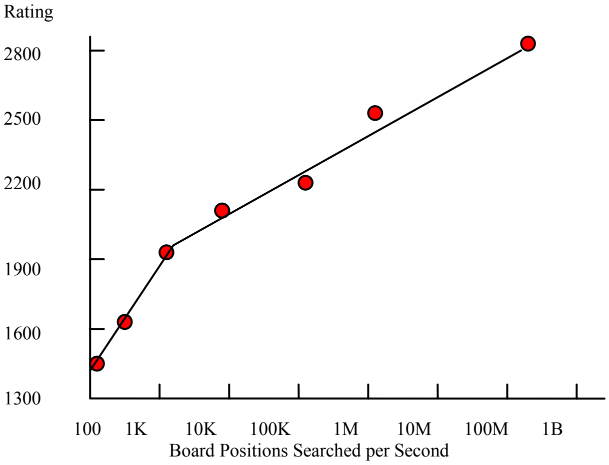


Figure 1: Rating vs Speed of Search

V. The Army XUV

Another interesting example of using cost-based search to generate complex behaviors is the vehicle control for the Army's Experimental Unmanned Ground Vehicle Program, commonly referred to as Demo III, which ended in 2003. [16, 24] (Demo I was teleoperation, Demo II was supervised autonomy, and Demo III was targeted at full autonomy for scout missions of reconnaissance, surveillance, and target acquisition). Many of the ideas from Demo III were embodied in the Stanford and Carnegie Mellon winners of the DARPA Grand Challenge Road Race in 2005 and this technology is now being developed into the Autonomous Navigations System for the Army's Future Combat Systems vehicles.

This special vehicle, shown in Figure 2, has four wheel hydraulic drive with four wheel steering. The navigation sensors include scanning ladar, stereo cameras and stereo FLIRs, microwave radar, bumpers, tilt sensors, GPS and inertial navigation. The mission sensors are in the dome package on a shock mount on top of the vehicle.

The path planning for Demo III used cost based search. [17, 18, 19, 21] The cost function is

$$\text{Cost} = \mathbf{\Sigma}\, c_i * v_i$$

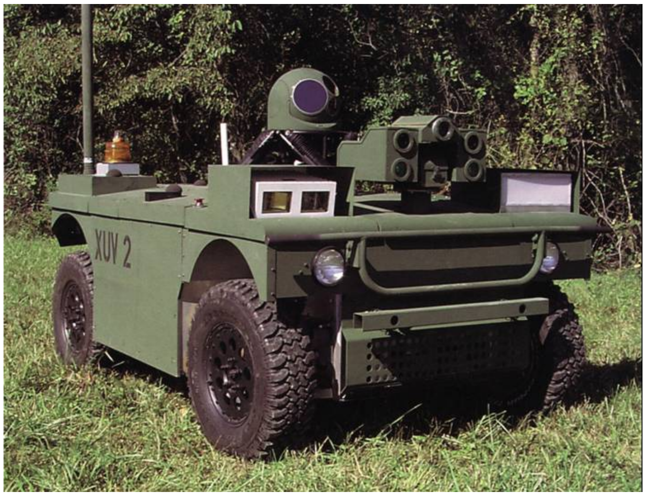


Figure 2: Army XUV (Experimental Unmanned Ground Vehicle)

where the c_i are relative costs or weights (relative importance) of different relevant state variables and v_i are the current values of those variables. There were 16 variables used in the cost function, including side slope, forward to back slope, ground roughness, ground center height, soil properties, on-road, off-road, vegetation, obstacles, and mission completion time.

Cost maps were created using a priori knowledge (maps) and real time terrain knowledge from vision and ladar scanners. A grid of points was cast onto the maps and points were connected to provide possible path segments. A search was then conducted, calculating the cost for each path segment encountered and deepening the search along favorable directions to find the lowest cost path from a starting point to a finishing point. Search was carried out at several levels of resolution: 5 m, 50 m and 500m range maps. The 500 m maps gave optimal start and end points for the 50 m maps, which in turn gave start and end points to the 5 m map. At the 5 m map level, pre-calculated trajectories that embodied vehicle dynamics (speed, inertia, possible steering rates, etc.) were used for computational efficiency instead of an arbitrary point grid. [19]

Neural nets, which model how our brains work, carry out exactly this type of computation, summing the product of neural signal strengths times synapse weights. As Churchland and Sejnoski note this neatly wraps vector representation with compatible matrix processing and allows for many types of mental computations. [20] We use chemistry to create emotions to modify weights for low level behaviors (e.g. fight or flee) and logical computation for more abstract behavior generation.

The following figures, developed by Balakirsky [21] show examples of behavior generation for the Demo III vehicle under different weight assumptions. The figures are from a simulated operator control unit and use data from topological maps of the grounds of the National Institute of Standards and Technology (NIST). The NIST maps were only course resolution, so the images seem blurred. Red areas are obstacles (trees, bushes and fence lines), blue are buildings, and green are roads and parking lots.

Figure 3 shows a hypothetical path from one point to another with the weight for obstacles being high and all other weights being low. The faint white line is a direct line from the start (lower right) to the finish (just beyond the road in the upper left) and the heavier yellow line shows the computed path. The vehicle finds an opening through the trees and then returns to the direct path to the goal.

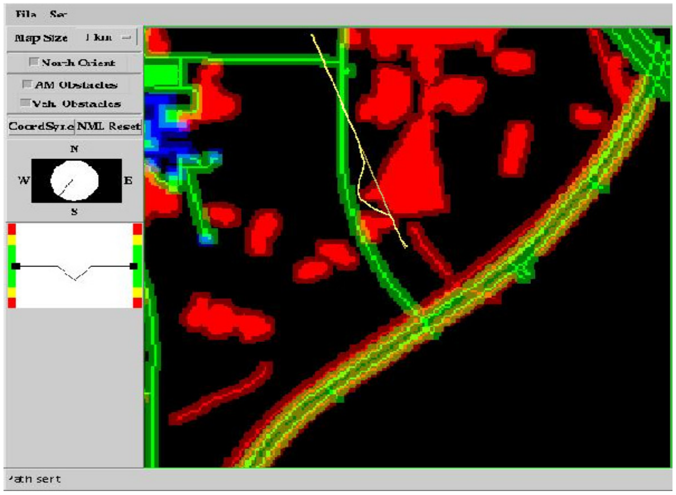


Figure 3: Simulated path planning with obstacles having high weights

In Figure 4 the weights for being off-road are set very high, so the vehicle does not go directly toward the goal but instead heads toward the nearest road and then stays on roadways until it is as close as it can get to the goal, at which point it departs from the road and dashes to the finish.

Finally, in Figure 5, the weight for being out in the open and hence potentially detectible by an enemy force is set very high and the cost for mission completion time is set low. The result is stealth behavior, running carefully along the tree line to stay under overhanging branches as much as possible. This is a tactically significant behavior for an Army scout, and it was generated by the wonderfully general approach of cost based search that matches how our own minds determine appropriate behavior.

Future work will include learning and recalling appropriate weights for different situations. The ability to recognize contexts and select appropriate weights depends in turn on improved perception to generate image understanding and situational awareness. The Demo III program only scratched the surface of these issues.

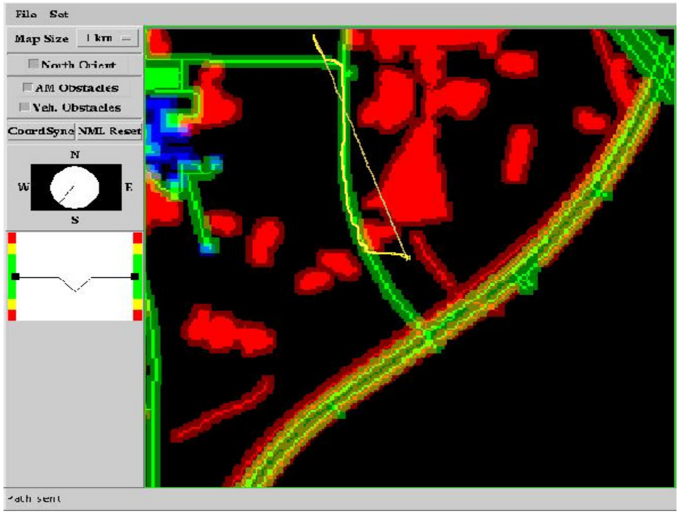


Figure 4: Path Planning with low cost for On-Road and high cost for Off-Road

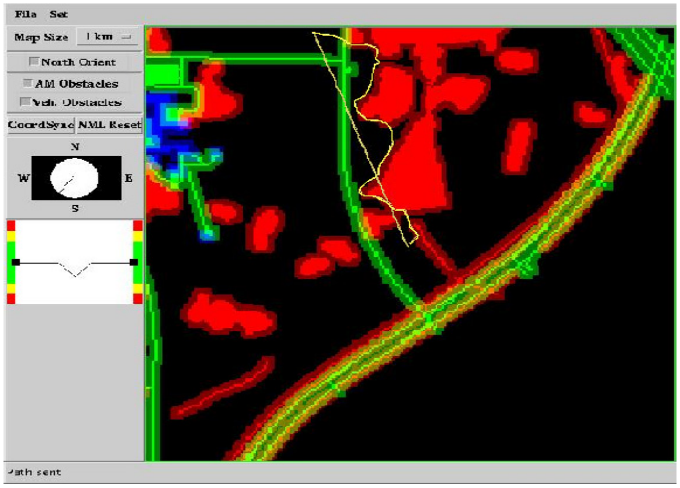


Figure 5: High Cost for being in the open (detectable by an enemy scout) and Low Cost for mission completion time.

VI. Performance Metrics

There is not single metric for performance of unmanned vehicle planning systems to match the chess rating system. Instead the focus has been on measuring the performance of unmanned systems in navigating known courses of various difficulties. Measures of performance include completion of segments of the course, time to completion, number of targets found in a course and number of interventions. [22, 23, 24] For smaller robots artificial test courses have been used for evaluation in both simulation and in contests. [23, 25].

VII. Conclusion

This paper has examined some interesting idiot savant capabilities: Deep Blue beat Kasparov and must be considered intelligent in its domain, but all it does is play chess. The Army XUV showed some intriguing elements of intelligent behavior, but it was badly nearsighted, had very limited perceptual understanding and was computationally bound, so it also showed some truly dumb behaviors at times and got lost on a regular basis. Still, the techniques of planning and problem solving, one aspect of intelligent robot systems, are now fairly well understood, objective test methods and metrics are being developed to benchmark capabilities and rapid progress is being made.

References

[1] Hsu, Feng-Hsiung, *Behind Deep Blue*. Princeton: Princeton University Press, 2002.
[2] Newborn, M., *Deep Blue*. New York: Springer-Verlag, 2003.
[3] Newell, A., Shaw, C. and Simon, H., from an early paper on chess, as quoted in Hsu, *op.cit.*
[4] Turing, Alan, "Computing Machinery and Intelligence". *Mind*, **LIX**, pp433-460, 1950.
[5] Hodges, A., *Turing.* New York: Routledge, 1999.
[6] Searle, J., *Minds, Brains and Science*. Cambridge: Harvard University Press, 1984.
[7] Searle, J., "Is the Brain's Mind a Computer Program?" *Scientific American*, **262** (1), 1990.
[8] Hawkins, J., *On Intelligence*. New York: Henry Holt & Co., 2004
[9] Kasparov, Garry, as quoted in Newborn, *op. cit*.
[10] Kasparov, Garry, as quoted in Hsu, *op. cit.*
[11] Turing, Alan, "Digital Computers Applied to Games", in *Faster Than Thought*, B.V. Bowden (ed.), London: Pitman, 1953.
[12] Shannon, Claude,"Programming a Computer for Playing Chess", *Philosophical Magazine*, **41**(7), 1950.
[13] Ross, Phillip E., "The Expert Mind". *Scientific American*, August, 2006.
[14] Elo, A., conceived of the statistical performance rating system used in chess. http://en.wikipedia.org/wiki/ELO_rating_system.
[15] Newborn, *op. cit.,* Appendix H.
[16] Shoemaker, C., et al., "Demo III: Department of Defense Testbed for unmanned Ground Mobility", *Proc. SPIE Conference on Unmanned Ground Vehicle Technology*, SPIE Vol. 3693, Orlando, FL, April, 1999.
[17] Albus, J., et al., *4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems*, NISTIR 6910, Gaithersburg, MD, 2002.
[18] Schlenoff, Madhavan, R., and Balakirsky, S., "Representing Dynamic Environments for Autonomous Navigation", *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, 2003.
[19] Balakirsky, S. and Lacaze, A., "World Modeling and Behavior Generation for Autonomous Ground Vehicle", *Proc. IEEE International Conference on Robots and Automation*, 2000.
[20] Churchland, P. and Sejnoski, T., *The Computational Brain.* Cambridge: MIT Press, 1999.
[21] Balakirsky, S., *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Spaces*, Berlin, Germany: IOS Press, 2003.
[22] Messina, E. and Meystel, A., *Proc. PerMIS* 2000-2004. Gaithersburg, NIST SP 982, 990, 1014, 1036, 2000-2004.
[23] Jacoff, A., Messina, E., and Evans, J., "Performance Evaluation of Autonomous Mobile Robots", *Industrial Robot*, **29**:3, May 2002.
[24] Madhaven, R., Messina, E., and Albus, J., *Intelligent Vehicle Systems*. New York: Nova Science Publishers, 2006. Chapter 3 covers behavior generation; chapter 9 covers performance evaluation; chapter 10 is a review of the Army Demo I-Demo III programs.
[25] Jacoff, A., Weiss, B., and Messina, E., "Evoluation of Metrics and Performance of USAR Competitions", *Proc. 2003 PerMIS*, Gaithersburg, MD: NIST SP 1014, 2003.

Automotive Turing Test

Steven F. Kalik and Danil V. Prokhorov
Toyota Technical Center-TEMA
Boston, MA, Ann Arbor, MI, USA
{steve.kalik, danil.prokhorov}@tema.toyota.com

***Abstract*—** The Turing Test is often cited when the intelligence level of a presumably intelligent computer program is to be assessed. We propose to extend the scope of the test to the domain of intelligent automotive vehicles. We discuss possible formats for such a test, and consider different measures produced by these tests.

***Keywords*:** *Turing test, imitation game, human-like driving behavior, human driving intelligence, intelligent vehicle, robot driver, vehicle intelligence, road test, driving simulator, vehicle interaction, safe driver.*

I. Introduction

Alan Turing's original paper [1] proposed two central ideas that revolutionized thinking about modern computing. Of the two, perhaps the more revolutionary idea was the conception of the Universal Computing machine. But, the one which in many ways did more to capture the imagination of society, is the Turing Test (TT), based on the "Imitation Game" as Turing originally coined it.

As originally proposed, the imitation game pits an interrogator (C) against a man (A) and a woman (B). The interrogator is separated from the man and woman in such a way as to limit his observations to only the interactions they have with him and with each other. As Turing originally conceived it, those interactions were all verbal, relayed to the interrogator via an intermediary or, more preferably, through a set of teletype terminals. Thus, the responses of each player to the questions of the interrogator, and to each other's responses to those questions, define the only input the interrogator receives to decide which is the man and which the woman. What makes the game far from simple is that only one of the players is trying to be correctly identified. In this case, let's describe it as Turing did, as player B. The remaining player, A in this case, seeks to deceive the interrogator by giving them answers and statements that they hope will convince the interrogator to apply the opposite labels to the two players. The interrogator may ask questions and observe the responses until such time as he feels he knows which labels to apply, but the ability of A to give false answers or to respond in ways that would deliberately mislead the interrogator ultimately leaves the interrogator with doubt about which player to trust, and therefore with only an estimate of which player is which. This makes the game challenging with the placement of labels decided in a sense statistically, based on a set of responses observed by the interrogator.

As is well know, Turing's ultimate variant of this game replaced A with a computer, posing the critical question: "Will the interrogator decide wrongly as often when the game is played like this, as he would when the game is played between a man and a woman?" [1]. This form of the question creates a statistical measure of distinguishability out of the set of decisions made by the interrogator as they play the imitation game multiple times. A comparison of the distinguishability measures between the cases when humans exclusively play the roles of A and B in the game, and when machines enter the game and take the role of the deceiver (A), utilizes these measurements to quantify the original question Turing sought to improve upon, which was simply "Can machines think?"[1]. Thinking in this case carries the normal but poorly-defined meaning humans usually ascribe to it when they speak of human thought.

We propose a similar approach in this paper: use human decisions about their notions of human and machine behaviors to determine statistically over a set of decisions whether behavior observed appears intelligent enough to be called by the same name. Thus we propose the Automotive Turing Test (ATT), and explore some basic ideas and properties suggested by such a test.

We begin by identifying the central features of the original Turing test in Section II, and consider the implementation of those features in the automotive domain in Section III. Section IV presents several different ways the automotive Turing test may be constructed. We discuss different features of those implementations. Section V considers the goals and measures of the ATT with regard to human goals for intelligent vehicles, pointing to required future work. Finally, Section VI concludes the paper.

II. Key components of the Turing Test

Turing's original conceptualization of the TT provided the following benefits:

1. A way to explicitly quantify a vague idea ("thinking" in Turing's case).
2. Use of a human as an active sensor to identify human-like behavior
3. Simplification of the test structure to probe the essence of the assessed behavior while maintaining impartiality.

It is this third point that attracts our attention here.

To legitimately retain the TT label in the work we propose for this paper, we must retain the key elements of the original TT :

1. Limited observations between the players and the interrogator

2. The ability of the players and the interrogator to interact directly in the domain of the test and not beyond
3. The opportunity to try to deceive the interrogator, and
4. The ability of both players and interrogator to recognize and influence the context of the observations and the responses returned during the game.

While we feel that it is critically important to maintain the quantitative nature of the TT, with its central measure that compares ensembles of label applications between cases when (1) two humans act as players and (2) a machine and a human act as the players, (in both cases the interrogator being human), we recognize that this is not the only version of the game people think of today (see, e.g., [3]). But, to the extent possible we adhere to these key features of the test as originally conceived. Where we diverge from this, we note it explicitly.

III. Turing test in the automotive domain

Just as Alan Turing posed the question "Can machines think?", we now pose the question "Can machines drive?" Just as Turing's use of "think" carried with it human aspects of thinking, so too do we carry components of human driving into this test, seeking to capture aspects that we associate with a human driver into our use of the word "drive". As this paper and perhaps future work will illustrate, the exact form of what we will call the **Automotive Turing Test (ATT)** is still evolving. The Turing test goal of being indistinguishable from a human may or may not be the target we as a society choose to approach, given the number of traffic accidents observed each year across the globe. On the other hand, as we point out, there may be some benefits to the limitations humans show, too.

To convert the Turing test into the automotive domain, we must transform the key elements from section II into a form appropriate for testing driving behavior, and into interactions appropriate to a driver controlling an automobile. To do so, we remove from consideration the verbal interactions central to the original TT, and replace them with behavioral and signaling interactions available to vehicles.

So far we conceive the ATT to be administered in two different environments, and in each environment, the test can be administered in two different ways. Each of these provides opportunities for the test to address different aspects of human driving behavior.

Environments

The environments in which we initially imagine the test being administered are:

- On real vehicles
- In a life-like simulation environment

In both cases, to stay with the spirit of the TT, the interrogator must be prevented from seeing anything inside any vehicles to which they will be expected to apply labels, and by extension, from seeing any information other than the vehicles behavior.

In either testing environment we require that the space over which the vehicles will drive be large enough, in terms of road length, space covered, and potential driving situations that can be encountered, so that drivers will need to learn a map of the environment and patterns of dynamics in that environment to get around conveniently and effectively. This assures the exercise of driving skills at both tactical (on the order of one second) and strategic (seconds – minutes – hours) scales during the administration of the test.

As some situations in the real world occur on the road much less frequently than others, we recognize the importance of explicitly building the opportunity for these events to occur into the testing environment. For example, near-accident situations are much less common than normal driving situations [2]. Yet, such rare situations, if recreated either by surprising behavior from another vehicle, or through the inclusion of unexpected behavior from other agents in the testing environment, would present great opportunities to assess the limits of human and intelligent machine capabilities. Hence the ATT environment, whether in a real or a virtual car, should include the ability to implement driving surprises.

Administration methods

The administration methods we initially imagine are:

- Vehicle to Vehicle format: Each participant – the interrogator and both players A and B - drive separate vehicles in the environment.
- Road Test format: A single player is to be labeled as a human or machine driver by an interrogator who sits in the vehicle to be observed, similar to the way a human driving instructor would pass judgment on a driver's license applicant during a student driving test.

In both of the administration methods, human drivers have access to maps or navigation systems to support their route choice decisions. We offer driver-readable maps or driver-understandable navigation systems to support preparation for and driving in the testing environment, so that a standard access to knowledge of the environment can be assured.

Since today's vehicles communicate with each other in only limited ways (obviously excluding driver sticking his head out the window and yelling or gesturing) if the test were to be run today, only standard signaling devices such as turn signals and brake lights would be available for communication from the driver. Observations of the environment including all vehicles, agents, objects, etc., are also visible to all players and the interrogator to inform everyone involved of the actions happening around them.

However, if at a future time, additional technologies were added to standard vehicles such as message passing equipment or warning systems for vehicle to vehicle communication, or aircraft-like "black box recorders" to record all vehicle and environmental data, one could easily imagine incorporating such tools into the ATT. We elaborate on the use of data recorders for the purpose of the ATT in Section IV.

How the tests are run

We distinguish the test vehicles of players from all others in the environment by clearly marking or labeling such vehicles. To run an instance of the imitation game allowing only vehicle to vehicle interaction for the case of two test vehicles, we place vehicles labeled X and Y respectively at their starting locations. The interrogator gives them goal locations to reach which may differ from each other's, but which will require them to traverse a common subset of streets and intersections, and which will lead them to encounter common driving situations, or to create driving situations for each other. The interrogator knows to which destinations these players will need to drive.

When the players reach their destinations, they may be given new destinations that will again take them over another common set of roads and through common intersections and traffic situations according to the requirements set in the previous paragraph. The interrogator always knows these destinations so that he can choose his own actions to allow observation of and/or desired behavioral interaction with the players over the course of their trip. We allow this process to iterate until the interrogator decides he knows which labels A and B to ascribe to the vehicles originally designated X and Y.

Once the labels have been applied, a round of the game is considered over, and the correctness of the decision recorded. New assignments, new starting positions and destinations are selected for the test vehicles.

Although Turing originally constructed the test to provide an elaborately quantitative measure of whether a human interrogator interacting with an intelligent machine could be convinced to mislabel the players in the game as frequently when the players were both human as when they were human and machine, a more common usage today asks a simpler variant of that question: can a human observer distinguish the behavior of the intelligent vehicle from the behavior of a vehicle driven exclusively by a human driver.

When this alternate version of the TT is used, as in the Road Test administrations, we modify the previous route granting procedure to allow the interrogator to explicitly modify the route as necessary at any time during the test to exercise the drivers' capacities and to explore their decisions and behaviors in the face of route selections or modifications, as will be discussed in more detail later in the paper.

In the next section, we explore the implications of the different testing environments and the opportunities raised by testing in each combination of administration and environment.

IV. Variants of the Automotive Turing Test

Given the two different administration methods described above, and the two different environments in which the test could be given, there are already four different variations of the test under consideration. A fifth, passively interrogated form of the test will also be briefly considered later, wherein only databases are reviewed and compared to decide whether the driver is human or machine. We begin here by describing four variants with active interrogators, as shown in Table 1.

TABLE I

AUTOMOTIVE TURING TEST VARIANTS

AUTOMOTIVE TURING TEST VARIANTS		ENVIRONMENT	
		Real World	Simulated Environment
ADMINISTRATION	Vehicle-to-Vehicle Format	Variant 1: Real World Environment with only Vehicle–to- Vehicle Interaction	Variant 2: Simulated Environment with only Vehicle-to-Vehicle Interaction
	Road Test Format	Variant 3: Real World Environment with Road Test Format	Variant 4: Simulated Environment with Road Test Format

Variant 1: Real World environment with Vehicle-to- Vehicle interactions.

In this first variant described, we imagine an interrogator and two separate players in their respectively marked vehicles, receiving destinations and route suggestions and driving to them as described earlier.

Remember that the interrogator in this vehicle-to-vehicle interaction variant can supply new destinations and route guidance only before a player sets out on their route to the destination. Along the way, the interrogator, when seeking to change a vehicle's route, must interact with the players' vehicles only in the same ways that other vehicles in the environment can interact with the players. That is, interrogators influence the players' vehicles by driving close enough to them to make the actions of the interrogators' vehicle relevant to the driving decisions of the players. The type of interaction described here is imagined to primarily influence a player's tactical behavior in the time surrounding the interaction.

The interrogator may also want to test strategic decisions by a driver, such as choosing when to take a new route to a destination. To do this, the interrogator might try something like trying to tie up traffic so the player's vehicle must re-evaluate, and perhaps revise, the route they planned to take. However, performing actions such as these influence a potentially very large number of people in addition to the players in the game, and may be considered inappropriate for the Real World Vehicle-to-Vehicle variant of the ATT. The route change can be more safely addressed in the domain of simulated environment, and more explicitly addressed through the Road Test variants.

Variant 2: Simulated environment with Vehicle-to-Vehicle interactions.

Just as the Vehicle-to-Vehicle interaction variants can be administered in the real world, so too can they be administered in a high-fidelity simulation environment, or in a virtual reality world. In this variant we remove the risk inherent in real world experiments to provide a safer test environment that still permits direct but controlled interactions between the players and the interrogator. This permits the interrogator to consider a broader range of actions with which to challenge the players than they had available to them before, when lives, limbs, property, and in fact even simply the convenience of non-involved parties were at stake. Now the penalties for behaviors that would influence these concerns can be modulated and explored, and if desired, even ignored in the construction of the testing environment.

For example, imagine an interrogator wishing to test the sensitivity of players to loss of time resulting from perturbations of the traffic flow through the environment in which the test is taking place. The interrogator could influence and test what it takes to get a response from the players by moving to a position he knows both vehicles will need to pass through or inescapably close to. (In many older cities, such central transit points are easy to imagine.) By uncivilly blocking a high traffic intersection, or perhaps more simply, by just adjusting the gaps at any intersection that the interrogator will accept to enter or pass across a traffic flow, the interrogator can modify some gross behaviors of the traffic system that will ripple out from that part of the environment. This puts players in the position of having to decide whether to wait out the delay, or to select an alternate route. Patterns in this behavior (and the resulting interactions with vehicles and other agents in the environment) can then be used to help the interrogator distinguish between different players before they make their decisions which player is human and which a robot driver. Possibly, in the simulated environment, players themselves might also select some of these behaviors in an effort to lead the interrogator to the labeling conclusion that the player seeks to achieve.

However, this experimental freedom for the interrogator comes at a price. Now, many of the things that came for free in the Real World environment such as sensory stimulation, and the inherent costs humans associate with risk to their person or property must be artificially created. For example, to provide the same sights, sounds, smells, and other sensory phenomena to the driver, we require novel simulation platforms that can re-create those effects. While a few high fidelity simulators exist in the world today, in most cases many variables are ignored if they are not critical to the experiment of interest. (Olfactory stimulation, for example, need not be included as part of following a truck in a simulator, but in the real world it might influence the decision a human driver would make about whether to pass a truck or remain behind it on the road.)

The key advantage to both Vehicle-to-Vehicle interaction variants of the ATT are that they adhere most closely to Turing’s original test.

Whether the simulated reality in the ATT is a perfect recreation of any known location in our world is not important for this vehicle-to-vehicle variant of the test. But, since one typical purpose in measuring the “intelligence” of a vehicle is to help decide when such a vehicle would be safe to introduce to our roads in the real world, we would benefit by requiring all vehicles to treat the simulated environment the same way they would treat the environment in the real world. To achieve this, we need them to act in the following ways and have the following characteristics:

- They must obey traffic laws and traffic control devices and customs;
- They must avoid curbs, trees, vehicles and pedestrian figures;
- They must have similar fields of view and limitations on their knowledge of the environment as they would in the real world (limited but wide field of view dependant on modeled head orientation; olfactory, auditory and tactile feedback separated from the visual field of view; reasonable directionality to all sensory input to provide the normal input available for a human driving in a vehicle); and
- They must have limited but reasonable control capabilities over the virtual vehicles they are driving, similar to the capabilities they would have over vehicles they would drive in the real world.

Having made these demands of our test to help fit it to a human goal for the domain of intelligent vehicles, we observe that this strong requirement can actually be relaxed without upsetting the pure *intelligence* testing measure developed by the imitation game. This is based upon the notion that the imitation game versions of the TT and ATT are not really about *safety*, or obeying traffic laws *per se*, but about producing behaviors that are indistinguishable from human behaviors in the environment in which they are observed.

We claim here that in imitation game based ATT’s, obeying traffic laws will only matter to the players of the game to the extent that humans also choose to obey those laws in the given environment. However, if we ultimately hope to convince ourselves and society that an intelligent vehicle is as safe as, or safer than, human driven vehicles on the road (and therefore worthy to share the road with humans), we will want to consider ways to create the test environment so the value of obeying traffic laws and civil behavior (and the cost of disobeying them) is clearly understood and similar for all players in the testing environment, and is similarly weighted to what it is now in real world. The often raised issue about how similar is human behavior in a simulated environment is then revisited here, but this time with an eye to parameters that can be built into the simulation environment to help enforce the kind of behavior we might desire. One considered approach to enforce this is to offer something of value (perhaps points that can be inherently valuable to a machine, or exchanged for prizes or money by a human player after they finish taking part in the test), and to place "law enforcement" drones in the simulated environment (location-locked camera-like systems or mobile police-like vehicles in the environment) that either deterministically or probabilistically penalize the player for violations of the laws in that environment. Such additions to the environment might be a noted increase in the effort necessary to construct it, but might still be beneficial for future consideration.

However, a test that tests for differences from a standard obviously carries with it all of the strengths and weaknesses of that standard, should one achieve the goal of being indistinguishable from the standard. With this in mind, we now turn our attention to other variants of the ATT whose measures perhaps allow more freedom in what is being measured. The Road Test format, with its implication of Pass or Fail grading for driving at a level suitable to be considered worthy of being called human, is perhaps more easily adapted to address the societal goal of intelligent vehicles that are not just human-like, but as clever as a human and at least as safe.

Variant 3: Real World environment with Road Test format.

This variant is similar to the variant 1 except that it prevents the interrogator from driving his or her own vehicle. However, in contrast to the variant 1, the interrogator is always present in the vehicle, and he can observe its behavior constantly, rather than episodically, as in the first two variants.

The interrogator can provide instructions to the driver in at least two ways, verbally or electronically. The instructions should be relevant to driving, e.g., "make lane change", "turn right over there", etc. This variant of the ATT should not turn into a domain-nonspecific testing the driver for natural language understanding.

The instructions provided to the driver must be given with a reasonable lead time for the driver to execute the appropriate maneuver safely. Alternatively, the instructions could be delivered at any time so long as the driver is empowered to just ignore illegal or unsafe instructions from the interrogator.

Variant 4: Simulated environment with Road Test format.

This variant is similar to variant 3 except that the real world environment is replaced by the simulated environment. In contrast to variant 2, the interrogator cannot drive his own virtual vehicle, but may enjoy the benefit of constantly observing the player's behavior.

Variant 5: Passive interrogation via recorded behavior comparison to database of human behavior.

A fifth variant, alluded to earlier, is available if recorded data becomes available through black-box type recorders. We proceed here to describe the ATT in that form, and to bring out the interesting points it reveals.

It bears stating that this departs from the original TT in two ways. First, passive interrogation removes any aspect of en-route interaction with the driver to be assessed, reducing the interrogator's role to that of mere observer. Second, this test could be implemented as a computerized classifier system, which offers interesting opportunities, but places the human even further out of the testing loop, so that now they only prescribe the statistical tests employed, with the rest of the test executable without human intervention.

To explore this version in some detail, in this variant of the ATT, the behavioral estimate of the player is made not on second-to-second unfolding observations, but upon the ensemble recording of their behaviors over a set of routes driven during the test run. This passively interrogated, after-the-fact analysis of the data recorded from a new driver is then compared to a larger set of recorded data from known human drivers driving under similar conditions. Such a comparison would help reveal the relationship of the newly recorded dataset to the pool of human data encapsulated by the larger multi-driver ensemble.

We note here that even if a newly recorded player's behavior differed significantly from that of other vehicles in the database of vehicles driven by humans, it would not necessarily mean that that new player was a machine. But, such a finding of significant difference form the behavior of the comparison set would certainly be worthy of further review, as it identifies a driver whose behavior doesn't fit the behaviors of the others in the ensemble to which it was compared. Obviously all proper statistical concerns must be observed to make sure that both the sample of new driver behavior and the ensemble of behavior to which it is compared are large enough and rich enough to provide statistical reliability in the estimates they provide. Interestingly, we see no reason why data recordings from the real world couldn't be just as valid as those collected in simulation.

Other variants

During considerations of the idea of the ATT, a number of other variants have come up.

One as yet poorly explored variant considers the inclusion of real vehicles driven by a player or interrogator via remote control. In both the vehicle-to-vehicle, and Road Test administrations of the ATT, this option could be included. This is interesting in particular because it raises a basic question. "Would the driving behavior be distinguishable between a human driving in a car as is normal today, and a human driving a real car via remote control?" Such a test adheres closely to the imitation game form of the ATT. But, more importantly, it gets directly at the heart of what it means to have your "skin in the game". We hypothesize that there will be discernable differences in behavior and interactions with other vehicles when the remote driver is safely ensconced away from the vehicle that is at risk.

If it were of interest, perhaps in a future where automated machines were the safer way to travel, another variant could be imagined, in which the passive interrogator version of the ATT is reversed to compare a human driver's recorded behavior data to a database of super-safe machine drivers. In that environment, this method might decide when the human driver's behavior differed little from super-safe machine drivers to be indistinguishable from them, making it safe to allow them to share the roads. Such a situation could also arise as part of the integration of human driven vehicles into environments that were established as isolated roadways initially built for the exclusive use of intelligent autonomous vehicles, as has been suggested by the earlier automated vehicle projects like those demonstrated by the DOT in the late 1990's.

V. What does this Automotive Turing Test (ATT) Actually Measure, and do we want that?

The measure of intelligence in the original TT [1] is actually quite interesting, as are the related measures coming out of the ATT variants described in this paper. In the original TT, the judgment made by the human interrogator is based upon their observations of the behaviors of the players. But, the behaviors of the players are actually based upon B's (likely) honest attempt to represent herself faithfully, and on the deceiver's (A's) ability to model, duplicate, and display B's behavior appropriately to deceive the interrogator into thinking that player A is in fact player B. The model that A builds could also be augmented to include a model of the expectations of the interrogator C, capturing what A expects C to look for in his attempt to recognize honest player B from deceiver A. This would allow deceiver A to exhibit that behavior first, or to prepare a counter behavior that either influences C's expectations of B or nullifies the effect of B's behavior (as in Turing's suggested example statement by B "I'm the woman", followed by A's comment "Don't listen to him, I'm the woman").

While human guile, and to some extent its duplication by a machine, are inherently part of the definition of human thinking in the original TT and imitation game, the other interpretations of the TT idea ease this requirement. Instead, they replace it with a demand for sufficient skill in a particular task or a set of tasks. This implies that outside of conversational games, guile is not necessarily the ultimate definition of intelligence. It is at this point that the Road Test version of the ATT comes into play, because for most people the goal of vehicle intelligence is not to take the risks that humans would, but rather to be as safe and effective on our roadways as humans are today, or more so. Thus, for the Road Test version of the ATT, we seek a vehicle that is *smart enough to be safe in situations where a human might not be*.

This differs substantially from the Vehicle-to-vehicle ATT variants, which would give us a measure of whether or not a human observing the behavior of our intelligent vehicle could reasonably have the same expectations of this vehicle's behavior that they would have of a human driver's. Given the number of accidents on our roadways every year, it is reasonable to ask even then whether this would be an acceptable standard. To add to the Vehicle-to-Vehicle variants the element of safety, we would need to require that our set of human players be limited to humans known to be notably good or safe drivers. This is a standard which, while obvious when grossly violated, may still be somewhat loosely defined currently and harder to assure than we might like.

So, if not absolute safety from an intelligent vehicle, what do we gain by using a purely human standard of intelligence? We may gain two important things: (1) a behavioral estimate of the driver we observe as matching a model of behavioral expectations which we build in our own mind, and (2) an estimate of how far we should trust that driver to actually use that model.

The creation of a model of the observed driver behaviors allows us to pre-plan actions to take if the observed driver either maintains their currently estimated course of action, or changes to one or the other of the predicted actions. This pre-computing and caching of solutions to more probable future situations is a valuable skill that allows faster identification of situation changes and faster reactions to those changing situations. This pre-computation is lost or significantly curtailed when encountering vehicles that drive in ways very different from the way we expect, as can be observed when driving for the first time in an unfamiliar location. This can be most striking when traveling in a land where cars drive on the opposite side of the road from a driver's previous experience. But, over time, through observation and mental modeling we reconstruct a model of the behavior of others around us. We can then incorporate their actions into our own repertoire and expectations. The inherent flexibility associated with this skill of learning and adapting is central to the aspects of intelligence that come to mind when we describe humans as smarter than a brittle artificial system that is otherwise highly trained but limited to a single fixed set of already-solved problems. But, underlying this aspect of intelligence, is the action of observing others, modeling their behavior, and adapting our own to be indistinguishable from theirs. This is the key measure tested by the TT and the Vehicle-to-Vehicle ATT, and this is what makes us safer in new situations, which is something we would like to see for intelligent vehicles.

If predictability similar to a human's is valuable because it allows us to prepare for upcoming situations, knowing the limit of that predictability is also valuable, but in a different way. One value is that, when we remember this limitation and explicitly include it in our thinking, it keeps us watching for the unexpected. This reduces our chances of becoming overconfident, and limits the risk to which we are willing to expose ourselves.

Another, and perhaps more important value of the limitation of the predictability of other vehicles, is that it creates a *social contract* between vehicles on the road to ensure their mutual safety. The expectation of some unpredictability in the behavior of others around us (the element of surprise mentioned in Section III), and a respect for the sovereign right of others to behave in ways that we might not have predicted, leads us to create buffer zones around vehicles that are larger than would be required if the behavior of other vehicles was perfectly predictable.

A good example illustrating the points above in everyday driving can be seen in severely foggy or snowy weather, when vehicles often close the gaps between themselves and the vehicle ahead and rely to a larger extent than usual on following the taillights of the vehicle in front of them. When a driver of the lead vehicle overestimates the predictability of the rest of the environment, that mistake can ripple back through the entire chain of vehicles, with each in turn overestimating the reliability of the actions of the vehicles and environment in front of them. These chains of unmet expectations in the reliability of other drivers may lead to much larger accidents than might occur if each driver adjusted their estimates of how far to trust the other drivers more appropriately to match the actual environmental conditions.

Concluding this section, we wish to touch upon the amount of intelligence required to pass Road Test variants of the ATT.

In general, it does not require a lot of conscious efforts for an experienced human driver to drive a vehicle safely, especially in a familiar environment, e.g., repetitive drives from home to work and back. Potentially significant mental efforts seem to be used only for complex navigational tasks in a busy traffic environment, or when a driver chooses to do several tasks simultaneously (e.g., talking on a cell phone while checking directions on the map and driving), likely at the expense of an elevated risk of an accident.

Sometimes a driver must exercise a quick and correct judgment and decision making to avoid an accident or minimize its severity. For example, if a child suddenly jumps out on the road in front of the vehicle, a driver must quickly execute a suitable avoidance maneuver (humans would do so instinctively, but unfortunately not always successfully). A driver might opt for driving in a ditch next to the road to avoid hitting the child.

When an animal suddenly appears on the road, drivers sometimes choose the “stay the course” behavior for their vehicles, as evident in many US states by the sight of dead animals lying on the road.

What if a ball suddenly appears in front of the moving vehicle? Does this mean that a child might follow the ball in the next second? This hints at the need for an intelligent vehicle to possess intelligence broader than what is immediately applicable to driving or simply distinguishing humans from animals. It is highly desirable that the intelligent vehicle have enough intelligence not only to react quickly but also to exceed any human driver in ability to avoid the collision with a human or an animal.

VI. Conclusion

In this paper we proposed several variants of the Automotive Turing Test (ATT) based on both the original imitation game version of the TT, and on subsequent extensions of the TT into domains of expert performance. We sketched out several implementations of the ATT for consideration and discussion on the topic. The first implementation is administered with only vehicle-to-vehicle interactions, while the second is administered “Road Test” style in a way very similar to student driver testing. These two administration styles and methods were described for implementation on both cars in the real world, and in a high fidelity simulation environment where risk to life, limb, or property can be removed. We discussed the existing value of testing for human levels of intelligence, including its inherent limitations and perhaps diverging goals. We distinguish it from a more typical automotive definition of vehicle intelligence, which uses the safety of vehicle occupants as a proxy for vehicle intelligence. We also pointed out how modeling human weaknesses within a system can actually strengthen the robustness of the system to protect against system perturbations or failures.

We acknowledge readily that there is a great deal more work to be done to instantiate the ATT, and to deliver intelligent vehicle driver systems and intelligent driver support systems that can compete with human drivers in the ATT. While focusing initially on human-like driving qualities, we point out how the ATT discussion might prove useful to target development of intermediate steps in the process of realizing intelligent driving systems in which accidents no longer occur. We also foresee opportunities through this work to explore the system of interacting human behaviors that take place on our roads, and to explore the true nature of human intelligence as it is reflected in vehicle control decisions.

Acknowledgement

SFK also thanks his family (DK, RK, & VK) for their support and understanding during the preparation of this paper. We also want to thank Michael Samples for useful comments about a draft of this paper.

References

[1] A.M. Turing, “Computational Intelligence and Machinery”, Mind, Vol LIX, No. 236, pp 433-460, 1950

[2] Dingus,T.A., Klauer,S.G., Neale,V.L., Petersen,A., Lee, S.E., Sudweeks,J., Perez, M. A., Hankey, J., Ramsey,D., Gupta,S., Bucher,C., Doerzaph,Z.R., Jermeland,J., and Knipling,R.R., “The 100-Car Naturalistic Driving Study, Phase II – Results of the 100-Car Field Experiment”, DOT HS 810 593, April 2006

[3] S.J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, p. 2, Prentice Hall, 2nd edition, Pearson Education Inc, Upper Saddle River NJ, 2003.

Autonomous Robots with Both Body and Behavior Self-Knowledge

B Brent Gordon

In the spirit of the conference theme, *the interplay between autonomy and intelligence,* the general direction of this paper is to describe how autonomous robots could begin to autonomously develop intelligence. (For us a robot is always a constructed, embodied, situated system.) Most of the discussion will take place at the abstract level of system architecture and design principles. For the sake of discussion we assume that the robots under consideration have a reactive or behavior-based system as part of their architecture. This is reasonable since it is true of a high proportion of autonomous robots built today.

1) Developmental robotics framework: The framework in which we will work is that of biologically motivated *developmental,* or *epigenetic,* robotics. Recall that developmental robotics is based on the idea of a (cognitive) robotic system developing knowledge of itself, its environment, and the dynamic activities of its world in (possibly overlapping and irregular) phases, analogously to the development of a human child [1] [2] [3]. In this approach learning is experiential—there is no alternative for animals and humans—and therefore the act of learning and creating of intelligence is, unavoidably, autonomous.

2) The biological analogy, sensing: For more insight we set up a biological analogy that takes into account some major components of a robot's, and human's, sensorimotor systems and software architecture. Clearly the external sensing on the robot should correspond to the human external senses of seeing, hearing, smell, and taste, even if seeing does contribute to proprioception. Human vestibular senses are calibrated against ambient gravity, so whether to consider them self-sensing or external depends on the circumstances. Touch is composed of several different kinds of nerve endings, among which we consider sensitivity to light, heavy, or sharp pressure, or to heat and cold, as external. Next, there is a good analogy between the *kinesthetic* sensors on the robot that measure motion and forces for moving parts and the human *somatic* nervous system, comprised of the nerves that run between the spinal cord and muscles and bones. Similarly, robotic sensors measuring power levels, fluid levels, temperature levels at different locations, etc., are roughly analogous to the nerves of the *autonomic* nervous system that run between the spinal cord and various internal organs.

3) The biological analogy, architecture: As for the software components, if the robot has a deliberative or cognitive component in its architecture, we will take that to be in analogy with human reasoning and planning, without particular concern for exactly where in the brain that is located, if indeed it is localized anywhere. From a behavioral perspective we will consider that, under the analogy, the behavior-based component should correspond to the movements we humans make all the time without thinking about them or directing any conscious attention to them—in other words, our learned habits. (The set of human behavioral reflex responses, which includes the vestibular-ocular reflex and the posture reflex among others, don't make as interesting or useful an analogy.) It turns out that most of our common and habitual movement is governed by the cerebellum. A useful and experimentally sound model is that the cerebellum contains multiple pairs of behavior models, where each pair consists of a forward, or predictive, and an inverse, or controller, model; and moreover, these behaviors can be combined together with weighting factors chosen according to how well their forward models match the new or desired behavior [4] [5] [6].

4) ***The Body and Behavior Self-Knowledge Design Principle*** *(BBSK):* Given a cognitive autonomous robot, increase, or make sure it is provided with a lot of, physical self-sensing. Then collect not only all the environmental and self-sensor readings but also the states and behaviors of each control element in the robot's behavior-based system, and make all this data available to robot's cognitive system in addition to logging it.

The idea to collect and log this data was presented by Doug Gage already last year [7], and it was around that same time that Aaron Sloman [8] gave me the idea of making it available to the robot's cognitive subsystem. To see in part why we have elevated these ideas to the status of a design principle, push BBSK across the biological analogy to the human side. Then the data being collected and provided to the brain is more than sufficient for *proprioception,* the knowledge available to us at all times of where all the parts of our bodies are and how they are moving, whether we consciously access that information or not. It also more than suffices for dynamic internal self-models. And insofar as it has been checked, proprioception and internal dynamic self-models are virtually universal across the animal kingdom; the latter, for example, have been confirmed in some insects [9]. In addition we can anticipate how BBSK should lead to a number of useful architectural features, some of which we describe below.

5) The value of logging: Originally [7] suggested that even just logging the data would provide users and developers "with hard data to support system adaptation and on which to base discrete product improvements," as well as substantially simplify developers' debugging problems. On these grounds and the declining cost of sensors, processing, and memory storage, he argued that the costs of implementing the BBSK principle even just to this extent (in our terminology) would be more than outweighed by the benefits.

6) Implementation substrate: To simplify the exposition from this point forward we will assume that the robot has been entirely behavior-based and that we are now adding to its architecture a suitably powerful cognitive component with plenty of learning capability. Even had the robotic system been constructed with a hybrid architecture from the start, there would still be a *Robotic Body-Mind Integration Problem* [10] since the cognitive component works in the symbolic language of logical reasoning while the behavior-based component works in the numerical language of control theory, and thus the two don't understand each other.

Next, as a thought experiment, for definiteness and to avoid complications in the exposition, imagine our robot is a multi-link arm with a dexterous manipulator mounted on a mobility platform, with whatever visual and other external sensors you like. Thus its physical design allows it to do much more than can be built into its behavior-based architecture, so it is a good candidate for taking the next step. So we add to it all manner of self-sensing, and alongside the old behavior-based system we install a powerful cognitive learning and reasoning package. Now the point is, in hardware and software we still have to set up the I/O from the sensors to the cognitive system, from the reactive system to the cognitive system, from the cognitive system to the reactive system, and from the cognitive system to low-level control, where its signals may override or be blended with those from the reactive system, perhaps. We henceforth assume all that has been handled.

Currently we believe that the input from the kinesthetic self-sensors should come into the cognitive system without any preset structure or hierarchy, in order to give the cognitive system itself the maximum flexibility. In particular, the cognitive system could then determine its own hierarchical indexing scheme, or have several, and change entries around, according to its needs and as those needs change. Of course real experiments or experience could ultimately suggest a different approach.

7) Multi-resolution, multi-perspective, dynamic, body self-model: One of the first and most significant outcomes is the feasibility of a more comprehensive and detailed explicit internal body self-model, i.e., of the robot's physical structure and how its parts move individually and together. Humans, for example, learn to know where the parts of their bodies are and how they are moving, and at any given moment one may consciously refer to this model focusing at any point on the body with any level of resolution, or may let it run in the background until some anomaly calls conscious attention to it. The basic idea is to implement something analogous in the robot.

8) Multi-level motor control: In a similar spirit as the previous paragraph, with a systematic increase in self-sensing throughout its body, the robot may be able to coordinate movements it could not previously. For example, new feedback loops, and hence finer motor control, may be available for some joints. Or, because sensory data from everywhere is centralized, it may become possible to coordinate the movements of parts whose motion could not previously be coordinated. The idea is that these new coordinated behaviors would supplement, not replace, the coordinations that were already present in the behavior-based system.

9) Behavior model: In the meanwhile, the cognitive system potentially has a complete model of the behavior-based system, and could be able to follow or even anticipate the behavior-based system's responses. This should provide yet another layer of supervisory control, for added overall system robustness.

10) Integrating new habits with old: One of the things humans can do is learn a new behavioral skill, such as walking or driving a car, and practice it until its proper execution requires a minimum of direct conscious cognitive attention. At that point, and in the case of basic physical coordination and competencies, only then, it becomes possible to learn something else or develop cognitively while performing that behavior. According to our analogy something very similar should be true for our robot. While the robot's behavior is being controlled by its behavior-based system, the cognitive system is freed up to learn or develop or do whatever. While the cognitive system is controlling the behavior, such as when the robot is learning a new skill, then the cognitive system is not free to develop. Thus, as new behavioral patterns are learned and mastered, they need to be saved as habits. In practice this will probably not be as delicate a question as that of how to integrate them with the existing behavior patterns of the reactive system, as this issue, while not identical with the Robotic Body-Mind Integration Problem mentioned earlier (paragraph -.6), skirts very close to it.

11) Interplay between development and learning: To come full circle, and return back to the ideas in paragraph -.1, we suggest that the intricate relationship between development and learning, apparently a topic of great debate in human development circles [1], provides a good mirror for questions about the interaction between autonomy and intelligence. Rephrasing some of the major views in the development-learning debate, maybe intelligence takes place in a context set up by autonomy (Piagetian), *cf.* [2]. Or maybe the two are mutually coupled, so that intelligence advances autonomy, and autonomy enables, limits, or triggers intelligence (dynamical systems), *cf.* [11]. Or maybe there is no real boundary between autonomy and intelligence at all when we properly consider the dynamics and subsystems at all scales (the complex systems view), *cf.* [12] for a biological study. Perhaps all of these views are useful in different contexts.

References

[1] Lungarella, M., et al., *Developmental robotics: a survey.* Connection Science **15** (2003), 151–190.

[2] Metta, G., et al., *Development and robotics.* In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots* (Tokyo, Japan), 2001, pp. 33–42.

[3] Asada, M., et al., *Cognitive developmental robotics as a new paradigm for the design of humanoid robots.* Robotics and Autonomous Systems **37** (2001), 185–193.

[4] Wolpert, D.M., Kawato, M., *Multiple paired forward and inverse models for motor control.* Neural Networks **11** (1998), 1317–1329.

[5] Kawato, M., *Internal models for motor control and trajectory planning.* Current Opinion in Neurobiology **9** (1999), 718–727.

[6] Haruno, M., Wolpert, D.M., Kawato, M., *MOSAIC Model for sensorimotor learning and control.* Neural Computation **13** (2001), 2201–2220.

[7] Gage, D.W., *Meaninful metrics and evaluation of embodied, situated, and taskable systems.* In *Performance Standards for Intelligent Systems Workshop* (Gaithersburg, Maryland, USA), 2006, pp. 52–53.

[8] Sloman, A., personal communication.

[9] Schomaker, L., *Anticipation in cybernetic systems: a case against mindless anti-representationalism.* In *2004 International Conference on Systems, Man and Cybernetics* vol. 2, pp. 2037–2045.

[10] Kawamura, K., Dodd, W., Ratanaswasd, P., *Robotic body-mind integration: next grand challenge in robotics.* In *Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication* (Kurashiki, Okayama, Japan), 2004, pp. 23–28.

[11] Kuhl, P., *Language, mind, and brain: experience alters perception.* In *The New Cognitive Neurosciences,* M.S. Gazzaniga, ed., 2000, pp. 99–115.

[12] Thelen, E., and Smith, L., *A Dynamic Systems Approach to the Development of Cognition and Action.* Cambridge, MA, MIT Press (A Bradford Book), 1994.

A Cognitive-based Agent Architecture for Autonomous Situation Analysis

Gary Berg-Cross
EM&I
Gary.berg-cross@em-i.com

Wai-Tat Fu
University of Illinois
wfu@uiuc.edu

Augustine Kwon
ICF International
akwon@icfi.com

Abstract

We discuss the start of a project investigating a cognitive basis for intelligent agents that can approach the problem of situation understanding. We suggest that a practical system of intelligent agents can be build adapting existing agent modeling frameworks, ontologies from semantic web technology as well as a reasonable situation domain models. These can be brought together with a suitable cognitive architecture ACT-R which could be used to provide key roles in more human like situational awareness capability in emergency and disaster operations, especially where sensor information is harvested from semantically heterogeneous data sources. Existing situational ontologies and vocabularies can be supplemented by using DOLCE's formal ontology. This serves as a metalevel ontology that can relate different ontology modules and can generate new categories to extend an ontology (by agent learning) as needed. semantically-rich, conceptual level representations of real-world events. A Descriptions & Situations ontology provides a theory of ontological contexts capable of describing various types of context including non- physical situations, plans, beliefs, as entities so they can be communicated and understood between agents. We believe our system architecture provides a relatively good built-in infrastructure to meet fairly rigorous performance measurement requirements and has general applicability in a wide variety of situations.

Keywords: cognitive models, disaster situations, information fusion, intelligent agents, ontologies, situation understanding

1. Introduction

As witnessed by the diversity of papers in past PerMIS conferences no single technique or tool available to build/develop intelligent systems (IS)/ agents has proven adequate to address all the functionality desired. This is true even for even relatively simple software information agents using the meaning of the data for information sharing such as envisioned in the original Semantic Web (SW) concept . Currently, there are multiple views on a suitable architecture and the nature of the knowledge needed to develop successful SW agents. Moreover, agent-based information integration, as typically discussed for the Semantic Web is not the only type of information fusion being actively researched. Sensors provide systems access to real-time (or near real-time) streams of actual, low-level events. These serve as input to other agents which structure this data and integrate it with higher level concepts. Such sensor-based processing is used in many domains, including disaster response, crisis management, modern battlefield operations and health monitoring. These situations are characterized by multiple, distributed heterogeneous information sources, and rapidly changing situations that may include mobile agents/objects. Special agent capabilities are needed because situations involve a large number of inter-dependent, dynamic objects that change their states in time and space, and engage each other in fairly complex relations. As a result, required intelligent systems capabilities include effective methods for situation recognition, prediction, and reasoning activities. Collectively these capabilities have been called situation management (Jakobson et al 2006). Research to build such situation understanding agents requires processing dynamic situations using complex cognitive modeling, design and population of formal situation ontologies, collection and fusion of sensor. Current agent systems still have difficulty accommodating things like diverse spatiotemporal information, within a single analytic context in a suitable period of time. Yet as part of analytic process for understanding situation humans easily integrate both quantitative and qualitative information assessments to quickly arrive at analytic conclusions. The discrepancy may in part be in part due to the duality behind human cognitive architecture. The dual processing theory (Evans, 2008) distinguishes between cognitive processes that are:

- unconscious, rapid, automatic and high capacity, and
- those that are conscious, slow and deliberative.

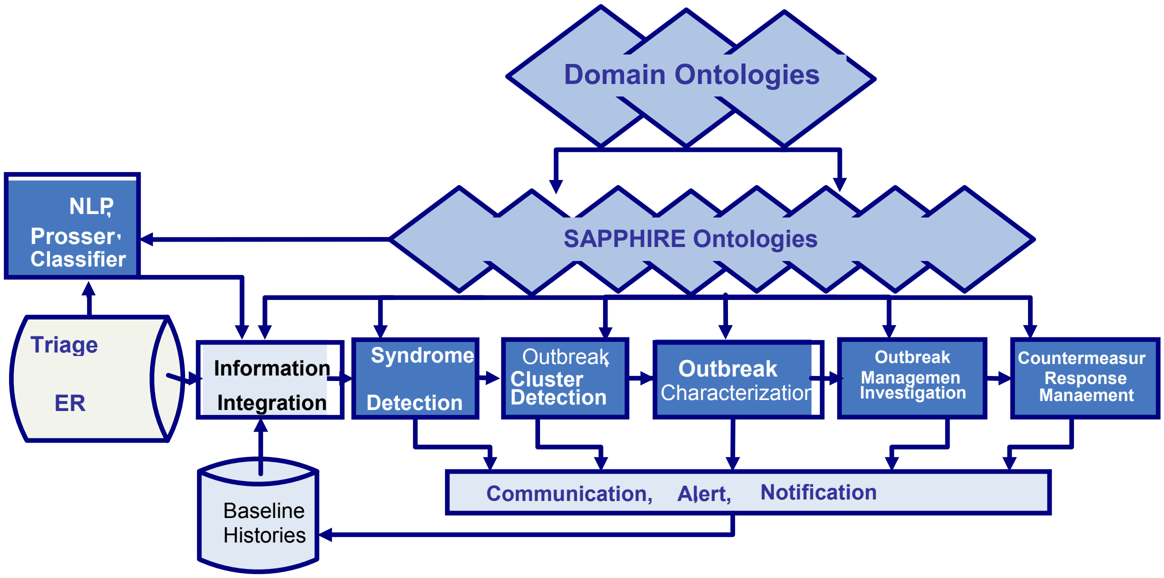
This characterizes human reasoning as a robust interplay between an easily believed perception-based system and a more cognitively demanding logic-based reasoning system. This views intelligence as a developed phenomena that balances multiple reasoning mechanisms, together with scruffy modules of knowledge which learn to deal with situations that are only partial predictability, due to dynamics, and the absence of precisely defined states. Berg-Cross (2004, 2006) has suggested that a multi-level hybrid architecture, based on a cognitively realistic foundation, could approximate human performance for this class of problems. To be practical such an architecture would build on the existing agent models, semantic web technology and standards, as well as a reasonably adequate knowledge and domain models. The direction proposed herein is an architecture that leverages recent advances in machine learning, distributed agent technology and semantic representation brought together within a suitable cognitive architecture - ACT-R. ACT-R is particularly suitable for performance measurements of intelligent capability because it has been widely shown to be capable of equally human performance on a range of task and match human learning improvement profiles over time. This paper discusses an ACT-R architecture designed to learn and perform aspects of situation assessment. Our discussion of work is divided into three parts. First we describe what is understood about situation understanding cognitively and the ingredients of a situational ontology is described. Simple domain ontologies can enhanced by leveraging some foundational ontology and their modules to formalize concepts like "Participation". Framing moels like Description of Situations can be used to model how knowledge of information can shared by agents. Second, the ACT-R architecture is described showing how it might be populated and training on situational understanding tasks. A third and final section summarizes the feasibility of the approach and describes future research and development plans using specific types of situations.

2. Situational Knowledge and & Ontologies

As intelligent agents ourselves we generally understand the idea of "situations" and an agent with such a comprehension can be said to have "situational awareness" (SAW). A rational empirical approach to SAW & understanding is general defined with three sequential components: (1) perception/awareness of elements/objects in the environment within a volume of time and space, (2) along with a comprehension of their functional nature and organizational relationships (their "meaning") (3) as well as an ability to go beyond SAW to project the status & relations of situated objects in the near term as an empirical test of "expectations". A top-down, rational model of SAW incorporates an agent's goals & objectives into its reasoning about events, relations and situations. This helps upper-level agents reduce the number of possible relations definable within an agent's knowledge to constrain situational possibilities. By knowing something about what is expected, attention on relevant events and relations can improve agent operation (Matheus et al. 2005).

Situation/context-aware systems have been proposed as an important class of applications and an important step towards ubiquitous computing. Examples of such agent systems described at the 1st International Workshop on Agent Technology for Disaster Management (Nicholas et al., 2006) include discussion of agent architectures to handle coordination via intents, multi-agent learning to support urban planning, and training based on agent-based situation simulation. Often such work involves sensor-based data being fused into situational information. Typical architectures are multi-agent with sensor agents responsible for an initial degree of processing and higher agents responsible for associating these inputs with object concepts with still "higher" agents responsible for assembly into a "situations". Such systems assemble operating pictures using precision geospatial environment information layers (modifiable digital overlays) that can support decision making based on the detailed "knowledge" shared by the agents sensing a physical environment. An example of a such a system called SAPPHIRE (Situation Awareness and Preparedness for Public Health Incidents using Reasoning Engines) shown below dealing with Public Health Incidents (Mirhaji & Coyne 2007). Pollution sensing includes CO,SO2,H2S,NO etc. with a half dozen meteorological factors and chromatography data such as ethane, ethylene etc. Systems like SAPPHIRE are not limited to direct sensor feeds, and can include entire reports of other agent's processing (hence NLP capability as shown in the Figure). In such architectures agents at higher levels in the "network" may have more responsibility in that they may use more elaborate communication protocols, taking into account and monitoring the information provided by lower level sensor agents. Several things are needed to make such systems effective. For sensor data fusion, common sensor standards are needed to create a common sensor data model. But to be able to handle conflicting information from sensors, higher agents might need a capability for incremental, flexible perceptual/ conceptual learning, which is feasible through retrieval of relevant memories. A common model is that situations are a high-order knowledge type of concept that are formed using existing concepts. We assume that situation knowledge is formed by an agent's interaction history with the environment and that agents can form situation "concepts" by "observing" that certain patterns of sequence of inputs from the environment . Thus for agent learning evaluative feedback should follow a certain action/class of actions, or the next input event from the environment given the current action. But beyond this proper knowledge is needed to support agent reasoning. Ontologies are used as part of the SW thrust into intelligent agents to define vocabularies so queries and assertions can be exchanged among

agents (Heflin, 2003). More recently ontologies understanding (Kokar et al, 2004). We illustrate the use for two levels of ontologies for understanding and representing situational knowledge – a mid-model of situations and a more foundational model that captures more of the event aspects of participation in situations as well as the relationship between descriptions and situations as meta-knowledge which can be used by agents. The SAW "model" (Matheus et al 2003) is a "light" ontology capturing the core elements and relations of a rational agents view of situations, as shown in Figure 2. In the SAW ontology model there are primary classes: SituationObject, PhysicalObject, and Events. The organizing point in the ontology and resulting models is



is defined as a relationship to 3 things: Goals, SituationObjectsharc and Relations. Situation Objects are entities in a situation that or c participate in Relations and can have characteristics (i.e., Attributes). These Attributes define values of specific object characteristics, such as expected/unexpected, weight or color. SituationObjects may be PhysicalObject (a sub-type) with Volume, Position and Velocity.

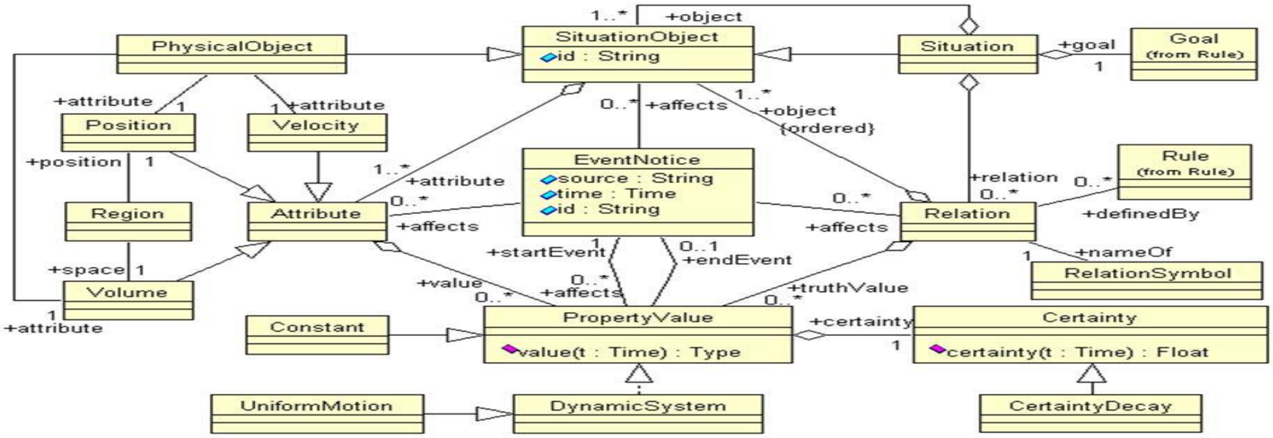


Figure 2. Core Concepts in the SAW Ontology (Matheus et al, 2003)

Relations are used in a structuring geometrical (not well represented in SAW) and positional aspects of these concepts. For example, to represent bus transportation systems for flood situations being reported on by environmental sensors, we need both realistic street information that can be overlayed by bus stops and also more abstract timetables along with timetables and circumstances that apply in emergencies as sensors report water height, closed streets etc. This would allow us to represent situations about stop on a bus route that is "near" the intersection of two streets. However, in event of a flood this information will be supplemented by elevation information and perhaps closeness to streams. Such features are not typically captured in geospatial data bases and processable by GIS functions. In our architecture this information would be integrated by a situational agent using components of information from sensor agents and also making use of geospatial repositories of information. To handle these requirements the SAW ontology can be improved by grounding it in a more foundational ontology like DOLCE developed within the WonderWeb Project (EU FP5). DOLCE is a cognitively based, "reference" ontology, consisting of about 30 classes, 80 properties and many more axioms. It is designed to provide a sufficiently neutral base to map, integrate, and build domain ontologies, such as an improved SAW ontology. DOLCE includes the idea a high-level participation pattern of objects taking part in the Events on the SAW model. DOLCE conceptualizes endurants (Objects or Substances) and perdurants (Events, States, or Processes) as distinct types linked by the relation of "participation". Participation patterns help us understand the structure of repeating events that occur for types of situation. As shown time in Figure 3 indexing is provided by the temporal location of the event at a *time interval* duration, while the respective spatial location at a *space region* is provided by the participating object. The general pattern in Figure 3 uses an extended version of UML which can be converted to the DOLCE-time-plus (Gangemi et al,

2004) light ontology and is intuitively applicable to situations of interest to us including disasters, and health monitoring. Particularly nice for representing the knowledge that agents need to fuse information into situations is DOLCE's use of an Information-object design schema/patter as shown in Figure 3. This model formalizes how descriptions, which one agent may receive from another, serve as Descriptions for Situations. This extension to the DOLCE foundational ontology, is called the Descriptions and Situations Ontology (D&S) and can be used to define agent workflow - how the information in messages can be used by an actor who play a specific role. The D&S ontology (Gangemi & Mika, 2003) is based on a conceptualization that supports a first-order manipulation of *descriptive objects* (such as clinical plans, evacuation routes, emergency plans, institutions, etc.) in effect, theories and *situations* (such as cases, facts, settings)[1]. D&S's explicitly committed conceptualization is a distinction between an *unstructured world* or *context*, and an *intentionality* (description) that recognizes (some say constructs) a *structure* (situation) in that world or context. One nice thing about the commitment using D&S is that it supports organizing domain theories for areas like disasters & healthcare into different ontologies as well as into different *descriptions* or *situations*. For example, "a flood situation" is a disaster entity whose conceptualization is realized in several modules - disaster, transportation, hydrology, geography etc. Finally the DOLCE models can serve as modular, meta-level ontologies that can relate different ontology modules and can generate new categories to extend an ontology (by agent learning) as needed.

3.Rationale for ACT-R

ACT-R was picked based on many well known technical merits, including its activation-based rational action selection processes that closely resemble a human cognition process, along with various domain models. Under the proposed multi-agent based framework, the primary functionality of individual agents can be determined by one or more plugged-in ACT-R models. This can use ontology translators similar to those proposed by Wray et al. (2004) to allow various ACT-R models and their host agents to share domain as along with inference knowledge captured in various ontologlies, A system of different intelligent agents involved in various steps of the SAW process can work collectively to achieve a common system goal. Each ACT-R model has its own set of knowledge representation, input/output modules that allow it to interact with the external world as well as with other agents/models, and has its own learning mechanisms that allow it to adapt to the new situations and environmental conditions. Since ACT-R is developed based on the principle that knowledge are always rationally deployed to decide on the next set of actions, each model can behave rationally based on the existing knowledge it has. For example, ACT-R has a built in Bayesian learning mechanism that allows it to retrieve the relevant knowledge structure based on its need probabilities at a particular context of situations (Anderson & Lebiere,1998), as well as a reinforcement-like learning mechanism that learns to adapt to the statistical structures of the environment so that actions that have led to successful outcomes before will more likely be selected in the future (Fu & Anderson, 2006). These sophisticated mechanisms allow each ACT-R model to gradually adapt and learn the skills and knowledge required for different situations.

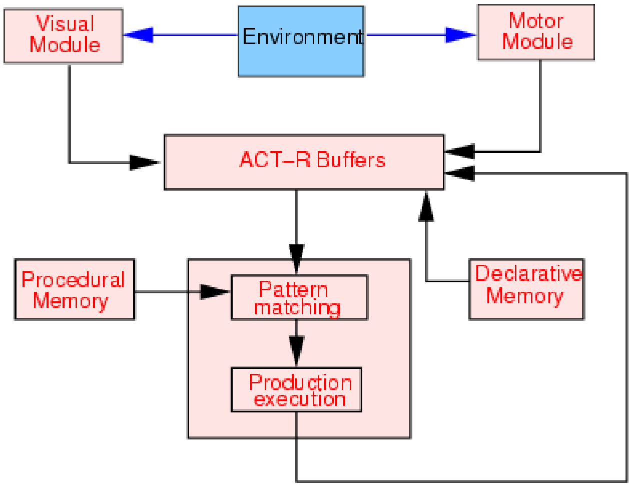
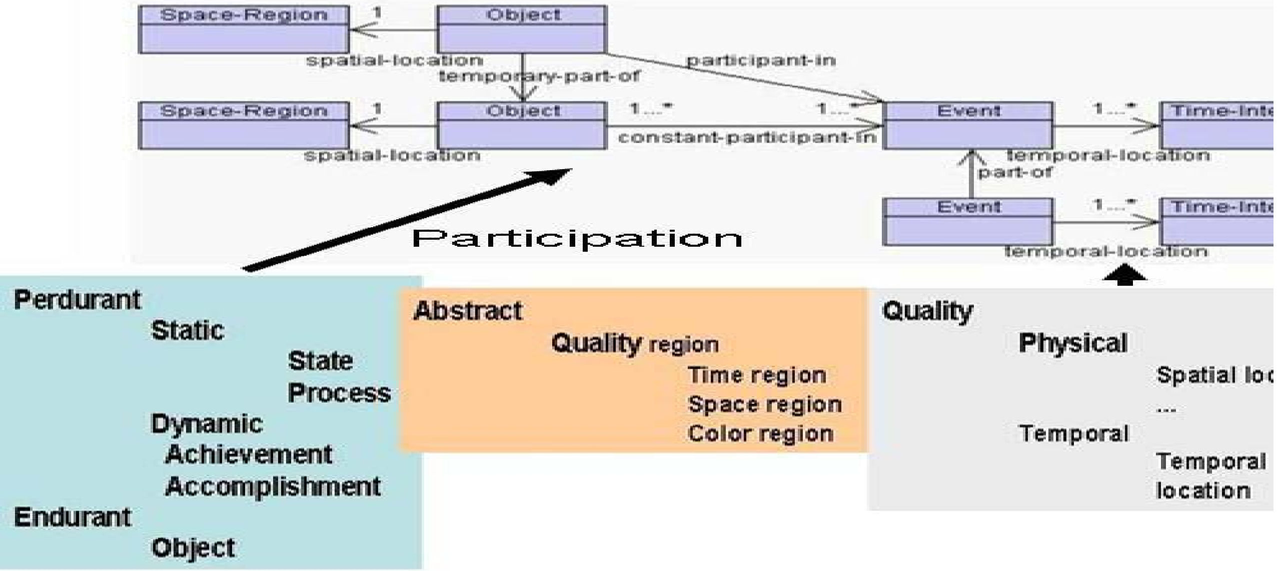


Figure 4 Individual ACT-R Architecture

Another important advantage of the ACT-R architecture is its ability to span components of cognition that have traditionally been treated as separate in cognitive psychology. For instance, a model of ISR analysis will likely involve reading and language processing, spatial processing, memory, problem solving, reasoning, and skill execution and acquisition. ACT-R not only has a generic knowledge representation across these different components of cognition, but it also specifies how these components are integrated to produce behavior. At its lowest level, ACT-R has both a spreading-activation mechanism to predict accessibility of declarative knowledge and a reinforcement-learning mechanism to predict the future success of certain actions. For example, knowledge that is needed frequently or repetitive actions with certain outcomes will eventually lead to skilled behavior that can be deployed with little cognitive resources. At the middle level, deliberate acts such as attending to relevant parts of the environment or pressing the right key on a device requires intelligent integration of multiple sources of information and background knowledge to generate intelligent behavior. To

[1] When D&S plugged into DOLCE it results in "DOLCE+" with description being a non-physical endurant. A situation is added as a top level.

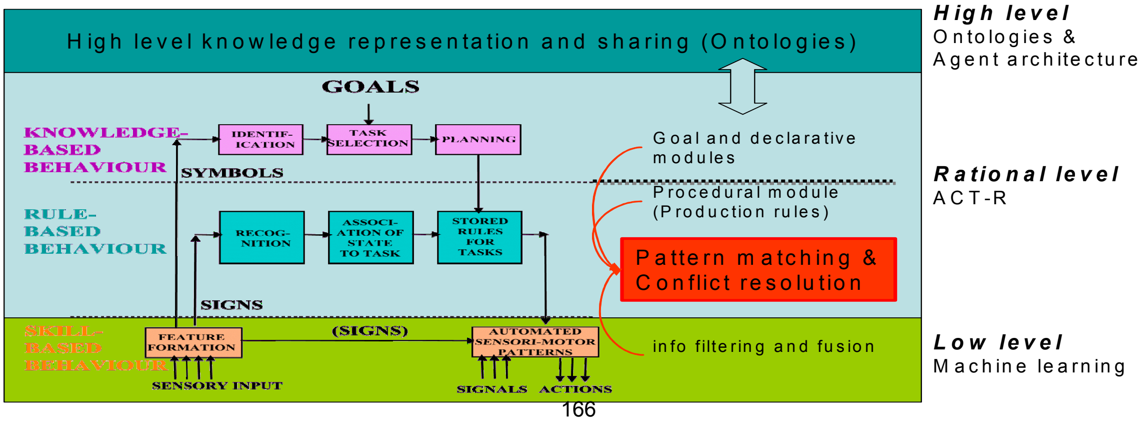


A

Figure 3 DOLCE's (A)Example of Situation-Descript

this end, ACT-R has a set of "subsymbolic" mechanisms that arbitrate how various sources of information should be integrated, and how and what actions should be selected and executed at different situations. Mechanisms at this level are found to be critical to various training goals. For example, work on computer-generated forces (Pew & Mavor, 1998) shows that training for people interacting with and against synthetic partners is effective only when these agents perform elementary actions like real people. Similar results were obtained by Jones et al. (1999), who show that training is effective only when synthetic pilots make turns with the timing of real pilots. A cognitive-based agent is therefore essential to ensure that the simulated environments appear "real" during training of operators. At the highest level, long-term knowledge are stored as a large set of declarative memory elements and procedural rules. This set of knowledge can obtained through a diverse set of training scenarios in various situation analysis environments. Taken together the direction proposed here is a multi-level approach that leverages our understanding of cognitive agent architecture in integrating three levels of information processing behavior as shown in Figure 5. In the high level a distributed agent architecture such as Cougaar (2007) and foundational ontology such as DOLCE will likely provide a means to process high level situational knowledge that may requires immediate attentions. At the rational level ACT-R will act as a bridge to connect high level and low level situational information processing behaviors through its proven strength in pattern matching and conflict resolution. At the low level an unconscious behavior such as machine learning will help to transform sensor fed raw data to ongoing situational knowledge through data filtering and fusion. Under the proposed hybrid approach, the anticipated major contribution of ACT-R will likely come from a

Figure 5 Three levels of Behavior


High level knowledge representation and sharing (Ontologies)
GOALS
KNOWLEDGE-BASED BEHAVIOUR
IDENTIF-ICATION
TASK SELECTION
PLANNING
SYMBOLS
RULE-BASED BEHAVIOUR
RECOG-NITION
ASSOCI-ATION OF STATE TO TASK
STORED RULES FOR TASKS
SIGNS
SKILL-BASED BEHAVIOUR
FEATURE FORMATION
(SIGNS)
AUTOMATED SENSORI-MOTOR PATTERNS
SENSORY INPUT
SIGNALS
ACTIONS
Goal and declarative modules
Procedural module (Production rules)
Pattern matching & Conflict resolution
info filtering and fusion
High level
Ontologies & Agent architecture
Rational level
ACT-R
Low level
Machine learning

rational level where most rule based behaviors through a human-like cognitive capability take place.
One of our research goals is to capitalize on the success of ACT-R in simulating the rational/adaptive nature of human information processing to coordinate activities in low level information fusion/selection and high level semantic ontological reasoning to support distributed decision making process in autonomous situation analysis.

4. Summary and Future Research

In this paper we presented a multi-level approach to cognitive agent situational understanding and awareness. The first level of performance analysis helps to understand cognitive criteria underlying success with SAW and pointed out potentially problematic areas and real-time issues with agent knowledge which can be addressed by improved ontology.
Performance measurement of a cognitive based distributed multi-agent systems (MAS) offers unique challenges that must be addressed explicitly in its agent infrastructure. A study done by Helsinger et al. (2003) shows that Cougaar's system architecture already provides a relatively good built-in infrastructure to meet fairly rigorous performance measurement requirements and a unique ability to use such data to adapt to environmental changes. The challenge and future research in our proposed solution is how to expand the Cougaar's built-in performance measurement capability to tightly integrate various ACT-R based cognitive plug-ins models to provide a more powerful and flexible autonomous situation analysis platform.

References

Anderson, J. R. & Lebiere, C. (1998). The atomic components of thought. Mahwah, NJ: Erlbaum.

Berg-Cross, Gary. A Pragmatic Approach to Discussing Intelligence in Systems, Performance Metrics for Intelligent Systems (PerMIS) conference 2004.

Berg-Cross, Gary Developing Knowledge for Intelligent Agents: Exploring Parallels in Ontological Analysis and Epigenetic Robotics, (invited paper) Performance Metrics for Intelligent Systems (PerMIS) conference 2006

Cougaar (2207) http:// cougaar.org/

EU FP5 WonderWeb project (http://wonderweb.semanticweb.org) by the Laboratory for Applied Ontology (http://www.loa-cnr.it.)

Evans, Jonathan "DUAL-PROCESSING ACCOUNTS OF REASONING, JUDGMENT AND SOCIAL COGNITION" in Annual Review of Psychology (2008, in press)

Fu, W-T. & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. Journal of Experimental Psychology: General, 135(2), 184-206.

Helsinger, A., R. Lazarus, W. Wright and J. Zinky (2003). Tools and techniques for performance measurement of large distributed multiagent systems. the second international joint conference on Autonomous agents and multiagent systems Melbourne, Australia ACM Press

Jakobson, Gabriel, Buford, John and Lewis, Lundy **A Framework of Cognitive Situation Modeling and Recognition Military Communications Conference,** 2006. MILCOM 2006, Washington, DC, Altusys Corp., Princeton, NJ, Oct. 2006

Nicholas R,. Jennings, Milind Tambe, Toru Ishida, Sarvapali D. Ramchurn, First International Workshop on Agent Technology for Disaster Management, Hakodate, Japan 8th May 2006

M. Kokar, C. Matheus, K. Baclawski, J. Letkowski, M. Hinman, J. Salerno, ***Use Cases for Ontologies in Information Fusion***. In Proceedings of FUSION'04, Stockholm, Sweden, pages 415-422, June 2004.

C. Matheus, M. Kokar, K. Baclawski, J. Letkowski, C. Call, M. Hinman, J. Salerno and D. Boulware, "SAWA: An Assistant for Higher-Level Fusion and Situation Awareness", In Proc. SPIE Conference on Multisensor, Multisource Information Fusion, pages 75-85. (2005)

Parsa Mirhaji, MD & Robert Coyne, The Semantic Web And Health Information Systems, SICoP Conference 2 (April 25 2007)

Pew, R. W., & Mavor, A. S. (1998). *Modeling human and organizational behavior: application to military simulations*. Washington, DC: National Academy Press.

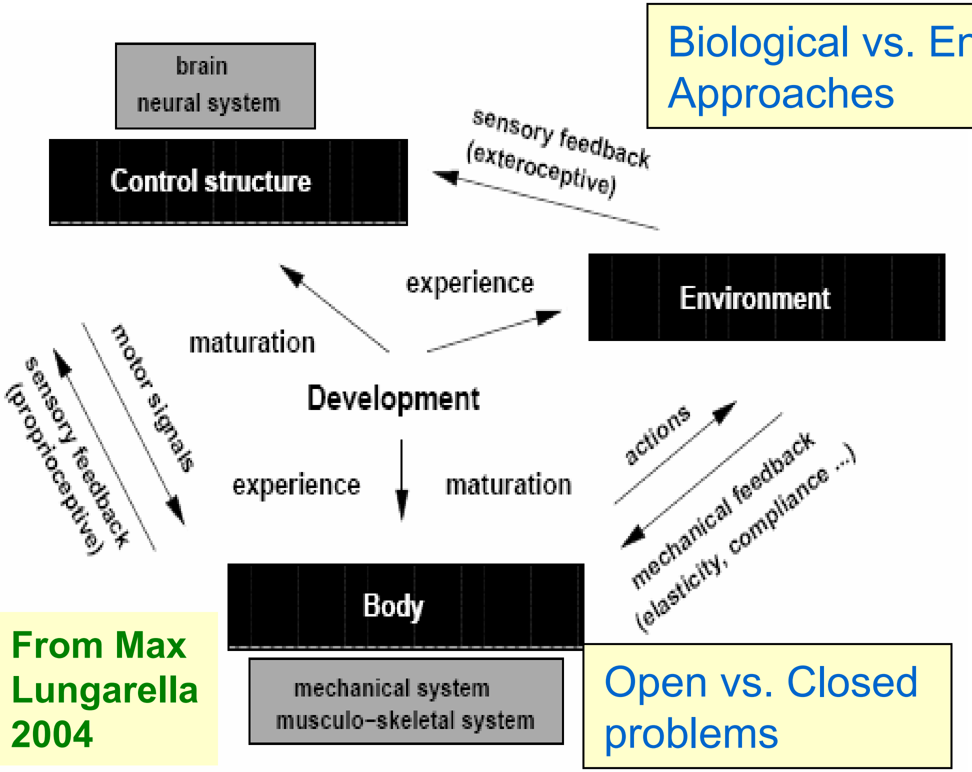
Wray, R. E., Lisse, S., & Beard J., ***Ontology infrastructure for execution-oriented autonomous agents.*** Knowledge Engineering and Ontologies for Autonomous Systems 2004 AAAI Spring Symposium Volume 49, Issues 1-2, 30 November 2004, Pages 113-122

Can the Development of Intelligent Robots be Benchmarked? Concepts & Issues from Developmental/Epigenetic Robotics

Lisa Meeden (Swarthmore College); Douglas Blank (Bryn Mawr College); James Marshall (Sarah Lawrence College); Odest Chadwicke (Chad) Jenkins; (Brown University); Charles C. Kemp (Georgia Tech); Gary Berg-Cross (EM&I) Organizer

Biological vs. Engineering Approaches

Challenges of Autonomy



From Max Lungarella 2004

Open vs. Closed problems

Different Concepts of Intelligence, Cognition and Rationality

Intelligence Metric should be Adaptive
And cannot really be evaluated by a fixed rule. Rather it ought to be a collective index reflecting the overall performance of the evaluated system on a variety of situations.
Intelligence as a Social Characteristic.., an isolated system.. is not of great interest. Measuring intelligence should be performed in the context of a group/society that includes other systems. Rong Gao and Lefteri H. Tsoukalas (2002)

More generally then, intelligence can be defined as "the ability for a system to adapt its behavior to meet its goals in a range of environments." …..Taking this cue from nature, it is reasonable to assess the intelligence capability of a machine that evolves solutions to problems in a manner similar to that of evolving phyletic lines in the natural environment. Fogel (2002)

On communicating with semantic machines

- The problem of designing …. semantic machines has been intractable because brains and machines work on very different principles.
- A solution to the problem is to describe how brains create meaning and then express it in information by making a symbol as a representation to another brain in pairwise communication.
- Understanding of the neurodynamics by which brains create meaning may enable engineers to build devices with which they can communicate pairwise, as they do now with colleagues, though not with words, but with shared actions.

Why do brains work this way? Animals and humans survive and flourish in an infinitely complex world despite having finite brains. Their mode of coping is to construct hypotheses in the form of neural activity patterns and test them by movements into the environment. All that they can know is the hypotheses they have constructed, tested, and either accepted or rejected. The same limitation is currently encountered in the failure of machines to function in environments that are not circumscribed and drastically reduced in complexity from the real world. Truly flexible and adaptive intelligence operating in realistic environments cannot flourish without meaning.

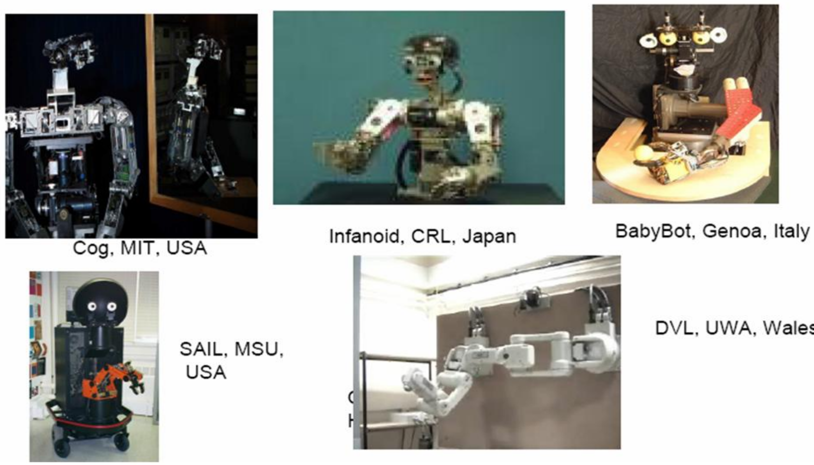
(Walter J Freeman) PerMIS 2003

Can the Development of Intelligent Robots be Benchmarked? Concepts & Issues from Developmental/Epigenetic Robotics

Lisa Meeden (Swarthmore College); Douglas Blank (Bryn Mawr College);
James Marshall (Sarah Lawrence College); Odest Chadwicke (Chad) Jenkins; (Brown University); Charles C. Kemp (Georgia Tech); Gary Berg-Cross (EM&I) Organizer

- Developmental robotics is a newly emerging interdisciplinary field that studies how autonomous robots can learn to acquire behavior & knowledge on their own, strictly through their interactions with the surrounding environment.
- What is our understanding of how innate mechanisms for abstraction, prediction, and self-motivation can be realized in such autonomous systems?
- What aspects of autonomous robot behavior should be preprogrammed as fixed policies and what aspects should be considered latent variable to be learned and adapted over time?
- How can the field reach consensus on the methods by which developmental robotic research should be evaluated, and progress benchmarked, and what is the relationship of developmental robotics to autonomous robot manipulation?

Examples of systems used in robotic developmental learning



Victorious robot makers gather (clockwise from top left): Jane Ng '01, Laura Brown '00, Seth Olthfski '00, Eli Silk '01, Jordan Wales '01, Assistant Professor of Computer Science Lisa Meeden, Nii Addo '00, and Assista Professor of Engineering Bruce Maxwell

Our Speakers & Their "Developments"

Overview of Session

Introduction: Gary Berg-Cross (EM&I) Session Organizer

Thematic Presentations (about 25 minutes each including questions)

- Overview of the developmental robotics field and its issues: Lisa Meeden, Swarthmore College.
- Self-motivation & How Innate Mechanisms for Abstraction, Prediction, and Self-motivation can be Realized in Autonomous Systems: Douglas Blank (Bryn Mawr College) & James Marshall (Sarah Lawrence College)
- Innate and Adaptive Behavior in Lifelong Robot Learning: Odest Chadwicke (Chad) Jenkins Brown University
- Can Developmental Robots Meet Real Human Needs?: Charles C. Kemp, Georgia Tech

Panel Discussion 45 minutes (Moderator Gary Berg-Cross)

Topics for discussion may include:

- Can developmental robotics meet real needs?
- How can the reliability of such applications be assured?
- Can a developmental approach out-perform existing approaches?
- What are reasonable milestones for the developmental robotics field?
- How will we know when the field has made progress towards its goals?
- Can standardized platforms be created?
- Are social situations necessary to enable the developmental process?
  - If so, what types of human/robot interactions will be needed?

Overview of Developmental Robotics

Lisa Meeden
Swarthmore College

What is developmental robotics?

- Interdisciplinary approach at the intersection of developmental biology, developmental psychology, neuroscience, AI and robotics
- Inspired by the fact that most complex biological organisms undergo an extended period of development before reaching their adult form and capabilities
- Rather than building robots to perform specific, pre-defined tasks, developmental robotics seeks to create open-ended, autonomous learning systems that continually adapt to their environment

You can only learn what you almost already know

- Machine learning systems work by taking small steps and building on what is already known
- However, it has proven difficult to move very far from the starting point
- Under a developmental process, a system can continually advance what it knows by placing itself into situations where it almost knows something, and then learning it
- Applied repeatedly, such a developmental process can potentially lead to much more complex, general-purpose behavior than has been achieved to date

Goals of the field

- Seeks to instantiate and investigate biological and psychological models by building robots (primarily the focus of **Epigenetic Robotics**)
- Seeks to design better robots by applying insights from developmental biology and psychology

Some origins of the field

- Drescher (1991), *Made up minds: A constructivist approach to Artificial Intelligence*
- Elman, Bates, Johnson, Karmiloff-Smith, Parisi and Plunkett (1996), *Rethinking innateness: A connectionist perspective on development*
- Ferrell & Kemp (1996), *An Ontogenetic Perspective to Scaling Sensorimotor Intelligence*

Origins (continued)

- April 2000, Workshop on Development and Learning, funded by NSF and DARPA
- October 2000, White paper suggesting a new initiative on Autonomous Mental Development submitted to NSF, NIH, and DARPA
- September 2001, First Epigenetic Robotics (EpiRob) Conference held (2007 conference at Rutgers next month)
- June 2002, First regularly scheduled International Conference on Development and Learning (ICDL)

Combines aspects of many previous approaches

- Embodied intelligence (Braitenberg, Brooks)
- Evolutionary robotics (Nolfi, Floreano)
- Lifelong learning (Thrun, Jenkins)
- Machine learning, especially reinforcement learning (Barto)

Distinctive aspects of developmental robotics

- Reduces reliance on innate knowledge
- Provides innate learning mechanisms, allowing the robot to construct its own representations of its body and its environment
- Reduces reliance on external goals and tasks
- Provides intrinsic motivation, allowing the robot to choose actions based on internally generated goals
- Increase reliance on human-robot interaction for providing the necessary scaffolding to learn

Without pre-defined tasks or goals, how can we judge success?

- This question was addressed in the April 2007 issue of the AMD Newsletter, responses included:
  - Evaluate whether the complexity of behavior has increased over time
  - Create a grand challenge, where the possible tasks are not specified in advance and encompass a wide range of simple yet varied behaviors
  - Use human psychometerics
  - Having very different morphologies from humans, robots may develop skills so different from our own that we may not be able to analyze them

Further reading

- Weng, McClelland, Pentland, Sporns, Stockman, Sur & Thelen (2001), *Autonomous Mental Development by Robots and Animals*
- Lungarella, Metta, Pfeifer & Sandini (2003), *Developmental Robotics: A Survey*
- Meeden & Blank (2006), *Introduction to Developmental Robotics*

Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop

Aaron Steinfeld, S. Rachael Bennett, Kyle Cunningham, Matt Lahut,
Pablo-Alejandro Quinones, Django Wexler, Dan Siewiorek*
Jordan Hayes†, Paul Cohen‡, Julie Fitzgerald**, Othar Hansson†, Mike Pool††,
Mark Drummond‡‡
* School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
† Bitway, Inc; ‡U. of Southern California; **JSF Consulting; ††IET; ‡‡SRI International
Correspondence: steinfeld@cmu.edu & radar@bitway.com

Abstract— Performance of a cognitive personal assistant, RADAR, consisting of multiple machine learning components, natural language processing, and optimization was examined with a test explicitly developed to measure the impact of integrated machine learning when used by a human user in a real world setting. Three conditions (conventional tools, Radar without learning, and Radar with learning) were evaluated in a large-scale, between-subjects study. The study revealed that integrated machine learning does produce a positive impact on overall performance. This paper also discusses how specific machine learning components contributed to human-system performance.

Keywords: *machine learning, intelligent systems, mixed-initiative assistants, evaluation*

I. Introduction

The RADAR (Reflective Agents with Distributed Adaptive Reasoning) project within the DARPA PAL (Personalized Assistant that Learns) program is centered on research and development towards a personal cognitive assistant. The underlying scientific advances within the project are predominantly within the realm of integrated machine learning (ML). These ML approaches are varied and the resulting technologies are diverse. As such, the integration result of this research effort, a system called Radar, is a multi-task machine learning system.

Annual evaluation on the integrated system is a major theme for the RADAR project, and the PAL program as a whole. Furthermore, there is an explicit directive to keep the test consistent throughout the program. As such, considerable effort was devoted towards designing, implementing, and executing the evaluation. This paper summarizes efforts to validate the hypothesized beneficial impact of the integrated machine learning present in Radar.

It is also important to note that the RADAR project differs from the bulk of its predecessors in that humans are in the loop for both the learning and evaluation steps. Radar was trained by junior members of the team who were largely unfamiliar with the underlying ML methods. Generic human subjects were then recruited to use Radar while handling a simulated crisis in a conference planning domain. This allowed concrete measurement of human-ML system performance. It is important to consider personal assistance systems in the context of human use due to their inherent purpose.

There have been past attempts at creating digital assistants to aid users in the performance of complex activities. Possibly the most memorable and infamous example of these is the animated paperclip accompanying Microsoft Word. Agents such as these are usually most valuable to a novice, as opposed to an experienced user.

On the opposite end of the spectrum of assistants, we can find those that are human. While human assistants are malleable, intuitive, accommodating, and are able to expand their knowledge, they lack certain characteristics present in an ideal digital assistant. Humans assistants lack perfect recall, incur latencies on time critical tasks, cannot rapidly compute optimizations and execute other taxing algorithms, are more susceptible to periodic performance losses due to turnover and constrained availability, and cannot operate continuously. Furthermore, human assistants do not scale well – providing an assistant to every human in an organization is cost prohibitive on several metrics.

Radar is an attempt to achieve the best of both worlds by focusing on a cognitive digital assistant. The presence of learning is the main distinction when using the prefix "cognitive." The knowledge it obtains can be used to automate and prep tasks, thus providing the assistance of a human without the limitations of a human and making digital assistance more adaptable and suitable for the user.

A. The Radar System

Radar is specifically designed to assist with a suite of white-collar tasks. In most cases, the specific technologies are designed to be domain agnostic (e.g., email categorizing, resource scrounging, etc). However, for the purposes of the evaluation, the base data present in Radar and used for learning is centric to the domain of conference planning. As such, certain components appear to be domain-specific but their underlying technologies are more extensible (e.g.,

Table 1. Radar components

LITW	Component	Capability
X	CMRadar-Rooms (Room Finder)	Resource scrounging by learning room reservation owner behaviors
X	Email Classifier	Task-oriented label assignment to email messages based on prior activity
X	Space-Time Planner (STP)	Elicitation of facts about the world in order to do better optimizations
X	Virtual Information Officer (VIO)	Classification and extraction to assist information updates on websites
X	Workflow by Example (WbE)	Batch website updates from training on input files
	Annotations Database (AnnoDB)	Email parsing and related natural language processing
	Scone	Knowledge representation support for the AnnoDB
X	Briefing Assistant (BA)	Summarization of activity based on prior activity *(Note: not deployed)*

conference-related email categories, room finding, etc).

While evaluation testing was performed on several Radar 1.x versions, they generally contained the same machine learning components (Table 1). The major variations were due to engineering and user interaction improvements in a number of components and the removal of the Briefing Assistant for engineering reasons. Again, the individual ML technologies will not be described in detail here – the focus here is to show that such integrated systems can provide real benefit and evaluation can be accomplished in a manner robust to unforeseen synergies and use.

An important distinction is whether a ML component "learns in the wild" or requires special interaction to gain knowledge. Learning in the wild (LITW) is a primary mission of the RADAR project and is specific to learning that occurs through the course of daily use. Brute force spoon-feeding and code-driven knowledge representation is not LITW. To count as LITW, learning must occur through regular user interaction and user interfaces present in Radar.

An example of brute force encoding would be asking someone to copy the campus building specifications into Radar all at once. However, learning is LITW if Radar decides knowing the capacity of a certain room is really important, Radar asks the user for the capacity, and the user looks it up and enters the specific value.

Table 1 details which components in Radar 1.1 were LITW and what their specific assistance entails. Note that this list is continuously growing and more components are expected in the next major release of Radar. Likewise, the next release is expected to include tighter integration between ML components. Additional detail on Radar components and capabilities is deferred to other papers.

B. Test Conditions and Hypotheses

In order to show the specific influence of learning on overall performance, there were two Radar conditions – one with learning (+L) and one without (-L). In the context of the evaluation test, learning was only LITW. Learning acquired through knowledge engineering by a programmer or through brute force encoding would be available in both the +L and -L Radar conditions.

To the user, Radar was essentially a system layered into Outlook. The components in Table 1 are either behind the scenes (e.g., Scone, AnnoDB) or visible as modified Outlook views (e.g., Email Classifier, VIO) or separate windows (e.g., STP). In many ways, the user interaction development aspect of Radar lagged behind the learning components. This was largely due to limitations in Outlook and user interaction will be improved in the next version of Radar.

A third condition where subjects utilize conventional off the shelf tools (COTS) allowed estimates to be made on the overall benefit of integration, optimization, engineered knowledge, and improvements in user interaction as compared to the current state of the art. For this application, this toolset consisted of an unaltered version of Outlook, the schedule in an Excel spreadsheet instead of the STP, a web portal to the room reservation system, and the conference website which could be manually updated.

The primary mission of the evaluation test was to examine two top-level hypotheses. These were:

1. Radar with learning (+L) will do better than Radar without learning (-L)
2. Radar will do better than conventional tools (COTS)

The comparison in Hypothesis 1 is commonly called the Learning Delta. Additional hypotheses, detail on methods, and findings can be found in [1].

C. Related Work

As previously mentioned, this was a multi-task ML system and therefore required a complex scenario for rigorous evaluation. Unfortunately, research utilizing human subjects to evaluate multi-task cognitive digital assistants with demanding tasks of this nature is limited, and so few comparison cases are available.

Furthermore, evaluations of ML systems are largely based on simulation (e.g., [2, 3]), comparison to traditional methods (e.g., [4]), subject judgments on system performance (e.g., [5]), or have sparse details on human subject evaluation (e.g., [6]). It is quite possible that this is generally the result of the kind of system that is built – something that is not meant to be an assistant but, rather, is designed to perform a task that has specific rules. An assistance system, when designed and evaluated, should be tested with humans in the loop (e.g., [7]).

As far as the rest of literature is concerned, there is relatively little literature on evaluation results of cognitive digital assistants and their focus tends to be specific to a narrow range of learning (e.g., [8, 9]). This may be because most of assistants of this nature are design exercises, lack resources for comprehensive evaluation, not evaluated with humans in the loop, and/or proprietary and unpublished.

II. Method and Materials

A key requirement for the annual evaluation test was repeatability and a consistent level of difficulty so that performance improvements can be measured across years. At a fundamental level, this is nearly impossible to achieve in a complex test of this nature. As such, the goal was to start with a test scenario that was challenging enough to accommodate synergistic learning effects, component advances, and new research directions for the out-years. A common condition, working the problem with conventional off the shelf tools (COTS) is run for each test, thus permitting benchmarking of small changes to the protocol and each test's stimulus package (e.g., specific crisis, additional tasks, etc). Furthermore, the stimulus package for the test is bound by parameters that are broad enough to prevent training to the test, but narrow enough to ensure that the stimulus package will measure the ML technologies present in the version of Radar being tested.

As mentioned, this is a system consisting of Radar and a human. At a high level this means that human subjects may need, or be required, to perform specific tasks manually. The utilization of a COTS condition where there are no Radar tools makes the ability for full manual execution a requirement. This nuance also allows for tasks and stimuli that are currently difficult for strictly software tools to complete autonomously – mixed effort towards task completion is perfectly acceptable and expected. Removal of manual control can occur if Radar technology replaces the manual inputs. For example, a user interface that allows subjects to manually scrounge for resources can be removed if a Radar component can be used to perform this task.

A. Storyline and Simulated World

The general scenario for the evaluation was that the subject was filling in for a conference planner who was indisposed, to resolve a crisis in the current conference plan. This crisis was major enough to require a major shuffling of the conference schedule and room assignments that, in turn, triggered secondary tasks. These included supporting plans (e.g., shifting catering, AV equipment delivery, adjusting room configuration, etc), reporting (e.g., make changes to the website, issue a daily briefing, etc), and customer handling (e.g., "here is the campus map"). Noise stimuli were also present in the form of unrelated email, unusable rooms, unrelated web pages, and other clutter content.

The materials included an email corpus and simulated world content. The need for repeatability over time led to the requirement for a simulated world. This consisted of facts about the world (e.g., characteristics of a particular room) and conference (e.g., characteristics of each event).

The simulated world and the initial conference were designed to provide clear boundaries on the types of tasks subjects would need to complete, yet also permit large-scale information gathering, precise measurement of learned facts, and the opportunity to induce a substantial crisis workload. The conference itself was a 4-day, multi-track technical conference complete with social events, an exhibit hall, poster sessions, tutorials, workshops, plenary talks, and a keynote address. The conference was populated with over 130 talks/posters, each with a designated speaker and title. All characters were provided with email addresses and phone numbers. Many were also given fax numbers, website addresses, and organizations.

The physical space was a modification and extension of the local university campus. In addition to modifying the student union, two academic buildings and a hotel were created and populated. These latter three buildings were instantiated to protect against campus entry knowledge in the subject pool. This information was presented to the subject in the form of revised university web pages easily accessible from the subject's home page.

Other static web content included a conference planning manual (complete with documentation of standard task constraints), a read-only file with the original schedule, and manuals for the tools used by the subjects.

Subjects were also given access to a working, realistic "university approved" vendor portal where goods and services could be ordered for the conference. These included audio-visual equipment, catering, security, floral arrangements, and general equipment rentals. Email receipts, complete with computed prices and hyperlinks to modification/cancellation pages, were delivered to the subject's mail client in real time. All vendor interactions were via web forms since automatic or Wizard of Oz handling of subject e-mails can lead to problems with stimulus consistency and realism. This had face validity since many real-life counterparts are web-based, including the subject signup website used during recruitment.

The corpus initialization for each experiment included:

- The predecessor's conference plan in the file format of the condition toolset
- Other world state information – e.g., room reservation schedule, web pages detailing room characteristics, etc. (Figure 1, top and middle)
- The vendor portal, loaded with the initial orders (Figure 1, bottom)
- Stored e-mail from the original conference planner, including noise messages and initial vendor orders
- Injected e-mail, including details of the crisis, new tasks, and noise (e.g., Table 2)

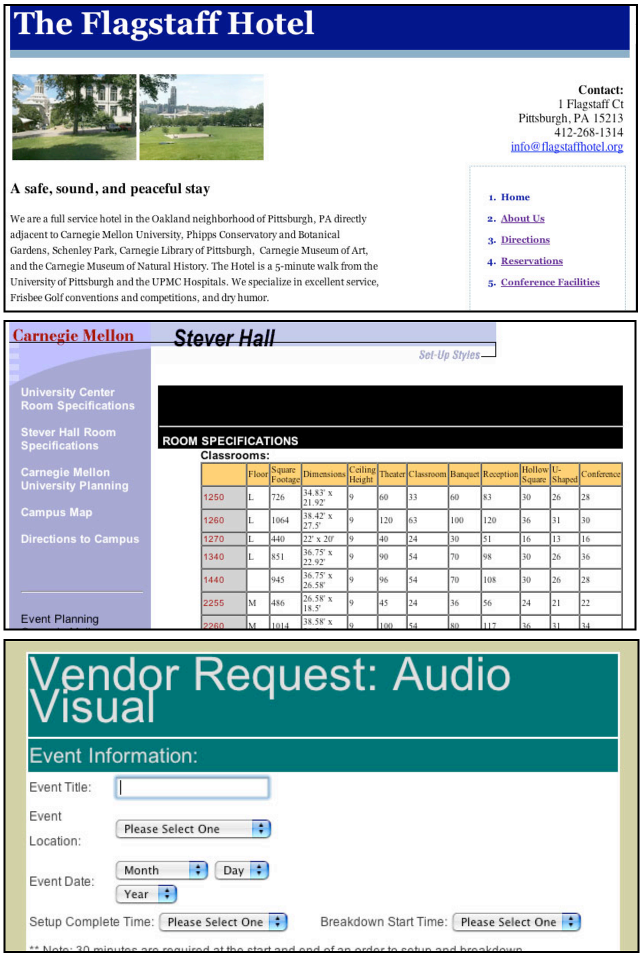
Cost is a major barrier for experimental research and a large portion is attributable to stimuli and artifact development. We have made the commitment to provide much of the stimuli and supporting content described here to external parties for re-use. This occurs through the Airspace website [10].

B. Email Corpus

The email corpus was constructed but occasionally utilized anonymized real content where appropriate (e.g., noise messages). There were initial attempts to acquire an existing email corpus centric to a conference planning activity but this posed significant challenges in the realm of Institutional Review Board (IRB) approval due to the need to anonymize all content – including subtle cues that would reveal identities. Prior attempts within the project to perform such a step produced haphazard results where entity anonymization was not sufficient.

Even a real conference planning email corpus free of IRB constraints would not be entirely adequate. A real corpus would still require considerable alignment with a simulated world (e.g., websites, rooms, etc.) and would not necessarily match the ML technologies present in the system. For example, the corpus for the real conference may completely lack website update tasks and focus heavily on what local tours to include in the registration packet.

This early investigation led to the determination that the corpus should be fabricated with an eye towards realism and the ML being tested. A team of undergraduate English majors was employed to create a detailed backstory corpus, independent messages detailing one or more tasks, and noise messages. The students were given a series of story arcs, guidelines, and a handful of characters with some specific assigned personalities (e.g., formal, annoying, etc). This effort included a directive to the email authors to let natural errors occur in their writing (e.g., signal message in Table 2). Some characters were assigned personality types that would also lead to different writing styles and email body structure (e.g., terse, bad spelling, etc). Other directives included the utilization of event, paper, and room descriptor variations (e.g., “Dowd in Stever”). Resulting content was screened for fit to the specifications, alignment with world facts, and template syntax adherence.



The Flagstaff Hotel

Contact:
1 Flagstaff Ct
Pittsburgh, PA 15213
412-268-1314
info@flagstaffhotel.org

A safe, sound, and peaceful stay

We are a full service hotel in the Oakland neighborhood of Pittsburgh, PA directly adjacent to Carnegie Mellon University, Phipps Conservatory and Botanical Gardens, Schenley Park, Carnegie Library of Pittsburgh, Carnegie Museum of Art, and the Carnegie Museum of Natural History. The Hotel is a 5-minute walk from the University of Pittsburgh and the UPMC Hospitals. We specialize in excellent service, Frisbee Golf conventions and competitions, and dry humor.

1. Home
2. About Us
3. Directions
4. Reservations
5. Conference Facilities

Carnegie Mellon **Stever Hall**
Set-Up Styles

University Center Room Specifications
Stever Hall Room Specifications
Carnegie Mellon University Planning
Campus Map
Directions to Campus
Event Planning

ROOM SPECIFICATIONS

Classrooms:

	Floor	Square Footage	Dimensions	Ceiling Height	Theater	Classroom	Banquet	Reception	Hollow Square	U-Shaped	Conference
1250	L	726	34.83' x 21.92'	9	60	33	60	83	30	26	28
1260	L	1064	38.42' x 27.5'	9	120	63	100	120	36	31	30
1270	L	440	22' x 20'	9	40	24	30	51	16	13	16
1340	L	851	36.75' x 22.92'	9	90	54	70	98	30	26	36
1440		945	36.75' x 26.58'	9	96	54	70	108	30	26	28
2255	M	486	26.58' x 18.5'	9	45	24	36	56	24	21	22
2260	M	1014	38.58' x	9	100	54	80	117	36	31	34

Vendor Request: Audio Visual

Event Information:

Event Title:
Event Location: Please Select One
Event Date: Month Day Year
Setup Complete Time: Please Select One Breakdown Start Time: Please Select One

** Note: 30 minutes are required at the start and end of an order to setup and breakdown

Figure 1. Static web and vendor portal examples

All email corpus content was in a structure which supported date shifting and variable substitution (e.g., Table 2, sender of the noise message). Date anchors and variables were stored in a separate file. These allowed for easy modification of key

Table 2. Sample messages

Signal Message	Noise Message
From: jpsontag@ardra.org To: bor@cs.cmu.edu Subject: Lucia di Lamermoor I hate to be a pest, but I finally got tickets to the opera, Lucia di Lamermoor for my wife on our aniversary. It is wednesday night. I want the whole day to ourselves, so I can avoid crashing out plans, that would be great! Let me know. The other days are fine. Thank! J.P.	From: var="kimMail" To: bor@cs.cmu.edu Subject: Hey Uncle Blake! I have a favor to ask you--Mom and Dad's anniversary is coming up, and I wanted to do something special for them, especially since they've been so supportive of the whole wedding concept. I was thinking about getting them tickets to go see "The Phantom of the Opera" when the Broadway Series came to Pittsburgh. I know that sometimes you can get cheaper tickets through work, so I was wondering if that was possible for this show. Please let me know asap so that I can make arrangements! Thanks, you're the best! Kim

values by the external program evaluators and time shifting of the corpus for experiment execution.

C. Objective Performance Measurement

As experiment-friendly conference planning performance measures are not readily available, a new method was utilized. It was extremely important that this measurement be tied to objective conference planning performance rather than a technology-specific algorithm (e.g., F1 for classification). This technology agnostic approach also permits accurate measurement of component synergies and human use strategies.

Creation of this measurement was largely achieved through an evaluation score designed and developed by the external program evaluators (authors JF, MP, and PC). This complex score function summarized overall performance into a single objective score ("Final_Score" range from 0.000 to 1.000). Performance was in terms of points collected by satisfying certain conditions coupled with penalties for specific costs. These included quality of conference schedule (e.g., constraints met, special requests handled, etc), adequate briefing to conference chair, accurate adjustment of the website (e.g., contact information changes, updating the schedule on the website, etc), and costs incurred while developing schedule. Such costs included both the budget and how often subjects asked fictional characters to give up their room reservations. Additional detail on scoring is deferred to other documents. At the top level, the score coefficients were 2/3rd for the schedule (including penalties for costs incurred), 1/6th for website updating, and 1/6th for briefing quality.

In addition to this measure, subjects also completed a post-test survey designed to measure perception of system benefit, assistance, and other related metrics. Details on the survey design and results are reported elsewhere [11].

D. Procedure

Each subject was run through approximately 3 hours of testing (1 for subject training and 2 for time on task). Each cohort of subjects for a particular session was run on a single condition (COTS, Radar -L, or Radar +L). When possible, cohorts were balanced over the week and time of day to prevent session start time bias. Follow-up analyses on this issue revealed no apparent bias. The nominal cohort size was 15 but was often lower due to dropouts, no-shows, and other subject losses (e.g., catastrophic software crash). Cohorts were run as needed to achieve approximately 30 subjects per condition.

Motivation was handled through supplemental payments for milestone completion (e.g., the conference plan at the end of the session satisfies the constraints provided). Subjects were given general milestone descriptions but not explicit targets. These milestones roughly corresponded to the top-level coefficients in the score function.

III. Results

A. Data Source for this Example

There were several test windows during the run-up to the data shown here. This corresponds to COTS and Radar 1.1 tested with a stimulus package of 107 messages, 42 of which were noise.

The crisis for this package was a loss of the bulk of the conference rooms for 1.5 days (out of 4 total). A variety of other small perturbations rounded out the task set. These

Table 3. RADAR 1.1 means and t-test comparisons

Condition	Mean	Comparison	p-value
COTS	0.452	Overall Delta (With Learning > COTS)	<0.0001
No Learning (-L)	0.492	Learning Delta (With Learning > No Learning)	<0.0001
With Learning (+L)	0.605	Nonlearning Delta (No Learning > COTS)	<0.041

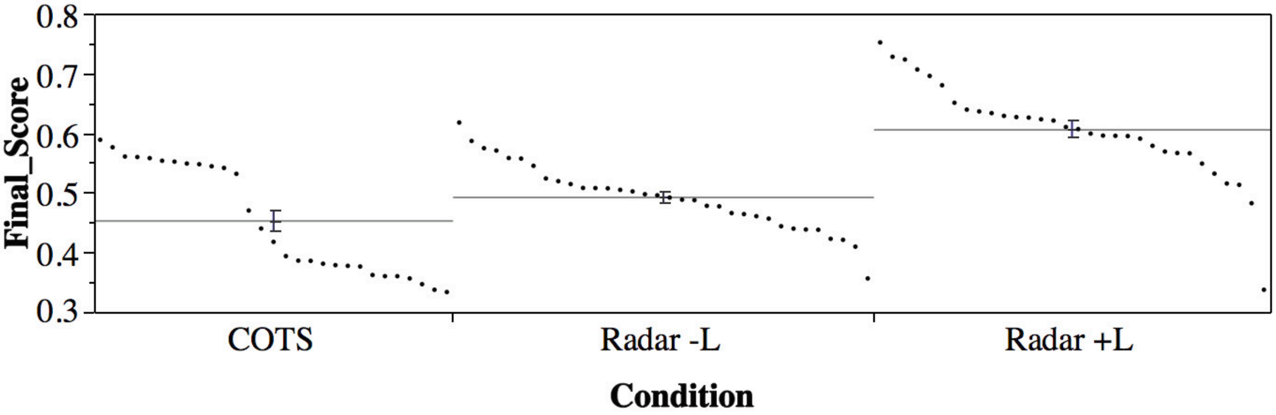


Figure 2. Radar 1.1 results on Crisis 1 (Score 2.0)

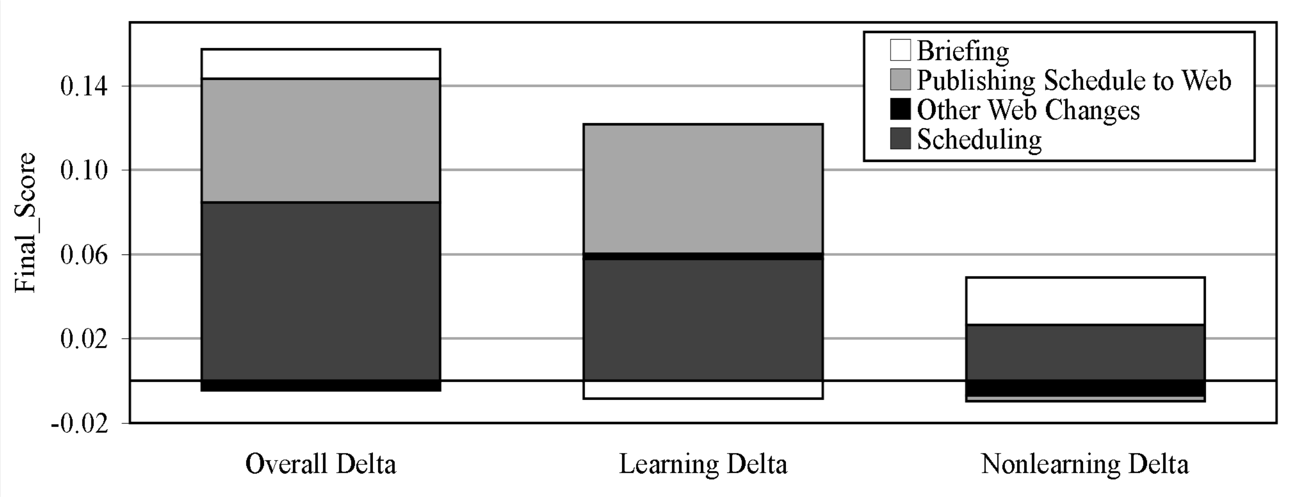


Figure 3 Score component impacts on the overall score (Score 2.0)

included changes to room details, speaker availability, session preferences, and website details. This stimulus package (aka Crisis 1) was designed by the external evaluators. As of this paper, the external evaluators have designed three different crisis packages.

The subject pool used for analysis, after exclusions and dropouts, was 29, 34, and 32 (COTS, Radar -L, and Radar +L). As such, this test accumulated 64 cumulative hours worth of time on task by subjects with a multi-ML system.

Scheduling and scoring for the conditions shown here was not in parallel. COTS data was collected in the fall of 2005 and the Radar data was collected in the spring of 2006. The data described here were scored with version 2.0 of the external evaluator's scoring algorithms (aka Score 2.0).

B. Final_Score Results

Figure 2 shows between subject performance across the three conditions. The Learning Delta (the difference due to the inclusion of machine learning) is 0.113, which is approximately 74% of the Overall Delta (improvement over COTS). This suggests that machine learning was the prime contributor to the performance gains. In this graph, all condition differences are significant and in the expected direction for the initial hypotheses (Table 3).

The need for an integrated evaluation with humans in the loop becomes especially apparent when examining the makeup of the Deltas (Figure 3). Subjects noticeably altered their strategies and use of assistance technology based on the presence/absence of specific features. For example, COTS subjects clearly focused on updating individual website corrections (e.g., "my name is spelled wrong") over other activities – probably due to familiarity with website form manipulations. Likewise, subjects in the Radar conditions took full advantage of autonomous components to relieve time pressure (i.e., schedule optimizer in both -L and +L, batch website updating in +L, etc).

Table 4. Learning contributors to score component

Score Component	Learning Contributors
Scheduling	STP, CMRadar-Rooms, Email Classifier
Publishing Schedule to Web	WbE
Other Web Changes	VIO, WbE, Email Classifier
Briefing	Email Classifier

Gains due to publishing the schedule to the website can be tied explicitly back to WbE, but is not the only place where WbE can contribute/detract from overall performance (Table 4). Note that while the Email Classifier contributes to many factors of the score function, its role is to surface the task and not to assist with the completion of the task itself. As such, the negative Learning Delta for the briefing component (Figure 3) is not solely due to a deficiency of the Email Classifier. In fact, this difference is due to human decision making related to task allocation – almost twice as many subjects in the nonlearning condition as in the learning condition compiled a briefing (56% vs. 28%). Task identification is not the same as task prioritization, hence the importance of an overall task performance measurement.

IV. Discussion

The results clearly show that Hypothesis 1 (ML helps) holds true. Likewise, Hypothesis 2 (Radar is better than COTS) is also true. Furthermore, it is clear that component value was highly dependent on how subjects allocated effort – some technologies were underutilized based on strategic decisions.

The initial concern at the start of this endeavor was that the methods and materials would not be adequately sensitive to measure mixtures of ML technologies that were still being formulated. This concern is still valid in that there are new ML components being developed for the next version of

Radar. The decision to measure at the top human-Radar system level was an attempt to be robust to unknown ML technologies. While this limits the ability to directly account for specific component benefit, this approach clearly captures high-level benefits and use patterns for human in the loop multi-task ML.

While not shown here, there have been other human subjects tests with other versions of the system and the protocol. These have shown changes in performance due to variations in ML, HCI, engineering, crisis difficulty, and human training. As such, the test method and materials have also been shown to be suitable for measuring shifts in performance due to a variety of system and scenario effects.

ACKNOWLEDGMENTS

Othar Hansson and Mike Pool joined Google and Convera (respectively) after contributing to this work.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

REFERENCES

[1] Steinfeld, A., Bennett, R., Cunningham, K., Lahut, M., Quinones, P.-A., Wexler, D., Siewiorek, D., Cohen, P., Fitzgerald, J., Hansson, O., Hayes, J., Pool, M., and Drummond, M., The RADAR Test Methodology: Evaluating a Multi-Task Machine Learning System with Humans in the Loop. 2006, Carnegie Mellon University, School of Computer Science: Pittsburgh, PA. http://reports-archive.adm.cs.cmu.edu/anon/2006/abstracts/06-125.html

[2] Clymer, J. R. Simulation of a vehicle traffic control network using a fuzzy classifier system. In Proc. of the IEEE Simulation Symposium. 2002.

[3] Clymer, J. R. and Harrsion, V. Simulation of air traffic control at a VFR airport using OpEMCSS. In Proc. IEEE Digital Avionics Systems Conference. 2002.

[4] Zhang, L., Samaras, D., Tomasi, D., Volkow, N., and Goldstein, R. Machine learning for clinical diagnosis from functional magnetic resonance imaging. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2005.

[5] Hu, Y., Li, H., Cao, Y., Meyerzon, D., and Zheng, Q. Automatic extraction of titles from general documents using machine learning. In Proc. of ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL). 2005.

[6] Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W. PLOW: A Collaborative Task Learning Agent. In Proc. Conference on Artificial Intelligence (AAAI). 2007. Vancouver, Canada.

[7] Schrag, R., Pool, M., Chaudhri, V., Kahlert, R., Powers, J., Cohen, P., Fitzgerald, J., and Mishra, S. Experimental evaluation of subject matter expert-oriented knowledge base authoring tools. In Proc. NIST Performance Metrics for Intelligent Systems Workshop. 2002. http://www.iet.com/Projects/RKF/PerMIS02.doc

[8] Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In Proc. International Conference on Intelligent User Interfaces (IUI). 2006.

[9] Yoo, J., Gervasio, M., and Langley, P. An adaptive stock tracker for personalized trading advice. In Proc. International Conference on Intelligent User Interfaces (IUI). 2003.

[10] Airspace: Tools for evaluating complex systems, machine language, and complex tasks. http://www.cs.cmu.edu/~airspace

[11] Steinfeld, A., Quinones, P.-A., Zimmerman, J., Bennett, S. R., and Siewiorek, D. Survey measures for evaluation of cognitive assistants. In Proc. NIST Performance Metrics for Intelligent Systems Workshop (PerMIS). 2007.

Survey Measures for Evaluation of Cognitive Assistants

Aaron Steinfeld, Pablo-Alejandro Quinones, John Zimmerman,
S. Rachael Bennett, Dan Siewiorek
School of Computer Science, Carnegie Mellon University
Pittsburgh, PA, USA
{steinfeld@, paq@andrew, johnz@cs, srbennet@andrew, dps@cs}.cmu.edu

Abstract— A survey designed to measure subject perception of benefit, ease of use, usefulness, collaboration, disorientation, flow, and assistance was used to evaluate two releases of an integrated machine learning cognitive assistance system. The design and validity of this evaluation survey is discussed in the context of an information overload experiment.

Keywords: *subjective performance, intelligent systems, evaluation*

I. Introduction

As part of the RADAR project, a cognitive assistant equipped with integrated machine learning capability is regularly evaluated in human subject experiments. This effort is driven by the belief that machine learning, especially when implemented in complex integrated systems, needs to be evaluated on realistic tasks with a human in the loop. Furthermore, the evaluation is designed to examine the impact of machine learning under information overload conditions.

Unfortunately, research utilizing human subjects to evaluate machine learning centric digital assistants with demanding tasks of this nature is limited. As such, few comparison cases are available. Worse, survey tools to measure user perception of such systems are even harder to find in the literature. Validated surveys are especially valuable in that cross-domain and cross-application comparisons are often more appropriate than purely objective metrics.

Evaluations of many machine learning systems are largely based on simulation (e.g., [1, 2]), comparison to traditional methods (e.g., [3]), and subject judgments on system performance (e.g., [4]). It is quite possible that this is generally the result of the kind of system that is built – something that is not meant to be an assistant but, rather, is designed to perform a task that has specific rules. An assistance system, when designed and evaluated, should be tested with humans in the loop (e.g., [5]).

There is relatively little literature on evaluation results of cognitive digital assistants and their focus tends to be specific to a narrow range of machine learning (e.g., [6, 7]). This may be because most of assistants of this nature are design exercises, lack resources for comprehensive evaluation, not evaluated with humans in the loop, and/or proprietary and unpublished.

Likewise, explorations of suitable exit surveys (e.g., [8-11]) provided promising survey questions but uncovered few measures validated for cognitive personal assistants. NASA-TLX was considered but deemed too narrow for examination of certain system assistance nuances.

This paper addresses the subsequent efforts by the RADAR testing team to develop and validate a survey for evaluating complex technologies under information overload.

A. System and Conditions

Radar, the project's implemented system, is specifically designed to assist with a suite of office tasks. In most cases, the specific technologies are designed to be domain agnostic (e.g., email categorizing, resource scrounging, etc). However, for the purposes of the evaluation, the base data present in Radar and used for learning is centric to the domain of conference planning. As such, certain components appear to be domain-specific but their underlying technologies are more extensible (e.g., conference-related email categories, room finding, etc).

In order to show the specific influence of learning on overall performance, there were two Radar conditions – one with learning (+L) and one without (-L). In the context of the evaluation test, learning was only "learning in the wild" (LITW). Such machine learning is specific to learning that occurs through the course of daily use. Brute force spoon-feeding and code-driven knowledge representation is not LITW. To count as LITW, learning must occur through regular user interaction and user interfaces present in Radar.

The other experimental condition described here is which version of Radar (1.0 or 1.1) was tested. There were significant improvements in both usability and engineering from Radar 1.0 to 1.1.

II. Method

A. Materials and Storyline

Extensive detail on the protocol, materials, and findings on other metrics, especially those specific to overall task performance, can be found in [12, 13]. As mentioned, this paper is focused on the survey design and results.

The general scenario for the evaluation was that the subject was filling in for a conference planner, who was indisposed, to resolve a crisis in the current conference plan. This crisis was major enough to require a major shuffling of the conference schedule and room assignments that, in turn, triggered secondary tasks. These included supporting plans (e.g., shifting catering, AV equipment delivery, adjusting room configuration, etc), reporting (e.g., make changes to the website, issue a daily briefing, etc), and customer handling (e.g., "here is the campus map"). Noise stimuli were also present in the form of unrelated email, unusable rooms, unrelated web pages, and other clutter content.

The materials included an email corpus and simulated world content. The need for repeatability over time led to the requirement for a simulated world. This consisted of facts about the world (e.g., characteristics of a particular room) and conference (e.g., characteristics of each event).

The simulated world and the initial conference were designed to provide clear boundaries on the types of tasks subjects would need to complete, yet also permit large-scale information gathering, high resolution on learned fact variation, and the opportunity to induce a substantial crisis workload.

The conference itself was a 4-day, multi-track technical conference complete with social events, an exhibit hall, poster sessions, tutorials, workshops, plenary talks, and a keynote address. The conference was populated with over 130 talks/posters, each with a designated speaker and title. All characters were provided with email addresses and phone numbers. Many were also given fax numbers, website addresses, and organizations.

The physical space was a modification and extension of the local university campus. In addition to modifying the student union, two academic buildings and a hotel were created and populated. These latter three buildings were instantiated to protect against campus entry knowledge in the subject pool. This information was presented to the subject in the form of revised university web pages easily accessible from the subject's home page.

Other static web content included a conference planning manual (complete with documentation of standard task constraints), a PDF of the original schedule, and manuals for the tools used by the subjects.

Subjects were also given access to a working, realistic "university approved" vendor portal where goods and services could be ordered for the conference. These included audio-visual equipment, catering, security, floral arrangements, and general equipment rentals. Email receipts, complete with hyperlinks to modification/cancellation pages and computed prices, were delivered to the subject's mail client in real time. All vendor interactions were via web forms since automatic or Wizard of Oz handling of subject e-mails can lead to problems with stimulus consistency and realism. This had face validity since many real-life counterparts are web-based, including the subject signup website used during recruitment.

The corpus initialization for each experiment included:

- The predecessor's conference plan in the file format of the condition toolset,
- Other world state information – e.g., room reservation schedule, web pages detailing room characteristics, etc.,
- Stored e-mail from the original conference planner, including noise messages and initial vendor orders,
- The vendor portal, loaded with the initial orders, and
- Injected e-mail, including details of the crisis, new tasks, and noise.

B. Survey Metrics

The survey questions, and their respective categories, are shown in Table 1. All ratings were a 7-point scale with anchors at 1, 4, and 7 (Strongly agree, Neutral, Strongly disagree). Categories – e.g., metrics – were not revealed to the subjects.

Questions in the Ease of Use, Usefulness, Disorientation, and Flow categories were drawn from surveys validated in other fields [10, 11]. Questions 10, 11, and 13 in the Collaboration section were adapted from surveys validated in computer supported cooperative work research [8, 9]. Given the dramatic differences from the fields in which these survey questions were validated, there was some concern that adaptation for complex intelligent systems would not result in valid measures.

For the purposes of analysis, responses to each question within each category were flipped to have the same positive/negative direction and averaged as a group. This category level rating is referred to as an index (e.g., Ease of Use index). The exception is the General category – these are not designed to measure a common metric, so they are left independent.

Questions 16 and 17 were specifically designed to examine how the specific mixture of user interaction, machine learning, and automation affected perceived relationships within collaboration. Ideally, a good mixture will lead to a low score for Question 16 and a higher score for Question 17. This would mean the system was perceived as behaving as an assistant, rather than a taskmaster. The fear with machine learning, and in fact all assistance software, is that the needs of the software (e.g., confirmation, corrections, reminders, etc) will lead to user perception that the locus of control is with the software, rather than the user. It is possible to envision cases where a system has good usability and excellent machine learning, but the nature of the interaction leads the user to feel that they are serving the software.

D. Procedure

Each subject was run through approximately 3 hours of testing (1 for subject training and 2 for time on task). The survey was given at the end of the session. Each cohort of subjects for a particular session was run on a single condition (COTS[1], Radar -L, or Radar +L). When possible, cohorts

[1] Conventional Off The Shelf, see [13] for more details.

were balanced over the week and time of day to prevent session start time bias. Follow-up analyses on this issue revealed no apparent bias. The nominal cohort size was 15 but was often lower due to dropouts, no-shows, and other subject losses (e.g., catastrophic software crash). Cohorts were run as needed to achieve approximately 30 subjects per condition.

Motivation was handled through supplemental payments for milestone completion (e.g., the conference plan at the end of the session satisfies the constraints provided). Subjects were given general milestone descriptions but not explicit targets.

All subjects were recruited from local universities and the general public using a local human subject recruitment website. Subjects were required to meet the following criteria:

- Between the ages of 18 and 65,
- Do not require computer modifications,
- Fluent in English, and
- Not affiliated with or working on the RADAR project.

III. Results

There were several test windows during the period reported here. The survey results data in this document correspond to Radar 1.0 and 1.1 tested on the stimulus package referred to as Crisis 1. The survey reliability data is for the Radar 1.1 test only. Details on Radar 1.1 and Crisis 1 can be found elsewhere [12, 13].

The Radar 1.0 subject pool used for results analysis, after exclusions and dropouts, was 31 and 47 (-L, and +L). Radar 1.1 pool size was 34 and 32. As such, these two tests accumulated 158 cumulative hours worth of time on task by subjects with a multi-task machine learning system.

A two-way ANOVA model on Version (1.0, 1.1) and Learning (-L, +L) was run. Differences between the latter on the survey measures were largely not significant. The exception to this was Usefulness which was viewed as better for Radar +L (F-Ratio, 5.05; p-value 0.026). However, almost every survey measure reported that Radar 1.1 was an improvement over Radar 1.0 (Table 2). Only Question 1 (Confident did task well) was marginally significant.

Figure 2 shows the corresponding means for Version and

Table 2. Improvement for new system version

General Survey Questions	**F-Ratio**	**p-value**
1. Confident did task well	3.89	0.051
2. Task difficult to complete	5.31	0.023
3. As good without software	17.3	<0.0001
Survey index	**F-Ratio**	**p-value**
Ease of Use	10.9	0.0012
Usefulness	4.88	0.029
Collaboration	6.03	0.015
Disorientation	4.13	0.044
Flow	4.31	0.040
Relationship Metric	**F-Ratio**	**p-value**
Assistant vs. Taskmaster (Q17 – Q16, higher is better)	10.2	0.0018

Table 1. Survey Questions

General

1. I am confident I completed the task well. *(r)*
2. The task was difficult to complete. *(r)*
3. I could have done as good of a job without the software tools. *(r)*

Ease of Use *Cronbach's alpha:* 0.87

4. Learning to use the software was easy. *(r)*
5. Becoming skillful at using the software was easy. *(r)*
6. The software was easy to navigate. *(r)*

Usefulness 0.94

7. Using similar software would improve my performance in my work. *(r)*
8. Using similar software in my work would increase my productivity. *(r)*
9. I would find similar software useful in my work. *(r)*

Collaboration 0.69

10. I disagreed with the way tasks were divided between me and the computer.
11. Tasks were clearly assigned. I knew what I was supposed to do. *(r)*
12. The software did exactly what I wanted it to do. *(r)*
13. I found myself duplicating work done by the software.
14. I could trust the software. *(r)*
15. The software kept track of details for me. *(r)*
16. The software was assisting me. *(r)*
17. I was assisting the software.

Disorientation 0.81

18. I felt like I was going around in circles.
19. It was difficult to find material that I had previously viewed.
20. Navigating between items was a problem.
21. I felt disoriented.
22. After working for a while I had no idea where to go next.

Flow 0.57

23. I thought about other things.
24. I was aware of other problems.
25. Time seemed to pass more quickly. *(r)*
26. I knew the right things to do. *(r)*
27. I felt like I received a lot of direct feedback. *(r)*
28. I felt in control of myself. *(r)*

All responses on 7-point scales:

1	2	3	4	5	6	7
Strongly agree			Neutral			Strongly disagree

(r) = scale reversed for index averages and analysis

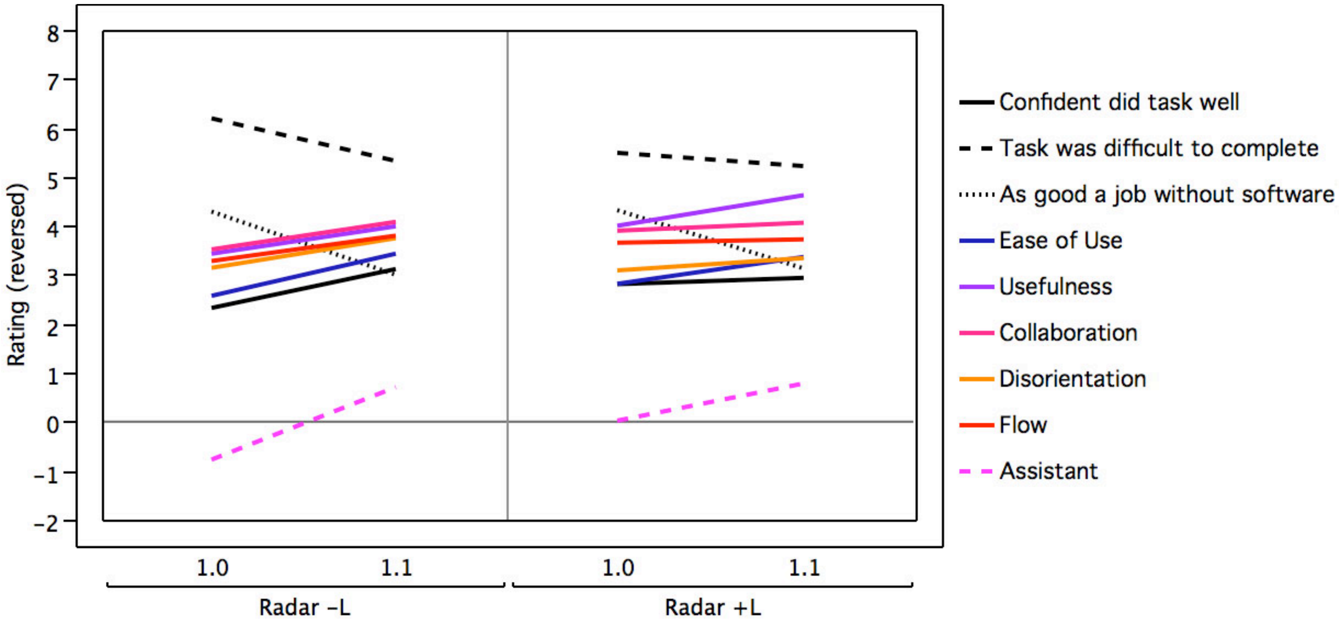


Figure 1. Mean survey responses (Assistant & indices, higher is better; statements, 1=strongly disagree, 7 strongly agree)

Learning. While the interaction comparisons were not significant, it is worth noting is that there is an apparent overall pattern where improvements across versions are less pronounced when machine learning is present in Radar. This matches ground truth in that the majority of the user detectable improvements between versions were in the usability area.

Also, Radar 1.0 -L has a negative Assistant value; subjects felt this instance of Radar was more of a taskmaster than an assistant. The latter finding is not surprising in that the Radar 1.0 user interaction was extremely onerous and only marginal assistance was provided by the software due to the lack of machine learning. This suggests that the machine learning in Radar 1.0 was enough to offset these known deficiencies.

In general, the index collections performed reasonably well when tested for measurement reliability using the Radar 1.1 data (Table 1). Only the Flow index was markedly below the 0.7 reliability acceptance threshold used in the literature. Collaboration was right on the edge.

An initial estimate of the validity of the Assistant vs. Taskmaster relationship metric is to examine how well it correlates to Question 3 (As good without software). Theoretically, ratings on this metric should decrease as Question 3 increases – i.e., software that is considered a taskmaster will not be regarded as valuable by the end user. This was indeed the result for this data set and these measures were correlated (-0.42; p-value <0.0001; Figure 2). As such, early indications are good with respect to metric validity. However, additional research is needed with more precise measures of assistant/taskmaster ground truth.

IV. Discussion

At the time of the Radar 1.1 test there were still unaddressed issues in usability and engineering. The limited perceived differences in the Learning effect beyond Usefulness, contrary to findings from performance metrics [12, 13], may be due to these remaining issues. Possible explanations include: (a) the poor user experience depressed positive machine learning influences and (b) the improvements in machine learning were not perceptible in a between subjects study design.

At the time of this writing, the next round of annual Radar experiments is underway and additional data on issues like the impact of machine learning and index reliability will become available. Early indications are especially promising on the ability of these metrics to capture the precieved value of machine learning. A larger Learning effect is expected since both the user experience and machine learning aspects of Radar have improved substantially. Unfortunately, final data and analyses are not available yet.

There was a clear feeling within the team that the user interfaces for Radar 1.0 and 1.1 were masking the value provided by the machine learning. To some degree, the results presented here confirm this suspicion and reinforce the importance of good user interaction design.

Having said this, the improvement in survey scores from Radar 1.0 to 1.1 mirrors the ground truth improvements made to the system itself. This, combined with the good reliability results, suggests that these survey measures have merit for other experiments on human use of intelligent assistance systems.

Acknowledgments

At Carnegie Mellon, Robert Kraut provided valuable help with previously validated surveys. Kyle Cunningham, Django Wexler, and Matt Lahut assisted with the development of the study materials.

Jordan Hayes (Bitway, Inc) and Othar Hansson integrated Radar and assisted with experiment execution. Julie Fitzgerald

(JSF Consulting), Mike Pool, and Paul Cohen (University of Southern California) served as external evaluators and generated the crisis stimulus. They, with Mark Drummond (SRI International), provided significant input on the protocol.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

REFERENCES

[1] Clymer, J. R. Simulation of a vehicle traffic control network using a fuzzy classifier system. In Proc. of the IEEE Simulation Symposium. 2002.

[2] Clymer, J. R. and Harrsion, V. Simulation of air traffic control at a VFR airport using OpEMCSS. In Proc. IEEE Digital Avionics Systems Conference. 2002.

[3] Zhang, L., Samaras, D., Tomasi, D., Volkow, N., and Goldstein, R. Machine learning for clinical diagnosis from functional magnetic resonance imaging. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2005.

[4] Hu, Y., Li, H., Cao, Y., Meyerzon, D., and Zheng, Q. Automatic extraction of titles from general documents using machine learning. In Proc. of ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL). 2005.

[5] Schrag, R., Pool, M., Chaudhri, V., Kahlert, R., Powers, J., Cohen, P., Fitzgerald, J., and Mishra, S. Experimental evaluation of subject matter expert-oriented knowledge base authoring tools. In Proc. NIST Performance Metrics for Intelligent Systems Workshop. 2002. http://www.iet.com/Projects/RKF/PerMIS02.doc

[6] Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In Proc. International Conference on Intelligent User Interfaces (IUI). 2006.

[7] Yoo, J., Gervasio, M., and Langley, P. An adaptive stock tracker for personalized trading advice. In Proc. International Conference on Intelligent User Interfaces (IUI). 2003.

[8] Fussell, S. R., Kraut, R. E., Lerch, F. J., Sherlis, W. L., McNally, M., and Cadiz, J. J. Coordination, overload and team performance: effects of team communication strategies. In Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW). 1998.

[9] Kraut, R. E., Fussell, S. R., Lerch, F. J., and Espinosa, A., Coordination in teams: Evidence from a simulated management game. Journal of Applied Psychology, under review.

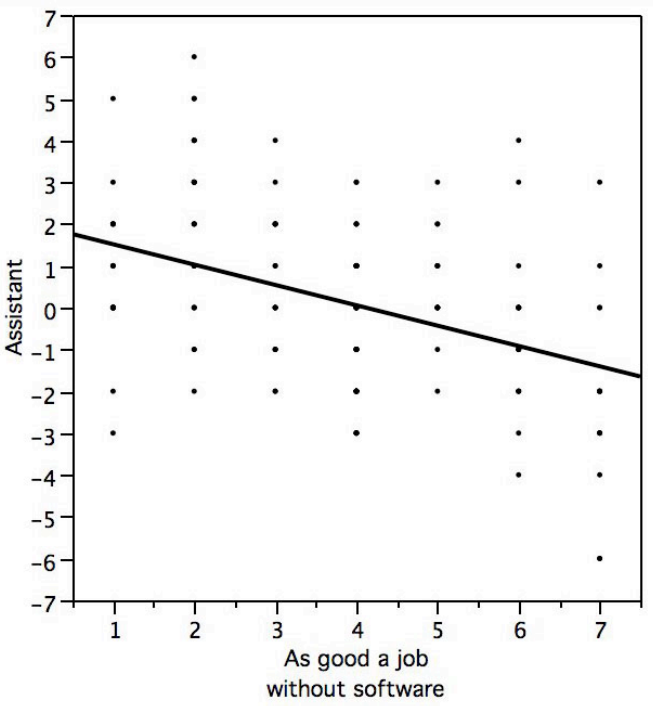


Figure 2. Assistant vs. Taskmaster metric as compared to "I could have done as good of a job without the software tools" (negative slope is better)

[10] van Schaik, P. and Ling, J., Using on-line surveys to measure three key constructs of the quality of human–computer interaction in web sites: psychometric properties and implications. Int. Journal of Human-Computer Studies, 2003. 59: p. 545-567.

[11] van Schaik, P. and Ling, J., Five psychometric scales for online measurement of the quality of human-computer interaction in web sites. Int. Journal of Human–Computer Interaction, 2005. 18(3): p. 309-322.

[12] Steinfeld, A., Bennett, S. R., Cunningham, K., Lahut, M., Quinones, P.-A., Wexler, D., Siewiorek, D., Hayes, J., Cohen, P., Fitzgerald, J., Hansson, O., Pool, M., and Drummond, M. Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop. In Proc. NIST Performance Metrics for Intelligent Systems Workshop (PerMIS). 2007.

[13] Steinfeld, A., Bennett, R., Cunningham, K., Lahut, M., Quinones, P.-A., Wexler, D., Siewiorek, D., Cohen, P., Fitzgerald, J., Hansson, O., Hayes, J., Pool, M., and Drummond, M., The RADAR Test Methodology: Evaluating a Multi-Task Machine Learning System with Humans in the Loop. 2006, Carnegie Mellon University, School of Computer Science: Pittsburgh, PA. http://reports-archive.adm.cs.cmu.edu/anon/2006/abstracts/06-125.html

Development of Tools for Measuring the Performance of Computer Assisted Orthopaedic Hip Surgery Systems

Nicholas G. Dagalakis, Yongsik Kim, Daniel Sawyer, Craig Shakarji
Manufacturing Engineering Laboratory
National Institute of Standards and Technology
100 Bureau Dr. Stop 8230
Gaithersburg, MD 20899-8230

Abstract— In the late seventies a sensor was invented, which could track the movement of athlete body parts. In the early eighties an improved version of this sensor was introduced, by a group of NIST researchers, for the calibration and the performance testing of industrial robots. In the late eighties people experimented with the use of these sensors for human brain operations and in the early nineties these sensors were introduced to orthopaedic operations and the field of Computer Assisted Orthopaedic Surgery (CAOS) was born. Although significant progress has been made in the design and use of these sensors for medical applications, there are still sources of accuracy errors that must be addressed. This paper describes our work on the development of tools for the calibration and performance testing of CAOS systems, which can be used inside operating rooms.

Keywords: *computer assisted surgery, computer assisted orthopaedic surgery, hip arthroplasty, phantom, artifact*

I. INTRODUCTION

In the early eighties a group of National Institute of Standards and Technology (NIST) researchers, working for the NIST/Center for Manufacturing Engineering, the predecessor of the NIST/Manufacturing Engineering Laboratory, modified an athlete body tracking sensor [1[1]], so that it can be used for robot calibration and performance measurements [2]. An extensive study of the sources of measurement errors of this sensor and its controller was performed. Soon this sensor became a commercial product and it has been used by manufacturers and users of industrial robots, for their robot calibration and performance measurements, for the last 20 years. In the early nineties Nolte L.P. [3] used this type of tracking sensor for precision enhancement in spine surgery. Spine surgery tools were equipped with probes holding three or more target Light Emitting Diodes (LEDs), that were tracked to determine the surgical tool position and orientation. A Dynamic Reference Base (DRB) coordinate frame, equipped with three or more target LEDs, was attached to the vertebra undergoing surgery. Appropriate mathematical transformations converted the surgical tool's position and orientation to DRB frame coordinates, thus facilitating the insertion of screws at the right position and orientation in the overwhelming majority of cases. At about the same time Lavallee S. [4] performed spine surgeries using a similar tracking sensor system. Lavallee experimented with surface registration for the identification of characteristic bone landmarks, instead of simple point registration. He also experimented with a robot carrying a laser beam for surgical drill tool alignment. Soon these techniques were extended to total hip and knee arthroplasties and the field of Computer Assisted Orthopaedic Surgery (CAOS) was born.

The market for the use of CAOS systems inside an operating room in order to guide orthopaedic operations has evolved significantly from the original Selspot athlete body tracking sensor system. The Selspot system used two lateral effect photodiode camera tracking sensors, while most of the modern CAOS systems use two or three Charge Coupled Device (CCD) cameras with active LED targets or passive sphere targets illuminated by infrared light. People have also experimented with electromagnetic tracking sensors, with electrical coil targets and other technologies. Although these types of targets do not require line of sight with the sensor and thus can operate inside the human body, they are susceptible to interference from electromagnetic waves reflected by metal surfaces inside the operating room. Computer Assisted Orthopaedic Surgery systems consist of tracking camera sensors, tracking markers (targets), a computer and other relevant electronics [5]. During an operation the markers are attached to bones, surgical tools and implants. The three dimensional space position of the markers is determined with

[1] Certain commercial products and processes are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products and processes identified are necessarily the best available for the purpose.

respect to a reference frame and based on that information the position and orientation of tools, bones and implants is calculated and used to generate useful surgery information. Comparison of conventional versus CAOS assisted arthroplasty operations have demonstrated that CAOS systems show significant improvement in the desired surgical result. In particular CAOS systems help reduce the variability of the positioning of prosthetic components from the desired optimum position and orientation, thus permitting a more consistent placement of the prosthetic components [6, 7].

It did not take very long though for the users of CAOS systems to recognize that the tracking sensors have accuracy problems, which may jeopardize the outcome of the surgical operation. The original NIST study identified several sources of errors. Some of them could still be relevant and can introduce positioning errors for the modern CAOS systems. Here is a list of these possible sources of errors:

1. Camera optics.
2. Detector irregularities.
3. Target operating conditions, like temperature, non uniform radiation field, distance from the camera sensors, etc.
4. Camera position and orientation determination with respect to the tracking sensor system reference coordinate frame.
5. Sampling rate frequency of multiple targets.

The image generated by each target on the camera tracking sensor is usually an irregular blob with non-uniform intensity distribution. It is up to the controller of each tracking system to decide how to assign XY coordinates to this type of image. A simple rotation of the target, with no position change, could alter the value of the measured XY coordinates. In the case of slow sampling rate tracking systems the target might move while its position is still being sampled. The general conclusion of the NIST study was that these tracking systems have a sweet region of low error for target positions located within the 80 % of the camera detector field of view. This error increases as the target moves away from this central region.

The focus of the work reported in this paper is to address the accuracy problems associated with the use of Computer Assisted Orthopaedic Surgery (CAOS) systems, by implementing well calibrated artifacts, called phantoms by most medical professionals.

II. BRIEF REVIEW OF TOTAL HIP ARTHROPLASTY OPERATION

Various human diseases and activities can damage the hip joint and lead to severe pain and loss of mobility. Surgery to replace the damaged joint with an artificial one, prosthesis, is usually the last resort in order to alleviate pain and restore mobility [8]. This operation was invented by Dr. Charnley, a British surgeon, in the sixties who was honored with knighthood for his contribution. During the operation the head of the femur (thigh bone) is removed with a saw and the pelvis socket is reshaped in to a hemisphere with a scraping tool called a reamer. There are two major categories of joint prostheses, the cemented and the uncemented ones. The cemented are attached to the bone with an epoxy cement, while the uncemented have a porous external surface where bone can grow in order to attach the prosthesis to the skeletal bone. The hip prosthesis consists of two major parts; the femoral component and the acetabular component (see images in Figure 1 and 2). The femoral component is made of a metal stem and a metal or ceramic ball head and is intended to replace the upper part of the femur bone. The acetabular component is usually made of a concave metal shell cup, and a plastic inner liner. During the operation the pelvis socket, is reshaped before the acetabular prosthesis head can be inserted. The initial step before the operation is to determine the coordinates of the center of rotation of the hip and ankle joints in order to calculate the length of the leg. This test must be repeated before the conclusion of the operation and adjustments must be made in order for the patient to exit the operating room with the proper length leg, since a portion of his femur bone and pelvis have been removed. Another critical step of this operation is the attachment of the acetabular component of the prosthesis. It has been found that the metal shell cup must be placed with precise angular orientation otherwise the prosthesis could fail due to dislocation, impingement and premature wear. The angles that define the correct angular orientation are defined with respect to the patient pelvis frontal (coronal) and transverse coordinate planes, which are difficult to locate while the patient is lying on the operating table.

After the acetabular component has been inserted the femur bone cavity is reshaped in order to accept the stem of the femoral prosthesis. The size and shape of the stem can vary from one patient to another. The femur bone cavity is usually shaped with manual tools although orthopaedic surgeons are also experimenting with robotic milling tools [9]. The robotic tool creates a smooth surface cavity, which should be less prone to stress concentrations that can lead to bone fractures.

After a brief stay at the hospital and sometimes a rehabilitation facility, the patient will walk briefly with the help of a walker, crutches or a cane and finally the great majority will walk freely without assistance. This operation together with the total knee arthroplasty operation, are considered by some to be the greatest surgical developments of the twentieth century, because of the number of patients who have benefited and the severity of the pain that has been alleviated.

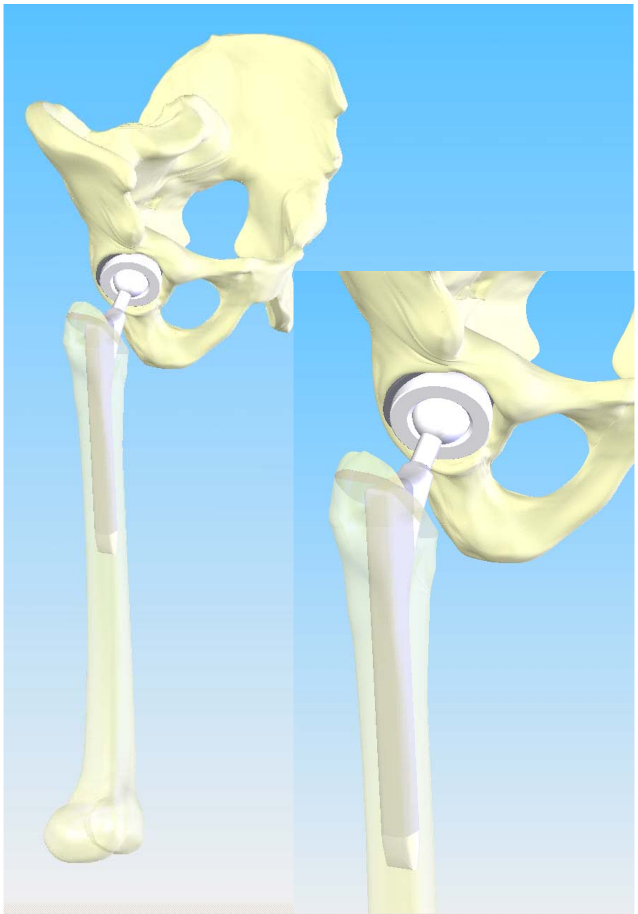


Figure 1. Femoral part of hip prosthesis

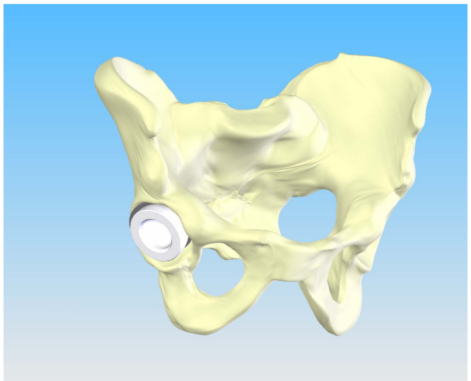


Figure 2. Acetabular part of hip prosthesis

III. PRECISION ENGINEERING TOOLS FOR TESTING COMPUTER ASSISTED ORTHOPAEDIC HIP SURGERY CAOHS SYSTEMS

Precision and robotic engineers have a need for high performance ball and socket joints, which have no backlash and low friction, so they invented the magnetic ball and socket joint shown in Figure 3. The basic component of this device is the magnetic socket shown in Figure 4 [10^1]. This device is usually made of magnetic stainless steel and has a cylindrical hollow cavity at its center. A cylindrical magnet is fitted in that cavity and secured at the desired position with plastic shims and epoxy glue. The image at the top of Figure 4 shows a socket fitted with a magnet, while the image below shows a socket before the attachment of the magnet. This design allows for the control of the magnetic force by selecting the proper magnet and shim thickness for the application. The shims control the size of the gap between the top of the magnet and the surface of the ball. The ball touches the rim of the socket at three small arcs located 120^0 from each other (see images on Figure 4). These arcs are created by pressing hard another ball on the rim of the socket. The socket joint ball is usually made of magnetic stainless steel and it is attracted to the socket by the force of the magnet. This force should be strong enough to keep the ball always in contact with the socket, but not very strong which might generate excessive wear on the ball surface.

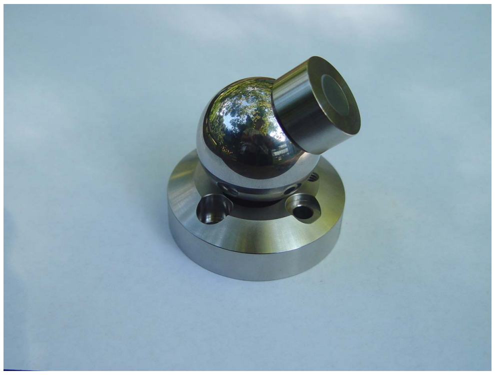


Figure 3. Precision magnetic ball and socket joint

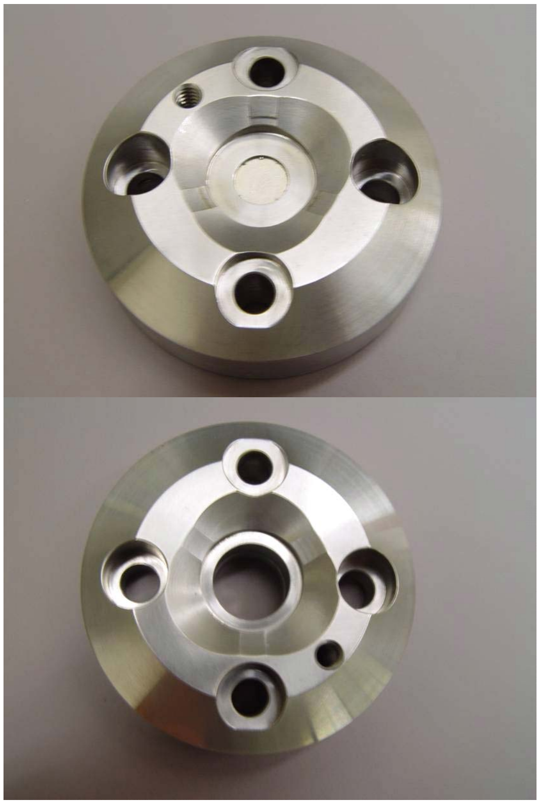


Figure 4. Precision magnetic sockets

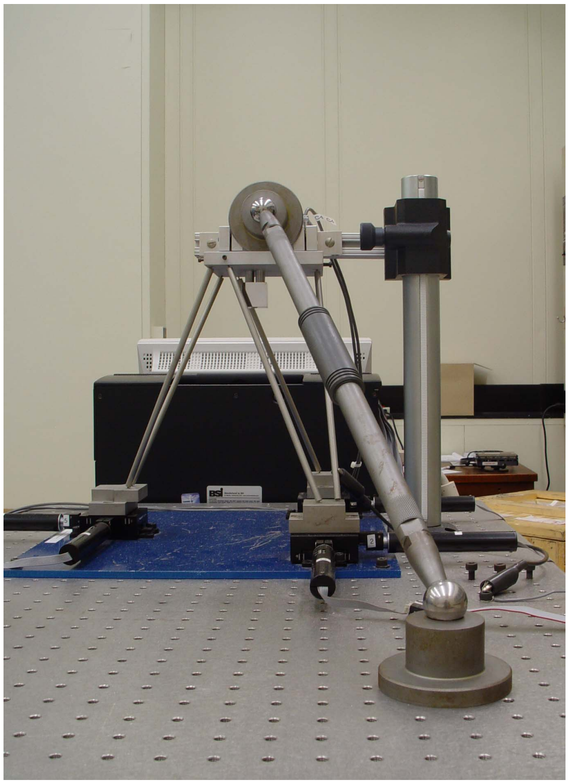


Figure 5. Ballbar calibration test

The magnetic ball and socket device offers a convenient precision joint tool, but many precision and robotic applications require fixed or adjustable length links. To meet that need people have invented the ballbar shown in Figure 5. A ballbar can have fixed or adjustable length and has magnetic balls at both ends, mounted on its tips. In the case of Coordinate Measuring Machines (CMMs), these artifacts can be employed to perform a subset of performance tests that are described in an American National Standard. Although not required these artifacts are frequently calibrated for center-to-center distance. That is, the distance between the centers of the two magnetic balls is independently calibrated. These artifacts are then measured, by the CMM, in several locations and orientations, which were selected because of their sensitivity to error sources associated with the geometrical construction of the particular class of CMM.

For the calibration of the phantom described in this paper, a Direct Computer Controlled (DCC) CMM was employed. This class of machine is error corrected using a high accuracy laser interferometer, electronic levels and precision straightedges. After error correction of the CMM, ballbars are then employed, as described in the American National Standard, to highlight possible problems in the CMM performance before measurement of critical parts are performed.

IV. THE COMPUTER ASSISTED ORTHOPAEDIC HIP SURGERY (CAOHS) ARTIFACT

For best clinical results our artifacts (phantoms) are designed to resemble the skeletal joint or organ, which is the subject of the operation and the suggested performance tests resemble important tasks of the actual surgical operation. In order to reduce the fabrication and maintenance cost of these devices, we use commercially available precision parts wherever possible in the phantom structure design.

The most important component of the hip joint is the ball and socket joint, which we decided to add to our artifact (phantom). Most ordinary mechanical ball and socket joints have backlash and are difficult to clean and inspect for wear, because they are sealed. However precision engineers use magnetic ball and socket joints (see Figure 3) and bars (see Figure 5), which have none of the above mentioned drawbacks and are commercially available for reasonable prices and are used for the calibration and testing of precision measurement machines, like CMMs and Industrial Robots (IRs). Furthermore these joints can be fitted with various strength small size magnets, which can be selected for the proper size bar and joint orientation, so that the contact force will be sufficient to ensure that the bar will not separate from the joint socket during the test and not so large that results in excessive surface wear.

Our first phantom resembles a pelvis coordinate frame, as shown in Figure 6 and a femur bone connected with a precision magnetic ball and socket joint, as shown in Figure 7. Because the magnetic socket of this device is horizontal it is called Horizontal Joint-Operating Room-CAOHS (HJ-OR-CAOHS).

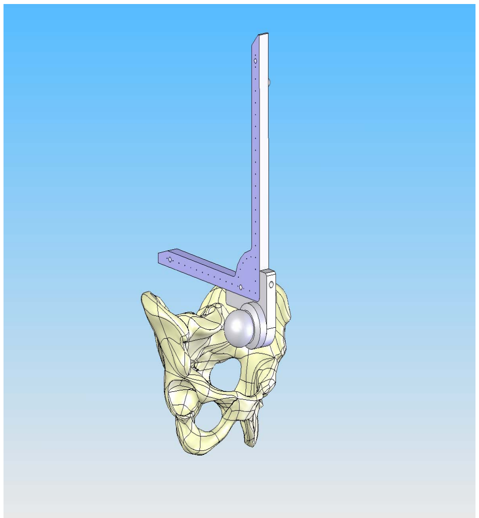


Figure 6. The CAOHS phantom coordinate frame superimposed on a pelvis model

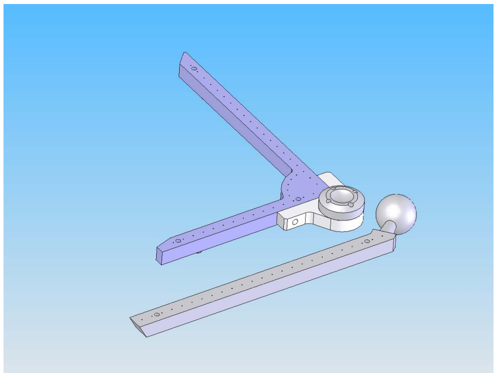


Figure 7. The CAOHS phantom coordinate frame and femur bar connected with a magnetic ball and socket joint

The CAOHS phantoms are designed to perform at least three performance tests relevant to hip arthroplasty operations. Such as are the following: 1) measure the CAOS system accuracy of the determination of the location of the coordinates of the center of rotation of the hip joint, represented here by the precision magnetic ball and socket joint, 2) measure the CAOS system accuracy of moving along straight lines at distances comparable to the size of human adult large bones, along two orthogonal directions, 3) measure the CAOS system accuracy of angular moves relevant to orthopaedic hip surgery. If the CAOHS phantoms prove useful for orthopaedic operations, similar devices will be developed for the human knee joint, shoulder joint, etc.

The first HJ-OR-CAOHS phantom was fabricated a few months ago (see Figure 8). It is made of an L shape horizontal XY orthogonal coordinate frame, a joint horizontal mount, the magnetic ball and socket joint and a femur bar. The XY coordinate frame has small target holes (see Figures 13 and 14) at regular intervals of 15 mm, designed to fit the pointed probe tip of the CAOS systems target assemblies. These are plates with four or more active or passive markers, which can be mounted on surgical tools. It also has two larger holes for the mounting of DRB target assemblies. The femur bar also has two larger holes for the mounting of DRB target assemblies, which can be used for the determination of the coordinates of the ball center of rotation. The tips of all the HJ-OR-CAOHS phantom bars are machined to form various angles, which are useful for hip arthroplasty operations (see Figures 11 and 12). An arc at the base of the coordinate frame has been fitted with target holes spaced at regular angular increments, which adds an additional angular calibration and testing capability (see Figure 9). The magnetic ball and socket joint are commercially available and are made of stainless steel material, while the rest of the parts are made of Invar, for better thermal stability inside an operating room.

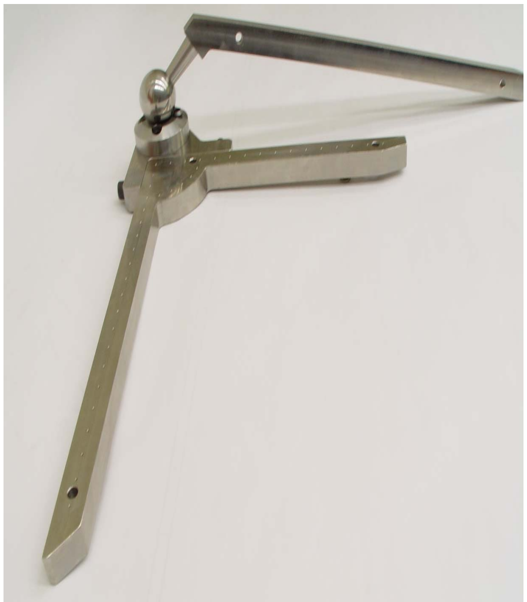


Figure 8. The first prototype of the HJ-OR-CAOHS phantom

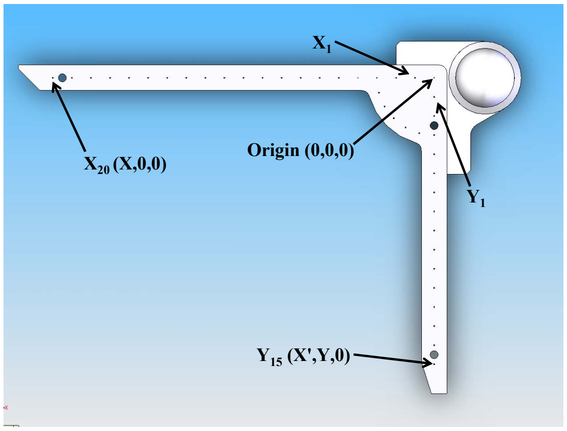


Figure 9. The L shape XY coordinate frame with the target holes

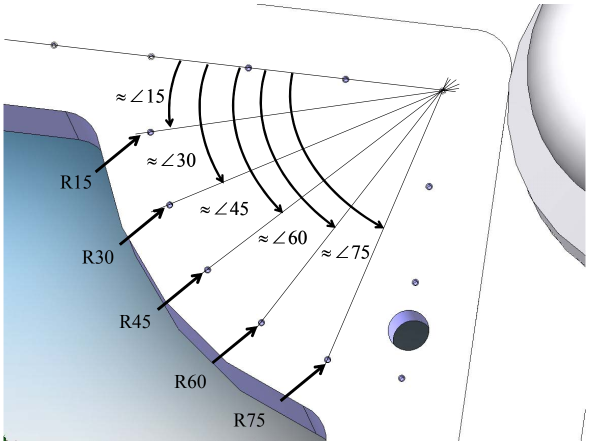


Figure 10. The origin arc with the target holes defining certain angles with respect to the X coordinate axis

Figure 9 shows the L shape XY coordinate frame with its target holes marked X_1 to X_{20} on the X axis (horizontal in the figure) and Y_1 to Y_{15} on the Y axis (vertical in the figure). The nominal incremental distance between these target points is 15 mm, which gives a nominal X axis length of 300 mm and a nominal Y axis length of 225 mm (see Figure 11). The X axis is longer because it is intended to approximate the length of an adult femur bone. The distance between any two target holes is measured between the tips of the two holes. Although the nominal distance can be calculated assuming a nominal increment of 15 mm, between neighboring holes, the actual distance is determined through careful calibration, which will be described in a future paper.

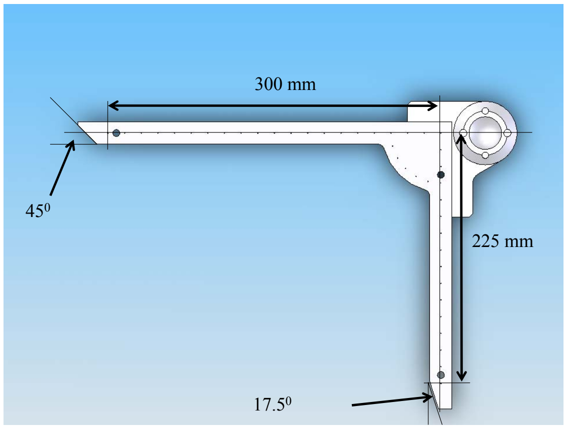


Figure 11. The angles between the adjacent planes labeled in the figure can be used for the evaluation of surgical cutting tools

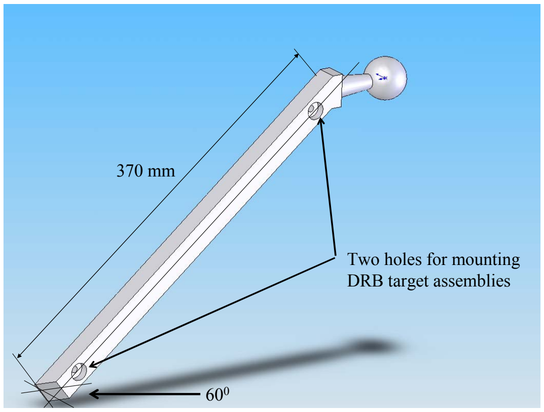


Figure 12. The femur bar showing the two angled planes and DRB mount holes

Figure 13 shows the nominal dimensions of the target holes. Special attention was given to the drilling of these holes in order to achieve smooth clean hole walls and tip and a hole axis, which is as close as possible orthogonal to the corresponding coordinate frame XY axis. Several drill bits were used and each one was not used for more than four holes. Every single one of the target holes was examined and photographed under a microscope. One concern was the presence of burrs, which could prevent the tip of the CAOS system target probe from reaching the tip of the target hole. Figure 14 shows a typical hole image, which reveals that the hole tip is really a hemispherical surface and not a sharp tip as Figure 13 implies. It is thus important that during CAOS testing the pointed probe tip of the CAOS systems target assemblies can reach that hemispherical surface and not be able to move laterally by any significant amount because that motion will introduce measurement errors.

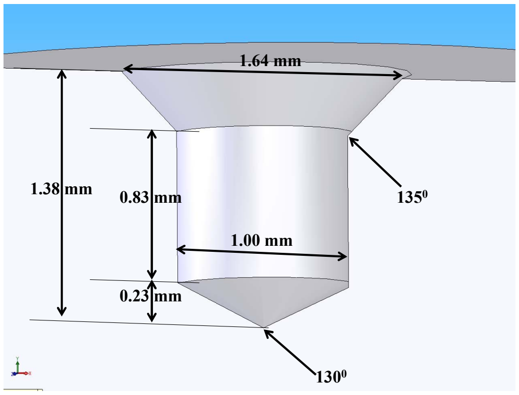


Figure 13. Cross section view of the target hole

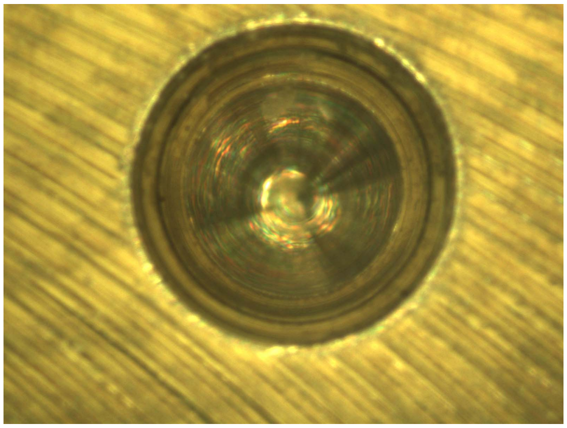


Figure 14. Microscope images of a target hole

The HJ-OR-CAOHS phantom offers two different options for testing the ability of a CAOS system to measure angles. One may involve the use of the saw blade, spatula or other similar tool and the other the pointed probe tip of the CAOS systems target assemblies. The end planes of all the phantom bars are shaped to form angles that are commonly used during hip orthopaedic operations. From Figure 11 it can be seen that the X axis bar of the phantom coordinate frame terminates at a nominal 45^0 angle, which is considered by many orthopaedic surgeons as the best choice for the hip acetabulum prosthesis inclination angle. The Y axis bar of the phantom coordinate frame terminates at a nominal 17.5^0 angle, which is considered by many orthopaedic surgeons as the best choice for the hip acetabulum prosthesis anteversion angle. The femur bar terminates at a nominal 60^0 angle, which is preferred by many orthopaedic surgeons for the decapitation of the damaged head of the femur bone. The arc around the origin of the coordinate XY frame axes has five target holes at nominal angles of 15^0, 30^0, 45^0, 60^0, 75^0, with respect to the X axis. These are three point angle measurements and allows pointed probe tip measurement tests.

NIST staff have calibrated all the critical features on the HJ-OR-CAOHS using an industrial grade Coordinate Measuring Machine (CMM). These features include the target hole locations and the center of rotation. In all cases the expanded uncertainty U with k = 2 in the determination of the three dimensional coordinate is less than 0.08 mm. A future publication will report on the calibration procedures and an additional publication will describe the results of industrial testing. The coordinates of the ball center of rotation are measured with respect to the CMM reference coordinate frame. Using coordinate transformation algorithms similar to those used for the calibration of IR work cells, it is possible to refer these coordinates to the HJ-OR-CAOHS phantom coordinate frame, thus making the use of the phantom independent of the metrology instrument used for its calibration.

A new version of the OR-CAOHS, which has an angled magnetic ball and socket joint similar to that of a human pelvis, is also being designed.

V. CONCLUSIONS

We have described the use of common and inexpensive precision engineering and industrial robot calibration tools for the design of an artifact (phantom), which may be used for measuring the performance of CAOS systems inside operating rooms. This phantom can also be used for the calibration of CAOS systems. Calibration is of course primarily the responsibility of the manufacturer of CAOS systems and it can be performed after fabrication and during servicing operations. We have designed and fabricated a horizontal joint computer assisted orthopaedic hip surgery phantom (artifact). This device appears to be working very well and it was recently calibrated and sent to a medical research group for testing. Calibration and testing results will be reported in future publications.

ACKNOWLEDGEMENTS

We would like to thank Dr. J. B. Stiehl, Dr. D. A. Heck and Dr. W. L. Bargar, for their valuable orthopaedic surgery advice and encouragement. We would also like to thank Mr. Damien Bertot for the laser scanning of several human bone models, Dr. Junhwan Kim for generating solid CAD models of individual bones and assemblies, and Mr. Marvin Kingsbury for the fabrication of the HJ-OR-CAOHS phantom.

REFERENCES

[1] Selspot[1] manufactured by SELCOM, Partille, Sweden (presently owned by LMI, Delta, British Columbia, Canada).

[2] Dainis A., Juberts M., "Accurate Remote Measurement of Robot Trajectory Motion," IEEE International Conference on Robotics and Automation, St. Louis, Missouri, pp 92-99, March 1985.

[3] Nolte L.P., Zamorano L., Visarius H., Berlemann U., Langlotz F., Arm E., Schwarzenbach O., "Clinical Evaluation of a System for Precision Enhancement in Spine Surgery," Clinical Biomechanics, Vol. 10, No. 6, pp.293-303, 1995.

[4] Lavallee S., Sautot P., Troccaz J., Cinquin P., Merloz P., "Computer Assisted Spine Surgery: A Technique for Accurate Transpedicular Screw Fixation Using CT Data and a 3-D Optical Localizer," Journal of Image Guided Surgery, Vol. 1, pp. 65-73, 1995.

[5] Nolte L.P., Langlotz F., "Basics of Computer-Assisted Orthopaedic Surgery (CAOS)," in the book "Navigation and Robotics in Total Joint and Spine Surgery," by Stiehl JB, Konermann WH, Haaker RG, Publisher Springer-Verlag, 2004.

[6] Haaker R.G.A., Tiedjen K., Ottersbach A., Rubenthaler F., Stockheim M., Stiehl J.B., "Comparison of Conventional Versus Computer-Navigated Acetabular Component Insertion," J of Arthroplasty, Vol. 22, No. 2, 2007.

[7] Nogler M., Kessler O., Prassl A., *et al.* "Reduced Variability of Acetabular Cup Positioning with Use of an Imageless Navigation System," Clin Orthop , Vol. 426, pp 159-163, 2004.

[8] Thomas B.J., Stiehl J.B., "Basics of Total Hip Replacement Surgery," in the book "Navigation and Robotics in Total Joint and Spine Surgery," by Stiehl JB, Konermann WH, Haaker RG, Publisher Springer-Verlag, 2004.

[9] Bargar W.L., Bauer A., Borner M., "Primary and Revision Total Hip Replacement Using the Robodoc System," Clinical Orthopaedics, Vol. 354, pp. 82-91, 1998.

[10] ATT Metrology Services, Redmond, WA, http://www.attinc.com/target-pg5.htm

Haptic Feedback System for Robot-Assisted Surgery[#]

Jaydev P. Desai*[a], Gregory Tholey[b], and Christopher W. Kennedy[c]
*Robotics, Automation, Manipulation, and Sensing (RAMS) Laboratory
University of Maryland, College Park

Abstract— **Minimally invasive surgical procedures using long instruments have profoundly influenced modern surgery by decreasing invasiveness, therefore minimizing patient recovery time and cost. However, surgical procedures using long tools inserted through small ports on the body deprive surgeons of the sense of touch (haptics), depth perception, dexterity, and straightforward hand eye coordination that they are accustomed to in open procedures. While there have been significant advances in almost all of the above areas, haptic feedback systems for robot-assisted surgery are lacking in development. In this paper we present: 1) the development of accurate robot-arm dynamic model (using model-based control) with the goal of minimizing unwanted tool-tissue interaction forces in robot-assisted surgery, 2) the development of an ergonomic 7-DOF haptic feedback system, and 3) the recently developed laparoscopic grasper with force feedback capability attached to the end of the robot arm and controlled by the haptic device.**

***Index Terms*— Mitsubishi PA-10, Haptic Device, Laparoscopic Grasper, Robot-Assisted Minimally Invasive Surgery.**

I. Introduction

ROBOT-assisted surgery has led to significant improvement within the medical field. These systems incorporate advantages from minimally invasive surgery (MIS), such as reduced patient trauma, recovery time, and lower health care costs, to name a few. In a surgical environment it is essential to have low interaction forces with the tissue and/or organ to prevent unwanted harm to the patient and the surgical staff. As a result, we developed a dynamic model of the Mitsubishi PA-10 robot arm for low velocity applications such as surgical tool placement or teleoperated soft tissue manipulation. The PA-10 is ideal for precise manipulation tasks due to the backdrivability, accurate positioning capability and zero backlash afforded by its harmonic drive transmission. However, the compliance and oscillations inherent in harmonic drive systems make the development of an accurate dynamic model of the robot extremely challenging. The Mitsubishi PA-10 robot is significantly used in research laboratories worldwide [1, 2] and in our prior work, we have addressed [3] the transmission modeling and low velocity, low impedance implementation for the PA-10 robot arm in a research environment.

Current robotic surgical systems, such as Da Vinci Surgical System (Intitutive Surgical Inc.) do not provide haptic feedback to the surgeon. This lack of haptic feedback has led several researchers to develop haptic devices for surgical and various applications. Massie and Salisbury [4] developed the Personal Haptic Interface Mechanism (PHANToM™), which is commercially available and used for many different applications. Additional mechanisms that use serial or parallel configurations have also been developed [5-8]. Serial mechanisms, such as the PHANToM, lack a sufficient force feedback capability without adding significant weight and inertia to the mechanism and typically do not have a grasping interface capable of providing force feedback. Parallel mechanisms can provide sufficient force; however, they have a smaller workspace and also lack a grasping interface. Therefore, a need exists for the development of a surgical haptic interface that can reflect forces for some of the robotically-assisted surgical procedures. Based on this motivation, we have developed a haptic device with seven degrees of positional feedback capability and four degrees of force feedback capability.

Several researchers have developed novel surgical tools to accurately measure the tool-tissue interaction forces during surgical procedures. One area of research involves solutions that incorporate sensors into current laparoscopic tools using strain gages, force/torque sensors, or custom designed sensors on the shaft or jaws of the tool to measure tool-tissue interaction forces. Morimoto et al [9] and Bicchi et al [10] implemented strain gage sensors on the tool shaft that allow for measurement of indirect grasping forces and surgical manipulation forces, respectively. Dargahi et al [11] utilized a MEMS-based approach to measure normal forces at the jaws, however, cost and sterilizability issues were not discussed. Prasad et al [12] developed a 2-DOF force sensing sleeve to measure bending forces in 5 mm laparoscopic instruments. While these various designs can accurately measure the surgical forces, they have disadvantages towards incorporating

[#]This work was supported in part by National Science Foundation grants EIA-0711040, and CAREER award IIS-0711038 and the American Heart Association grant 0160368U.

[a] Corresponding author. J. .P. Desai is with the Department of Mechanical Engineering, University of Maryland, College Park, MD 20742 USA (email: jaydev@umd.edu)

[b]G. Tholey is with Conferno Solutions, King of Prussia, PA, USA (email: gtholey@conferosolutions.com)

[c]C. W. Kennedy is with the John Hopkins University, Baltimore, MD USA (email: ckenne10@jhmi.edu)

1. Portions reprinted, with permission, from "Gregory Tholey and Jaydev P. Desai, *A Modular, Automated Laparoscopic Grasper with Three-Dimensional Force Measurement Capability*, page 250-255, 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 2007". © [2007] IEEE.
2. Portions reprinted, with permission, from "Gregory Tholey and Jaydev P. Desai, *A General Purpose 7 DOF Haptic Device: Applications towards Robot-Assisted Surgery*, Accepted for publication in IEEE/ASME Transactions on Mechatronics, To appear in 2007". © [To appear in 2007] IEEE.
3. Portions reprinted, with permission, from "Christopher Kennedy and Jaydev P. Desai, *Model-Based Control of the Mitsubishi PA-10 Robot Arm: Application to Robot-Assisted Surgery*, 2004 IEEE International Conference on Robotics and Automation, vol. 3, pp. 2523-2528, New Orleans, LA, 2004". © [2004] IEEE.

them into an actual surgical setting. Previously developed surgical instruments with force measurement capabilities [12-15], have the disadvantage of costly, non-disposable sensors, large jaw designs, and low degrees-of-freedom for force measurement. In addition, most of the previous research lacks modularity for easy conversion between tool types (e.g. grasper, cutter, and dissector) without removing the entire surgical tool. Based on this motivation, we have developed a modular and automated laparoscopic grasper with tri-directional force measurement capability and a modular, disposable tool shaft for quick conversion between surgical modalities, such as grasping, cutting, and dissection. The current prototype has incorporated the advantages of previous graspers in a compact design for use in a clinical setting

The paper is organized as follows: In section 2, a dynamic model of the Mitsubishi PA-10 robot arm is presented. In section 3, the development of a haptic device with seven degrees of freedom is presented and in section 4, the automated laparoscopic grasper with tri-directional force measurement capability is presented. Finally in section 5 concluding remarks are presented. Our overall research goal is the development of a haptic feedback surgical system that uses a robotic arm with an attached laparoscopic tool to perform surgical procedures and have the capabilities of measuring the tool-tissue interaction forces through a haptic feedback interface (see Fig. 1).

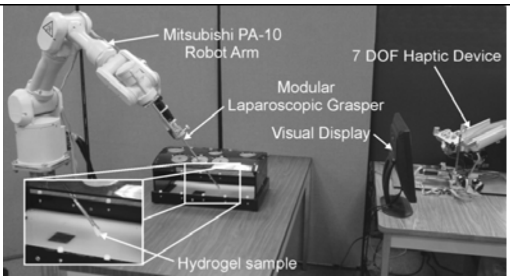


Fig.1. Telemanipulation setup for characterizing tissue using a haptic device to control the robot arm with attached laparoscopic grasper. © [2007] IEEE.

II. The Mitsubishi PA-10 Robot Arm

The Mitsubishi PA-10 robot arm is a 7 degree-of-freedom robot arm with open control architecture and is manufactured by Mitsubishi Heavy Industries (see Fig. 2a). The four layer control architecture is made up of the robot arm, servo controller, motion control card, and the upper control computer.

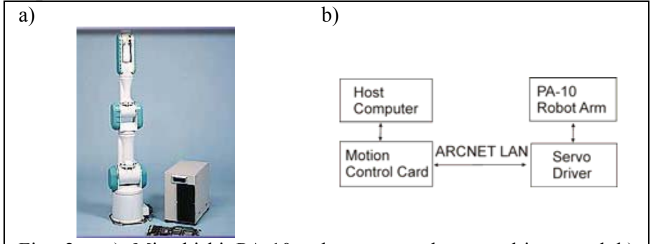


Fig. 2. a) Mitsubishi PA-10 robot arm and servo driver, and b) Flowchart of Mitsubishi PA-10 four layer control architecture. © [2004] IEEE.

A flow chart for the control system is shown in Fig. 2b. The host computer runs the QNX real-time operating system and we have been able to achieve communication rates of up to 700 Hz with the robot servo driver through the ARCNET (ARCNET is a token passing LAN protocol developed by Datapoint Corporation) motion control card and custom-made software. The robot joints are actuated by three-phase AC servo motors and harmonic gear transmissions. Joint positions are measured through resolvers at the joint output axis, with a resolution of 0.000439^o over +/- 3 output revolutions. Control of the robot can be achieved in either 'Velocity mode' or 'Torque mode'. HDT provides advantages such as zero backlash due to natural pre-loading. However, there are also several disadvantages such as non-linearity due to friction, alignment error of the components, and transmission losses due to the compliance in the system. All of these were found to be critical in the modeling of the Mitsubishi PA-10 robot arm. In the following subsections we will describe in detail our methodology to estimate and model the: a) velocity dependent and position dependent friction, b) torsional stiffness, and c) gravity effects. The above items (a) through (c) comprise the model of the HDT.

A. Modeling

Parameter identification of the Mitsubishi PA-10 robot arm was carried out using the system and control architecture described above. Although the system did allow us to control the motor torque of each joint, we preferred to conduct our experiments in 'Velocity mode' when possible because this allowed for better high-gain trajectory tracking due to the increased feedback loop rate.

1) Harmonic Drive Model

We consider the model of the harmonic drive to be composed of friction, gravity, and stiffness. The non-linear expression for torque transmission in harmonic drives is thus given by:

$$T_{in} N = T_{cf}(\theta) + T_{vf}(\dot{\theta}) + T_{g}(\theta) + T_{c}(T_{cf}, T_{vf}, T_{g}) \qquad (1)$$

where T_{in} is the input torque, N is the transmission ratio (N is 50 for all joints), T_{cf} is the coulomb friction, T_{vf} is the velocity dependent friction torque, T_g is the gravity torque, and T_c is the torque used to deform the wave generator. A schematic for the proposed control system is shown in Fig. 3.

2) Friction

(a) Velocity-dependent friction:

To determine the friction-velocity relationship for the joints of the PA-10, each joint of the robot was commanded to move at a constant velocity and the mean torque required to maintain the velocity was taken to be the friction for that value of velocity. To characterize the friction behavior at low velocity, data for velocities between 0.02 rad/s and 0.1 rad/s were collected in 0.02 rad/sec increments and between 0.1 rad/sec and 0.4 rad/sec in 0.1 rad/s increments. Five trials were performed for each velocity value in both the positive and negative directions, for a total of 80 measurements per joint. For joints 2, 4, and 6 (which are influenced by gravity), the robot was mounted on the wall to negate the effect of gravity. Data for joints 1, 3, 5, and 7 were collected with all the joints in the vertical position. After collecting data for all 7 joints, we fit three different friction models using least-squares techniques. The three models tested were: 1) kinetic plus viscous friction model, 2) cubic polynomial model, and 3) Stribeck curve model. The results of this analysis are shown in Table 1. We compared different friction models based on their ability to fit experimental data. The results presented in Table 1 are the weighted residual variance and the mean-squared error (MSE) per degree-of-freedom (DOF) for each model after being fit to the collected experimental data. The Stribeck model provided a reasonably good approximation for the friction torque in all 7 joints of the Mitsubishi PA-10 robot arm that we have modeled.

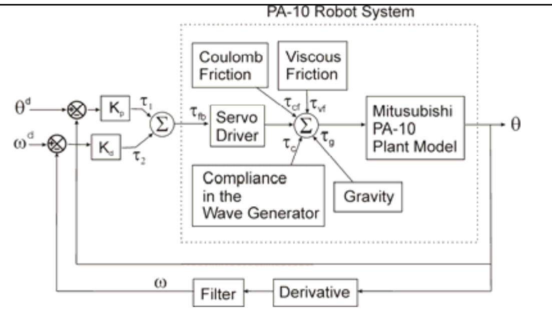


Fig. 3. The proposed control system includes models of friction, gravity, and nonlinear stiffness of the wave generator. © [2004] IEEE.

The expression for this model is given by:

$$T_{vf}\,(\dot{\theta}) = F_O \operatorname{sgn}(\dot{\theta}) + F_V\dot{\theta} + F_S\,(1 - e^{\left(-\frac{\dot{\theta}}{V_c}\right)}) \qquad (2)$$

where T_{vf} is the viscous friction torque, F_o, F_v, F_s and V_c are the Stribeck coefficients, and $\dot{\theta}$ is the rotational velocity. The best fit approximation of experimental data by a Stribeck curve is shown in Fig. 4 for both positive and negative velocities for joint 1. This approach was used for all the velocity and position calculations in this paper.

(b) Position-dependent friction:

Friction in the HDT is strongly position dependent due to kinematic error in the transmission. The error signature can display frequency components at two cycles per wave-generator revolution and several subsequent harmonics. Based on the above, the error function including two harmonics of wave generator rotation can be expressed as:

$$\theta_{erfn} = A_1\sin(\theta_{wg} + \varphi_1) + A_2\sin(2\theta_{wg} + \varphi_2) \qquad (3)$$

where A_i are the amplitudes of the sinusoids, φ_i is the phase shift, and θ_{wg} is the wave generator position. This expression is of limited use in our case, because it is not possible to measure the output axis rotation. The amplitudes in equation (3) are therefore impossible to accurately determine for our robot. Although kinematic error has a significant effect on the torque transmission characteristics of HDTs, we found that compensating for coulomb friction using the torque required to maintain slow velocity eliminated almost all the effects of kinematic error. Therefore, we neglected the effect of kinematic error in the feedforward implementation of our model. The parameters for periodic torque function for 7 joints are shown in Table 2.

3) *Gravity Compensation*

The parameters used for gravity compensation in our model were taken from the catalog values for the masses and the center of mass locations for the robot links. The effect of gravity was significant only for joints 2 through 6 when the robot was mounted on a pedestal. The gravity torques for joints 2 through 6 were calculated and the catalog values for the link masses and lengths.

4) *Estimation and modeling of nonlinear stiffness*

Harmonic drives exhibit significant compliance when externally loaded. This is apparently due to deformation of the wave generator [16]. Our experimental tests on the Mitsubishi PA-10 robot arm revealed that wave generator compliance has a significant effect on the robot arm dynamics. Since wave generator deformation must be a function of the load on the system, we chose to model this torque as a function of the gravity torque and friction torque. Our methodology for determining the stiffness parameters for joint 4 is described in detail in [17]. Fig. 5 shows the steps in the model identification process for joint 1. The parameters for joints 1 through 4 are given in Table 3, including the slopes of each of the three linear regions, the transition points for the linear regions, and the value for stiffness when the external torque is zero. The effects of stiffness are not significant for joints 5 through 7; therefore we have neglected them in our model.

B. *Experimental Verification of the HDT Model*

To verify our model-based controller for the Mitsubishi PA-10 robot arm, we fed-forward the torques computed by our model to track an end-effector trajectory. The chosen trajectory was a lemniscate in the y-z plane of the base coordinate system given by:

$$x = 0.6,\; y = 0.2\frac{\cos\left(\frac{t}{2}\right)}{\left(1+\sin\left(\frac{t}{2}\right)^2\right)},\; z = 0.1 + 0.4\frac{\sin\left(\frac{t}{2}\right)\cos\left(\frac{t}{2}\right)}{\left(1+\sin\left(\frac{t}{2}\right)^2\right)} \qquad (4)$$

The end-effector position followed the lemniscate position while the end-effector orientation remained constant relative to the base coordinate system. We computed the necessary joint angles for this trajectory using the kinematic parameters and inverse kinematics solution for 6 joints. The results of the end-effector trajectory tracking experiments are shown in Fig. 6. The mean end-effector error was 7 mm, with a maximum error of 42 mm occurring at the beginning of the experiment. After the initial large error, the maximum tracking error was 19 mm.

	Viscous Model			**Cubic Model**			**Stribeck Model**		
Joint	Model DOF	Residual variance, weighted	MSE per DOF	Model DOF	Residual variance, weighted	MSE per DOF	Model DOF	Residual variance, weighted	MSE per DOF
1	2	12.1404	4.0351	4	1.8519	1.9260	4	2.2302	2.1151
2	2	22.2687	6.5672	4	2.8653	2.4327	4	5.2306	3.6153
3	2	114.6001	29.6500	4	16.9548	9.4774	4	2.7248	2.3624
4	2	415.8925	104.9730	4	0.1180	1.0590	4	1.4081	1.7041
5	2	147.7424	37.9256	4	33.4068	17.7034	4	0.6805	1.3403
6	2	1.7577	1.4394	4	0.4013	1.2006	4	0.5518	1.2759
7	2	741.2334	186.3080	4	1.9773	1.9887	4	0.7370	1.3685

Table 1. Data for three different models of velocity dependent friction including the viscous friction model, the cubic model, and the Stribeck model. © [2004] IEEE.

Joint	f_1	A_1	φ_1	f_2	A_2	φ_2
1	15.87	7.50	2.30	47.60	5.05	12.80
2	47.60	6.00	-1.50	310.50	2.50	-6.90
3	39.65	0.70	0.10	239.20	0.55	5.50
4	39.65	5.00	-3.15	262.50	0.10	3.98
5	23.80	0.50	4.95	95.80	0.40	7.50
6	23.80	0.10	-1.50	95.80	0.50	5.50
7	23.80	0.10	-1.00	95.80	0.50	2.50

Table 2. Parameters for periodic friction torque for 7 joints including two harmonics of wave generator rotation. © [2004] IEEE.

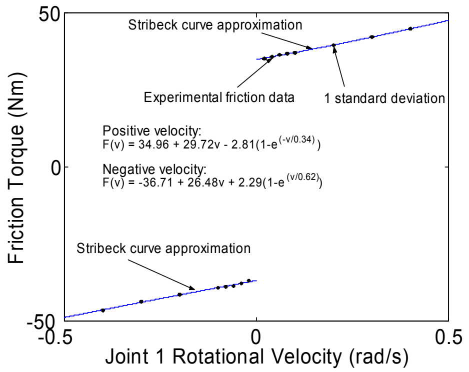


Fig. 4. Represenative plot of stribeck curve fit to velocity-dependent friction data for joint 1. © [2004] IEEE.

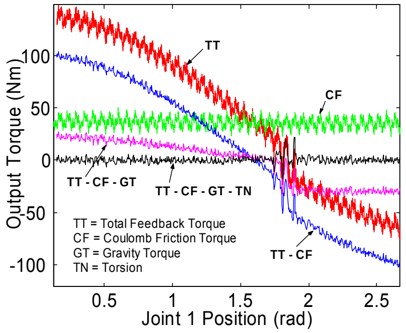


Fig. 5. Representative plot for joint 1 for steps in model identification process. © [2004] IEEE.

Joint	Slope 1 (Nm/Nm)	T_1 (Nm)	Slope 2 (Nm/Nm)	T_2 (Nm)	Slope 3 (Nm/Nm)	Zero Gravity Stiffness (Nm)
Joint 1 +ve	-0.0044	-25.36	2.33	-12.70	0.26	2.32
Joint 1 -ve	0.04	12.16	-1.93	19.53	-0.34	0.19
Joint 2 +ve	0.31	-23.48	2.63	-11.75	0.20	-1.44
Joint 2 -ve	-0.06	11.16	-2.80	21.75	-0.27	0.45
Joint 3 +ve	0.21	-5.78	4.26	-3.49	0.09	0.81
Joint 3 -ve	0.02	5.30	-3.60	2.64	-0.07	-0.30
Joint 4 +ve	0.22	-19.58	5.08	-16.27	0.92	2.57
Joint 4 -ve	-0.68	15.14	-4.11	19.23	-0.18	-1.45

Table 3. Parameters for nonlinear stiffness as a function of gravity torque for joints 1 through 4. © [2004] IEEE.

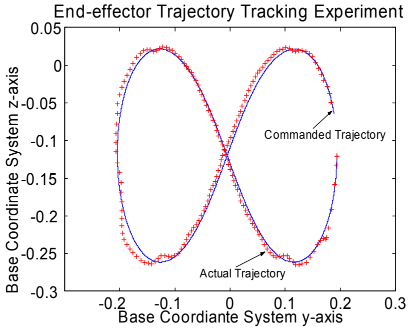


Fig. 6. End-effector trajectory tracking experiment. © [2004] IEEE.

III. 7 DOF Haptic Device

We have developed a haptic device with seven degrees of positional feedback capability and four degrees of force feedback capability. It is a closed-kinematic chain that consists of a user interface and spatial mechanism connected via a universal joint. The haptic device provides force feedback along three orthogonal axes and also the grasping /parting force. This device is part of an overall haptic feedback system (see Fig. 1). Through the haptic device, we will be able to control the robot arm with attached surgical tool that is capable of measuring the forces at the end-effector in 3D [15]. The user interface (see Fig. 7) of the haptic device will control the slave robot, such as the Mitsubishi PA-10. The surgical tool attached to the robot arm can measure and feedback forces to the haptic device to reflect them to the user through the spatial force feedback mechanism (see Fig. 7), which would include forces in X, Y, and Z direction, in addition to the grasping force through the grasping mechanism (θ) (see Fig. 8). This device can also be used for a variety of other applications such as the automotive industry, gaming industry, or as a rehabilitation aid for people with finger, hand, and/or forearm injuries.

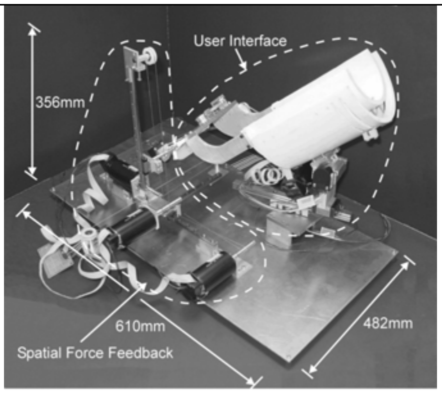


Fig. 7. Prototype of the haptic device. © [To appear in 2007] IEEE.

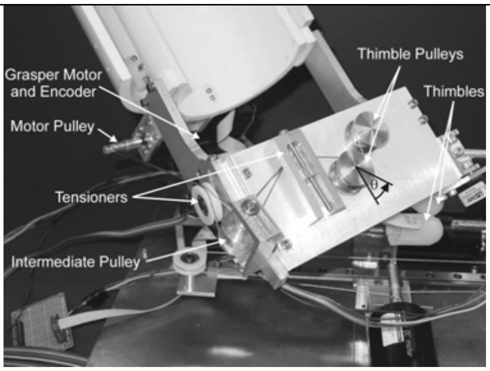


Fig. 8. Details of the grasping mechanism. © [To appear in 2007] IEEE.

The three degree-of-freedom spatial force feedback mechanism mounts to the base of the haptic device and attaches to the user interface at the grasping mechanism via a universal joint (see Fig. 9). It consists of 3 direct drive DC motors with encoders (manufactured by Maxon, Burlingame, CA) for the X, Y, and Z directions and three linear slide guides (manufactured by Misumi, Schaumburg, IL) that act as prismatic joints. The three slide guides are mounted in series and orthogonal to each other and thus create a three degree-of-freedom mechanism for position and force feedback.

The maximum force output of each joint of the spatial mechanism is governed by the motor characteristics, pulley dimensions, and frictional losses. The direct drive DC motors are capable of providing up to 181 mNm of continuous torque. In addition, all three motor pulleys on the spatial mechanism consist of a 6.35 mm diameter grooved pulley. Therefore, the motor/pulley system can theoretically produce approximately 56N of force. However, frictional losses reduce this number to approximately 40N (as measured experimentally on each axis).

A. Kinematics and Workspace

The haptic device is designed as a closed kinematic chain with a universal joint connecting the spatial force feedback mechanism to the user interface. Therefore, the kinematics of the haptic device can be decoupled into two separate halves that both end at the universal joint. These kinematic equations can then be used to the find the position of the prismatic joint on the user interface. The position of the prismatic joint can be mapped to the corresponding translation of the end-effector of the slave robot in the global coordinate frame (mapping not described in this paper). The movement of the grasping mechanism correlates to the opening/closing of the jaws of the laparoscope. Starting with the forward kinematics of the user interface, we placed coordinate frames on both halves of the haptic device. Next, we obtained the D-H parameters of each half that are shown in Table 4 and Table 5. The reachable workspace of the haptic device is the intersection of the workspace of the user interface half and the workspace of the spatial force feedback mechanism. Therefore, we developed an algorithm to determine this volume to verify the reachable workspace of the haptic device was sufficient for the range of motion in MIS procedures. This reachable workspace represents an estimated volume of 0.0041 cubic meters with dimensions of 0.1905m wide by 0.1905m deep by 0.1143m high as shown in Fig 10.

IV. AUTOMATED LAPAROSCOPIC GRASPER WITH THREE-DIMENSIONAL FORCE MEASURING CAPABILITY

We have developed a modular and automated laparoscopic grasper with tri-directional force measurement capability and a modular, disposable tool shaft for quick conversion between surgical modalities, such as grasping, cutting, and dissection. The current prototype has incorporated the advantages of previous graspers in a compact design for use in a clinical setting. The design of the laparoscopic grasper was guided by our previously designed automated laparoscopic graspers and the advantages they incorporated, such as low backlash, compact design, and tri-directional force measurement capability [15]. In addition to these characteristics, the current prototype was significantly improved by using smaller sensors, a significantly smaller shaft diameter (~8 mm), a linear actuation mechanism requiring no tensioning, and a disposable, modular instrument for easy conversion between surgical modalities.

Our design consists of two components; namely, the actuation mechanism and the modular tool (see Fig. 11). The actuation mechanism uses a DC motor with gearbox and encoder that drives a leadscrew and linear positioning assembly (see Fig. 12). This linear positioning assembly connects to a push-rod that is part of the modular tool. The push-rod is contained within and translates along the shaft of the tool and actuates the two jaws of the tool using a linkage. The design uses 4 strain gages mounted on the shaft of the tool near the end of the shaft closest to the jaws to measure the horizontal and vertical forces exerted on the tool end-effector (see Fig. 13). Additionally, a small resistive force sensor is mounted in one of the jaws to measure the normal force during grasping and palpation tasks. The resistive sensor (SF-4 model, manufactured by

CUI, Inc) has an overall size of 5mm long by 5 mm wide with a height of 1mm.

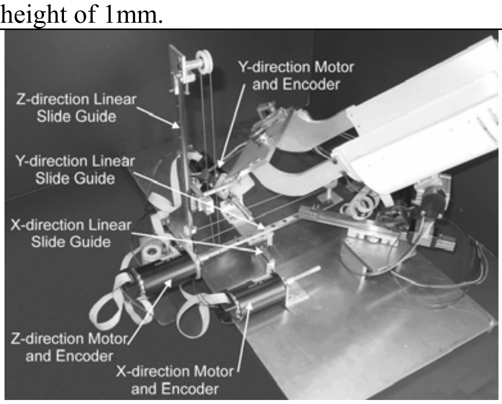

Y-direction Motor and Encoder
Z-direction Linear Slide Guide
Y-direction Linear Slide Guide
X-direction Linear Slide Guide
Z-direction Motor and Encoder
X-direction Motor and Encoder

Fig. 9. Details of the spatial force feedback mechanism. © [To appear in 2007] IEEE.

Joint	θ	α	a (mm)	d (mm)
1	θ_1	$\pi/2$	0	38.583
2	$\pi/2$	$-\pi/2$	66.675	d_2
3	θ_3	$\pi/2$	109.55	0
4	θ_4	0	53.772	192.34
5	$-\pi/2$	0	82.98	0

Table 4. D-H parameters for the user interface. © [To appear in 2007] IEEE.

Joint	θ	α	a (mm)	d (mm)
1	$\pi/2$	$\pi/2$	36.627	$\bar{d}_1$
2	$\pi/2$	$\pi/2$	0	$\bar{d}_2$
3	$\pi/2$	0	0	$\bar{d}_3$
4	0	0	57.633	0

Table 5. D-H parameters for the spatial force feedback mechanism. © [To appear in 2007] IEEE.

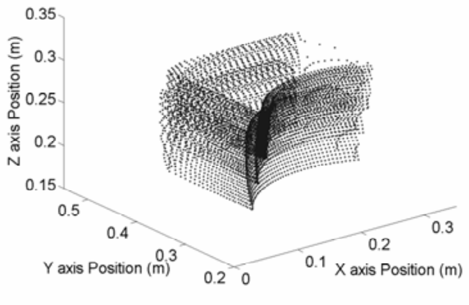


Fig. 10. Reachable workspace of the haptic device. © [To appear in 2007] IEEE.

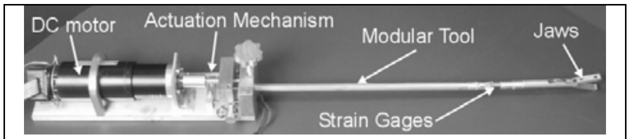


Fig. 11. Prototype of the modular laparoscopic grasper. © [2007] IEEE.

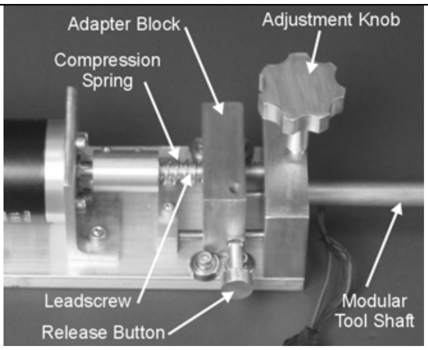


Fig. 12. Actuation mechanism of the modular laparoscopic grasper. © [2007] IEEE.

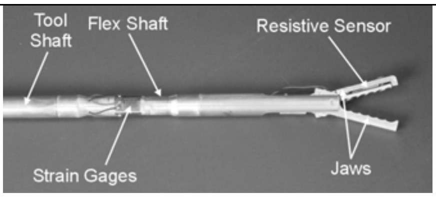


Fig. 13. Sensor locations on the modular tool. © [2007] IEEE.

Some of the characteristics of this sensor are a response time of less than 6 µsec, a maximum load of approximately 29 N, and a resistance range from 10,000 MΩ at zero load to approximately 15 Ω at full load. The four strain gages (model 125UN, manufactured by Vishay Intertechnology, Inc.) have overall dimensions of 3.05 mm wide by 6.99 mm long with a strain range of ±3%. The control of the laparoscopic grasper is achieved using the QNX real-time operating system, data acquisition card, and motor amplifier. This program operates at 500 Hz and implements a PD controller to control the position of the jaws, which is given by:

$$T = K_p\left(q_d - q\right) + K_d\left(\dot{q}_d - \dot{q}\right) \quad (5)$$

where T is the motor torque, K_p and K_d are the proportional and derivative gains, q_d and q are the desired and actual positions of the jaws, and $\dot{q}_d$ and $\dot{q}$ are the desired and actual velocities of the jaws.

A. Calibration

The calibration of the sensors on the prototype laparoscopic tool is required for accurate measurement of the tool-tissue interaction forces. Specifically, the preload in the normal force sensor and the manufacturing tolerances of the prototype make it necessary to calibrate each sensor once it has been placed on the tool. To perform this calibration, an electro-mechanical device that is capable of generating a linear force and recording the values was used (see Fig. 14). Calibration of the resistive sensor for normal force was performed by removing the jaw with the resistive

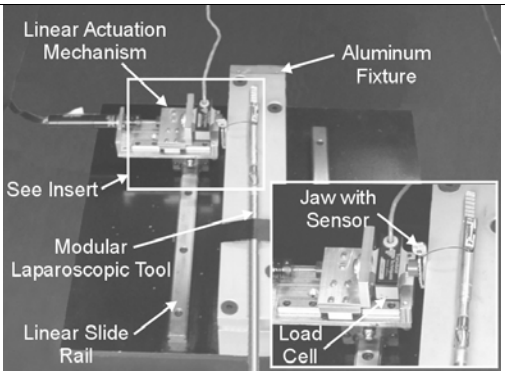


Fig. 14. Electro-mechanical device for calibration of force sensors. © [2007] IEEE.

sensor from the modular tool and attaching it to the side of the aluminum fixture using a cyanoacrylate adhesive (see Fig. 14). Fig. 15 shows the results of this calibration procedure. The resistive sensor measurements have been filtered using a 5th order Butterworth filter to eliminate the high frequency noise. Additionally, a least squares linear regression was used to derive a best-fit mathematical model for the loading and unloading of the calibration curve, given by:

$$\mathrm{F}_{loading} = 0.60x^3 - 2.4x^2 + 4.9x - 0.13 \qquad (6)$$

$$\mathrm{F}_{unloading} = 0.43x^5 - 3.4x^4 + 10x^3 - 13x^2 + 7.6x - 1.2 \quad (7)$$

where $F_{loading}$ and $F_{unloading}$ is the normal force in Newtons for loading and unloading curves respectively.

Calibration of the strain gages was performed by mounting the entire prototype to the aluminum fixture on the mechanical calibration device. The prototype was clamped at the actuation mechanism base with sufficient force to prevent any movement; thus mimicking the constraint in surgery where the tool would be attached to the end-effector of a robot arm. The loading was performed at a rate of 0.1 N/sec at 90° intervals (top, bottom, left, and right) on the jaw (see Fig. 16). The measurements from the load cell and all four strain gages were recorded and plotted to obtain the calibration curve. Fig. 17 shows an example calibration curve for the top loading point with similar plots for each loading point that are shown in Fig. 16. As shown by Fig. 17, a linear relationship exists between the strain gage output and the actual force measured by the load cell. A least squares linear regression was used to derive the best-fit mathematical model of the calibration curve for each strain gage. The models for each of the calibration curves for the top, bottom, left, and right loading points are given by:

$$F_{top} = 59x_{top} + 0.092 \qquad (8)$$

$$F_{bottom} = 61x_{bottom} - 0.11 \qquad (9)$$

$$F_{left} = 59x_{left} - 0.23 \qquad (10)$$

$$F_{right} = 62x_{right} - 0.27 \qquad (11)$$

where F is the magnitude of force in Newtons exerted at the specified loading point and x is the strain gage output in volts at the specified loading point.

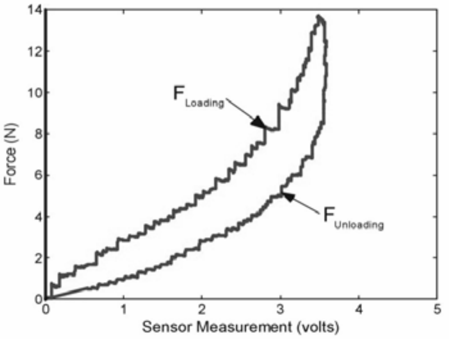


Fig. 15. Calibration curve for the resistive sensor. © [2007] IEEE.

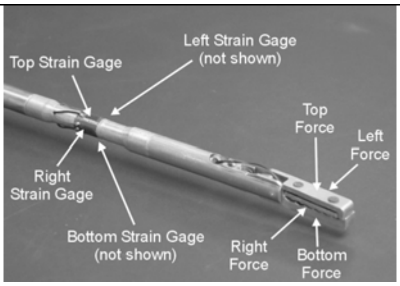


Fig. 16. Location of the strain gages on the flex shaft and loading positions for the strain gage calibration. © [2007] IEEE.

B. Tissue Characterization Experiment

As a demonstration of the capabilities of our modular laparoscopic tool, we have conducted a tissue characterization experiment to evaluate the force measured by the tool when grasping simulated tissue samples of varying stiffness. The simulated tissue samples were made up of Hydrogel material [18]. For this experiment, we selected three Hydrogel samples (corresponding to soft, medium, and hard tissue) that had a significant variation in stiffness and would be easily differentiated with one's fingers. The samples were identical in size and thickness, therefore, the only variable would be the required force to deform the samples by the same magnitude. The experimental setup consisted of using our 7 DOF haptic device (in section III) to control the laparoscopic tool's jaws and grasp each of the simulated tissue samples. As each sample was grasped, the normal force measurement at the jaws and the angle of the jaws were recorded. As shown in Fig. 18, the results show that the laparoscopic grasper can differentiate between samples of different stiffness. All three samples shown were grasped to a jaw angle of approximately 1°, therefore, all incurring the same deformation but a significantly different normal force for each sample. The soft Hydrogel sample showed a maximum force of 0.4 N while the medium Hydrogel sample showed a maximum force of 1 N and the hard Hydrogel sample showed a maximum force of 2.2 N for the same angular displacement of the jaw. Additional tissue grasping trials were performed with similar results for validation. Therefore, the grasper's capability of distinguishing between tissues of different stiffness has been demonstrated.

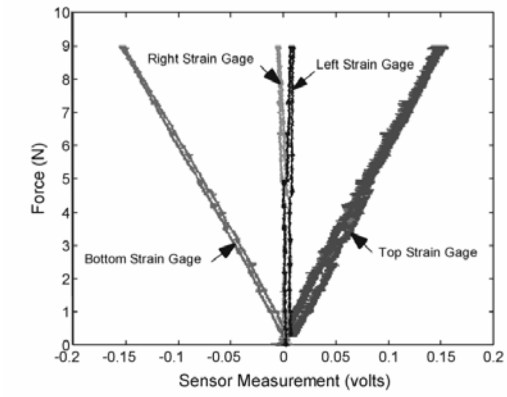


Fig. 17. Strain gage calibration curve for application of a force to the top of the jaws. © [2007] IEEE.

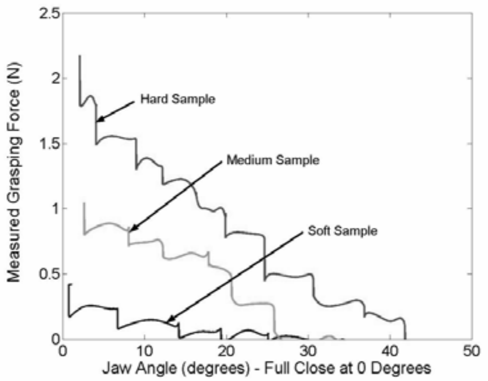


Fig. 18. Characterization of the Hydrogel samples using the modular laparoscopic grasper. © [2007] IEEE.

V. Conclusion

In this paper we presented: (a) a dynamic model of the Mitsubishi PA-10 robot arm for low velocity applications such as surgical tasks, (b) a seven degree of freedom haptic device that can be used for applications in robot-assisted minimally invasive surgery and, (c) a modular, automated laparoscopic grasper with tri-directional force measurement capability. The integration of the robot arm, the haptic device and the laparoscopic grasper form a complete haptic surgical system. Such a system incorporates advantages from minimally invasive surgery, improves dexterity, eliminates surgeon tremor, reduces surgeon fatigue and above all provides haptic feedback to the surgeon. Future work includes telemanipulation experiments that use the 7 DOF haptic device to control PA-10 robot arm with the prototype laparoscopic tool mounted on its end-effector for performing tissue suturing experiments and evaluating the overall system capability.

References

[1] M. M. Olsen and H. G. Peterson, "A new method for estimating parameters of a dynamic robot model," *IEEE Transactions on Robotics and Automation,* vol. 17, pp. 95-100, 2001.

[2] W. K. Yoon, Y. Tsumaki, and M. Uchiyama, "An experimental system for dual-arm robot teleoperation in space with concepts of virtual grip and ball," in *Proceedings of International Conference on Advanced Robotics*, 1999, pp. 225-230.

[3] C. W. Kennedy and J. P. Desai, "Modeling and Control of the Mitsubishi PA-10 Robot Arm Harmonic Drive System," *IEEE/ASME Transactions on Mechatronics,* vol. 10, pp. 263-274, 2005.

[4] T. H. Massie and K. J. Salisbury, "Force reflecting haptic interface," US: Massachusetts Institute of Technology, 1993.

[5] E. L. Faulring, J. E. Colgate, and M. A. Peshkin, "A High Performance 6-DOF Haptic Cobot," in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, 2004, pp. 1980-1985.

[6] K. Kim, W. K. Chung, and Y. Yourn, "Design and Analysis of a New 7-DOF Parallel Type Haptic Device: PATHOS-II," in *IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003, pp. 2241-2246.

[7] L. Birglen, C. Gosselin, N. Pouliot, B. Monsarrat, and T. Laliberte, "SHaDe, A New 3-DOF Haptic Device," *IEEE Transactions on Robotics and Automation,* vol. 18, pp. 166-175, 2002.

[8] J. H. Lee, K. S. Eom, B. J. Yi, and I. H. Suh, "Design of a New 6-DOF Parallel Haptic Device," in *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001, pp. 886-891.

[9] A. K. Morimoto, R. D. Foral, J. L. Kuhlman, K. A. Zucker, M. J. Curet, R. Bocklage, T. I. MacFarlane, and L. Kory, "Force Sensor for Laparoscopic Babcock," in *Medicine Meets Virtual Reality*, 1997, pp. 354-361.

[10] A. Bicchi, G. Canepa, D. DeRossi, P. Iacconi, and E. Scilingo, "A sensor-based minimally invasive surgery tool for detecting tissue elastic properties," in *IEEE International Conference on Robotics and Automation*, 1996, pp. 884-888.

[11] J. Dargahi, M. Parameswaran, and S. Payandeh, "A Micromachined Piezoelectric Tactile Sensor for an Endoscopic Grasper - Theory, Fabrication and Experiments," *Journal of Microelectromechanical Systems,* vol. 9, pp. 329-335, September 2000.

[12] S. K. Prasad, M. Kitagawa, G. S. Fischer, J. Zand, M. A. Talamani, R. H. Taylor, and A. M. Okamura, "A modular 2-DOF force-sensing instrument for laparoscopic surgery," in *International Conference on Medical Image Computing and Computer Assisted Intervention* Montreal, Canada, 2003, pp. 279-286.

[13] G. S. Fischer, T. Akinbiyi, S. Saha, J. Zand, M. Talamini, M. Marohn, and R. H. Taylor, "Ischemia and force sensing surgical instruments for augmenting available surgeon information," in *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* Pisa, Italy, 2006, pp. 1030-1035.

[14] J. Rosen, J. D. Brown, L. Chang, M. Barreca, M. Sinanan, and B. Hannaford, "The BlueDRAGON - A System for Measuring the Kinematics and Dynamics of Minimally Invasive Surgical Tools In-Vivo," in *IEEE International Conference on Robotics and Automation*, Washington, D.C., 2002, pp. 1876-1881.

[15] G. Tholey, A. Pillarisetti, and J. P. Desai, "On-Site Three Dimensional Force Sensing Capability in a Laparoscopic Grasper," *Industrial Robot,* vol. 31, pp. 509-518, 2004.

[16] T. W. Nye and R. P. Kraml, "Harmonic drive gear error: characterization and compensation for precision pointing and tracking," in *Proceedings of 25th Aerospace Mechanisms Symposium*, 1991, pp. 237-252.

[17] C. W. Kennedy and J. P. Desai, "Estimation and modeling of the harmonic drive transmission in the Mitsubishi PA-10 robot arm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, 2003.

[18] G. Tholey, J. P. Desai, and A. E. Castellanos, "Force Feedback plays a sgnificant role in Minimally Invasive Surgery - Results and Analysis," *Annals of Surgery,* vol. 241, p. 102, January 2005 2004.

Prototype Rover Field Testing and Planetary Surface Operations

Edward Tunstel
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA USA
tunstel@robotics.jpl.nasa.gov

***Abstract*—**The success of robotic surface missions on remote planetary surfaces can be facilitated by test campaigns on Earth using a combination of prototype, proto-flight, and space flight hardware. A series of field operations and testing activities were performed using prototype and proto-flight rovers in the years leading up to the NASA Mars Exploration Rovers mission (MER). A brief overview of those activities is presented. Robotic activities and tasks exercised during the Earth-based field trials provided insights for later flight rover operations during the Mars missions. The distribution of mission intelligence and autonomy across a remote human-robot collaborative team is highlighted. Aspects of the mobility and robotic arm performances achieved by the MER flight rovers, spanning several years of surface operations, are summarized.

***Keywords*:** *Mars exploration, field testing, surface operations*

I. Introduction

In January 2004, NASA began a Mars surface mission by landing two spacecraft there, each delivering a rover to explore distinct surface regions on opposite sides of the planet. The first rover, *Spirit*, was landed in the Gusev Crater on Mars and second, *Opportunity*, was landed in an area on Mars called Meridiani Planum. Scientific and technological objectives for the mission are accomplished using the two rovers and their science instrument payloads.

Preparation for and execution of the MER mission involved the use of various rover prototypes, engineering models and, of course, space flight qualified rovers to refine and conduct an effective field operations approach. Ensuring the functionality of rover systems and their operation for remote planetary missions is complicated by a number of challenges. Among them is a need to produce highly robust and/or fault tolerant hardware and software since operational failures cannot easily be addressed at the remote site by humans. It is also a challenge to ensure that the system will work as designed in an environment that is unknown or not well understood prior to the mission. The lack of information about the target environment presents additional challenges associated with anticipating most possible scenarios that could be encountered by the system and therefore dealt with in a safe and acceptable manner. It is often difficult if not impossible to duplicate conditions and scenarios expected in the target planetary environment when testing the system on Earth. Nonetheless, Earth-based tests serve to build confidence in how the designed system will behave and help to reduce or mitigate risks associated with untested modes of the system due to lack of information about the target environment. Depending on the budgets and schedules for development of robotic flight mission hardware, a project may not have sufficient resources or time to perform all tests on the actual flight rover hardware that will be used on the planet surface. In such cases, prototype and proto-flight (engineering model) hardware are used before flight hardware becomes available [1]. However flight-like these systems may be, they can usually be used to exercise or develop operations approaches for some portions of the hardware/software or the mission operations system typically being developed in parallel. Prototype and proto-flight systems enable some of the testing needed to develop and verify operational functionality in outdoor settings while specific testing done to qualify flight hardware for launch, space, and planetary physical environments are done using the actual flight vehicles in the cleanest facilities.

During the years leading up to the launch of the MER vehicles and their journey to Mars, a series of Earth-based field operations campaigns were conducted in desert locations of southwestern USA. The intent was to use MER-like prototype rovers to explore ways of performing remote field science including planning and assessment of rover traverses and robotic arm operations in Earth terrain similar to that expected at the Mars landing sites. A prototype rover called *FIDO* (Field Integrated Design & Operations) and a high-fidelity MER engineering model were used to rehearse mission operability and validate onboard mobility functionality in complex geological settings, respectively. These were end-to-end field trials involving networked operations and command workstations, satellite communications, remote field networking and support equipment, and integrated science instrumentation onboard the rovers. The tests were conducted via satellite from the Jet

Propulsion Laboratory (JPL), hundreds of miles from the test sites. For each field trial, multiple Martian days (sols) of mission-like rover activity sequences were physically simulated with a focus on developing and rehearsing the MER surface mission operations approach. The end-to-end implementation, however, was not an exercise of the many parallel activities involved in the overall MER operations process [2] since the test environment was limited as a mission emulation and certain aspects of the MER process were still under development at the time. Nonetheless, these activities provided insights into rover and operations team performance that could be expected during the mission on Mars, ultimately facilitating successful and sustained mission operations on Mars for over 3.5 years thus far.

This paper provides a brief overview of field operations activities conducted using prototype rovers prior to the MER landings. It also summarizes aspects of the performances achieved by *Spirit* and *Opportunity* during their surface missions on Mars with a focus on mobility/navigation and robotic arm functionality achieved by these flight rovers.

II. Prototype Rover Operations Testing on Earth

It is often difficult to find, or impractical test in, Earth environments with the same conditions of target planetary environments. However, analogue Earth terrains are accessible that allow meaningful functional evaluation of rover systems for planetary environments. Field operations using Mars rover prototypes were conducted in such analogue terrains in the years of 2001-2003.

A. FIDO Field Trials

In 2001 and 2002 *FIDO* (Fig. 1) was used to rehearse realistic simulations of rover operations planned for the MER mission [3, 4]. Each mission rehearsal involved over 60 science team participants from multiple institutions including NASA, U.S. Geological Survey, and a host of universities. This team constituted the Science Operations Working Group (SOWG) responsible for conducting the test from JPL. Each time, the *FIDO* system was used to physically simulate a 20-sol mission scenario generally representative of what the MER *Spirit* rover would be commanded to execute over a period of 10 Earth days. At the field sites, a small team of field geologists and rover engineers handled logistics and related activities.

The primary objective was for the SOWG to use a remote rover system and rover-mounted instruments to acquire data for formulating and testing hypotheses about the geologic evolution of the field site. Mission operations for those field trials were "blind" and fully remote. That is, the rover was commanded via satellite communications from JPL, and prior knowledge of the desert test sites was limited to large (tens of square kilometers) aerial images and spectral data typical of real Mars orbital observations.

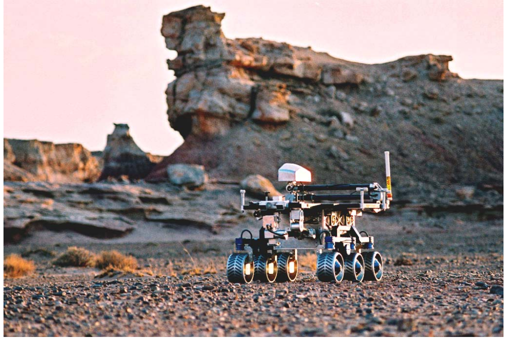


Fig. 1. *FIDO* rover during MER-FIDO field trial, Arizona USA, 2002.

For all intents and purposes, all operations were conducted as if the rover were on Mars, in compliance with MER flight rules (operational constraints on rover and ground data system use) and using many of the same types of rover planning and command functions to be employed during the actual MER flight mission. The Planetary Robotics Laboratory at JPL was used to emulate the MER mission operations area, and collaborative software tools for robotic science operations planning [5] were used as the Ground Data System (GDS). The GDS processed and distributed downlink imagery and data to those participating in the mission operations facilities at JPL. Satellite link capability was available via satellite modem connection between networked computers and a satellite dish antenna allowing remote commanding of the rover via the Internet. Fig. 2 roughly illustrates the communications arrangement for each *FIDO* field trial.

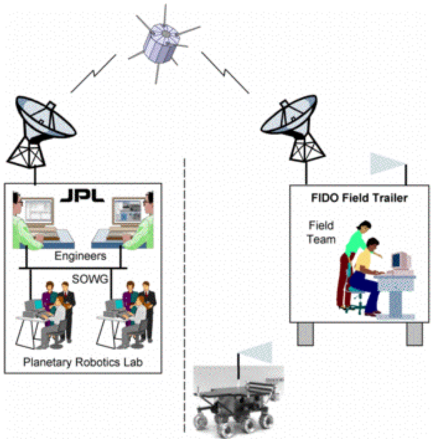


Fig. 2. Illustrated operations configuration for MER-FIDO field trials.

B. MER Engineering Model Field Trial

An additional field test was conducted to further support preparation for conducting semi-autonomous rover activities on Mars. It was conducted over a period of five days in the summer of 2003, between the launches of *Spirit* and *Opportunity*. A high-fidelity MER engineering model called the Surface System Test Bed (*SSTB*) (Fig. 3) was used. This rover is essentially identical in form, function, and capability to *Spirit* and *Opportunity*, with the exception of actual solar arrays and some electronics that operate off-board via a physical power, electronics, and communications tether.

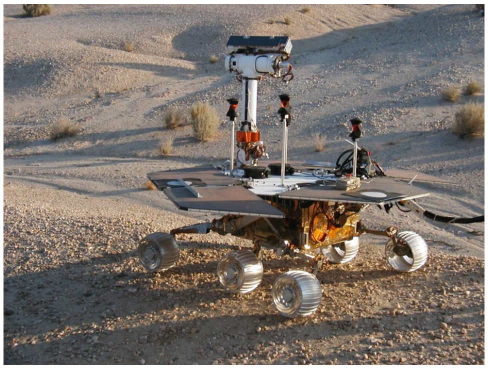


Fig. 3. MER Surface System Test Bed engineering model.

At the field site, a small team of rover engineers conducted surface navigation tests and handled logistics and related activities while several field geologists conducted soil properties experiments and measurements on soil excavated using one of the rover's wheels. A small subset of the MER mission operations team (command sequence developers, rover mobility engineering analysts, and ground data system personnel) participated remotely from JPL.

The main objectives for the MER *SSTB* field tests were to validate onboard navigation software functionality in realistic outdoor terrain and acquire outdoor imagery for related off-line use. An additional focus was verification and validation of the end-to-end command sequencing and uplink as well as the telemetry downlink and health assessment cycle of mission operations. Objectives included testing surface operations capabilities for navigation in undulated terrain under natural (Earth) lighting conditions, and in scenarios involving science target approaches and fine positioning at science targets for robotic arm use.

A portion of the actual MER mission support facility and GDS at JPL was used, including collaborative software tools for science operations planning as well as robotic motion planning and sequencing [5, 6]. The same satellite link capability used for prior *FIDO* trials was employed with an additional field trailer needed to support *SSTB* operations via tether between the rover and the in-trailer support equipment. Command and telemetry were transmitted wirelessly between the field trailers in this case as indicated in Fig. 4, which roughly illustrates the communications arrangement for the *SSTB* field trial.

Mission operations for the MER field trial were fully remote but not "blind." The MER GDS processed and distributed data received via satellite to those participating in the MER mission operations facilities at JPL. With the use of a MER engineering model as the test rover, this field test was most relevant for the actual flight rovers and the GDS software tools employed, all of which allowed rehearsal of actual tactical mission operations processes and procedures.

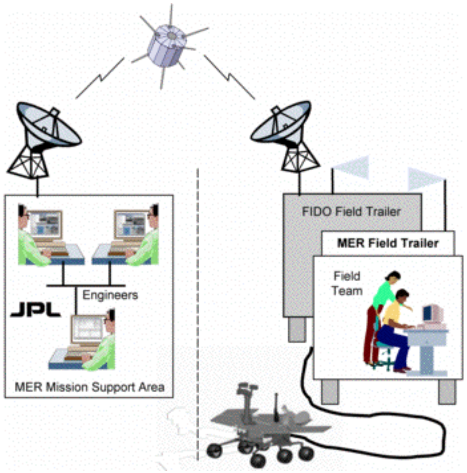


Fig. 4. Illustrated operations configuration for MER-SSTB field trial.

III. Summary of Earth-Based Field Results

The field venue for the 2001 trial was a small arroyo on the southern edge of the Soda Mountains in the Mojave Desert of California USA (almost 180 miles from JPL). Activities executed in the field included autonomous traversal to specified targets, approaching rock targets and deploying multiple instruments among other things [3]. *FIDO* traversed a total distance of 135 m over the desert terrain throughout the field trial. A number of short and long traverses were interspersed among many stationary science investigation activities. The longest continuous autonomous traverse was 40 m, and the average rover speed during traverses was 60 m/hr while negotiating and avoiding obstacles and terrain hazards [3].

The venue for the 2002 trial was an ancient flood plain in Gray Mountain, Arizona located approximately 40 miles north of Flagstaff, Arizona USA (almost 500 miles from JPL). Field operations consisted of similar science and robotic activity sequences exercised in 2001 with *FIDO* traversing a total distance of 202 m including short and long traverses interspersed among stationary science activities along the way. The longest continuous autonomous traverse was ~70 m, and the average rover speed during traverses was 60 m/hr [4]. A narrative of the 2002 field trial, daily rover operations, and scientific findings can be found on the Internet at http://marsrovers.jpl.nasa.gov/fido.

Field trials with *FIDO* helped develop the operations approach for MER by enabling simulation of MER processes for rover activity planning and sequencing relatively early in the project when flight hardware and certain GDS software tools were not yet available. The processes would eventually be exercised via Operations Readiness Tests, which could fully exercise the MER operations processes using high-fidelity proto-flight rovers and MER GDS software tools under realistic conditions.

The field venue for the MER *SSTB* field trial was a dry lakebed located in Edwards, California USA (almost 100 miles from JPL) where tests were performed over a period of five days. Two test sites were used; one was relatively flat with a sparse rock distribution and the other was a gulley with undulated terrain. Building upon results of prior indoor and outdoor mobility validation tests at JPL, navigation traverses were completed over shallow hills including short (~10 m) approaches toward vertical walls, sloped walls, and discontinuous terrain drop-offs. The first remotely planned, commanded, and analyzed traverses using part of the MER GDS and onboard navigation software were performed during this field trial. The testing served to validate stereo image processing software (used for hazard detection and avoidance in support of autonomous navigation) in nominal outdoor terrain. It was also valuable for acquiring imagery of different types of terrain than were available in earlier test environments and was useful for off-line testing of image processing and hazard detection software at a later date. *SSTB* is still used today to occasionally validate first-time or otherwise risky sequences before trying them on Mars.

IV. Intelligence and Autonomy in MER Operations

The experiences and results gained from Earth-based field tests generated lessons-learned for remote semi-autonomous rover operations. Many were directly followed in the conduct of actual MER mission operations and/or led to feature enhancements for GDS or rover onboard flight software. Comparable robotic activities and tasks were exercised during the Earth-based field trials and provided insights for expected performance during the mission on Mars. Some scenarios were encountered on Mars that were not thoroughly tested for during field trials but the training and expertise of the mission operations team along with the flexible rover system design enabled compensation and adaptability in all cases encountered.

As benefactors of experience gained during field testing on Earth, the *Spirit* and *Opportunity* rovers performed well throughout their respective 90-sol prime missions on Mars [7, 8] and several extended mission periods since then [9]. Both rovers have far out-lived their prime mission durations and continue to explore for over 3.5 years beyond their landing dates thus far. Fig. 5 shows photo-realistic MER models as insets in actual images during their extended missions. What has turned out to be a long duration mission may not have been possible without the Earth-based mission operations team (and longer than projected hardware operational lifetimes). In this case, humans supplied and gained science intelligence through remote use of meager robotic autonomy. For the MER mission, intelligence is largely human while autonomy is necessarily robotic (until human missions to Mars become possible).

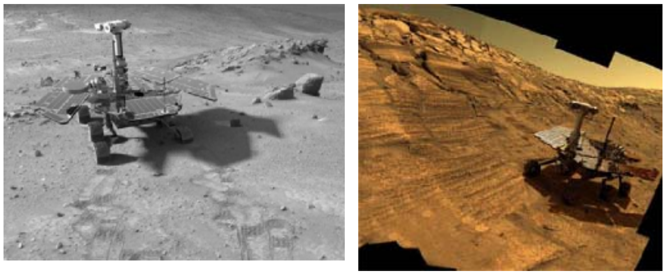


Fig 5. *Spirit* (left) in Columbia Hills and *Opportunity* (right) in Endurance Crater on Mars (Special-effects images created using photo-realistic rover models & image mosaics acquired during their missions. Rover model size approximated based on size of rover tracks in actual mosaic).

Within the MER surface mission operations system (Fig. 6) the SOWG, a rover engineering analysis and sequencing team, and the rovers form a closed-loop human-robot control system (notwithstanding the NASA's Deep Space Network and supporting teams and systems beyond the scope of this paper). Humans collaborate with the rovers to achieve best performance of onboard mobility and robotic arm software to maximize the acquisition and return of science data. Engineering analysts effectively function in the feedback loop of the human-robot system (Fig. 7) as human observers of rover state as well as maintainers of the best state knowledge for delivery to the uplink planning team. SOWG and engineering sequencing functions are manifested in the feed forward loop and can be thought of as providing reference inputs and serving as compensators for the rover system based on engineering state and recommendations from engineering analysts. Fig. 7 shows a simplified view of the operations process as a closed-loop control system; see [2] for a more definitive reference detailing the process.

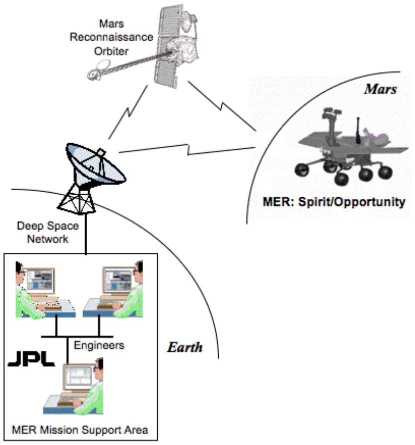


Fig. 6. Illustrated operations configuration for the MER mission.

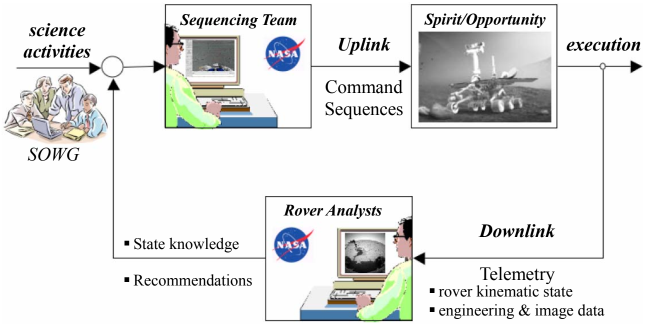


Fig. 7. Simplified human-robot system for MER remote surface operations.

Within this closed-loop human-robot system the science instrument, image, and engineering data telemetered to Earth drive the exploration plans for the next sol. The Earth-based planning process proceeds with generation of rover motion and science instrument command sequences that will carry out the intended activities. The engineering sequencing team refines motion commands using their perception of the rover surroundings and knowledge of rover behavior [10, 11]. This is facilitated by analyses that result in engineering recommendations for making the best use of the rover functionality including, for example, offline use of ground tools and human perception to localize the rovers and update their poses after accumulation of onboard odometry errors. This collaborative loop of human intelligence and rover autonomy serves to facilitate proper execution of the sol's command load on Mars. Nominally, each rover is sent a command load daily and autonomously executes uplinked sequences throughout a period of 3-6 hours around local noon (with occasional nighttime communications or immobile science activity). In this manner, human-guided robotic execution leads to exploration progress, which generates new data and images that feedback into the cyclic process, ultimately leading to scientific discovery.

V. Summary of Performance on Mars

Selected aspects of *Spirit* and *Opportunity* mobility, navigation, and robotic arm performances are summarized below. Additional engineering details about their missions and relevant operational performance metrics can be found in [10-12] and [9], respectively. The performance summarized here is based on operations through mid-September 2006, after which a new software upgrade was uplinked to the rovers including substantial functional enhancements expected to yield performance improvements. To date, the new functional enhancements have not been used frequently enough to generate new performance data of sufficient quantity to report here.

A. Total Traverse Distance and Navigation Speed

September 15, 2006 was sol 960 for *Spirit* and sol 940 for *Opportunity*. By that date, the total traverse distance for *Spirit* was 6876 m, of which 3126 m (45%) was traversed autonomously. Also by then, *Opportunity* traversed 9130 m total, of which 2465 m (27%) was traversed autonomously.

Rover average autonomous traverse rates depend on terrain traversability. A given rover may traverse flat and hazard-free terrain at a faster average rate than it would a sloped and rocky terrain due to the increased deliberation required to assure safe traversal in the latter case. While directed/blind traverses can achieve speeds over 100 m/hr, autonomous traverses employ onboard image processing to detect and avoid geometric hazards and achieve, as a result, driving speeds from 10 m/hr in obstacle-laden terrain up to 36 m/hr in safe terrain [13]. Traverses that employ visual odometry execute more slowly due to the more frequent image processing required to provide the best position estimates, and due to short mobility motions needed to ensure close spacing between consecutive images used for visual odometry. These image acquisition and motion constraints limit the visual odometry traverse rate to about 10 m/hr. Other factors contributing to the relatively slow execution times of MER autonomous mobility are the low computation speed of the 20 MHz RAD6000 and the fact that dozens of tasks in the real-time system share a single address space and cache [13].

Spirit's average and maximum autonomous traverse rates at its Mars landing site (Gusev crater, ~ 7% rock abundance [14]) thus far are 15.06 m/hr and 34.35 m/hr, respectively. The same traverse rates for *Opportunity* at its landing site (Meridiani Planum) thus far are 22.09 m/hr and 36.0 m/hr, respectively. The Meridiani Planum site is largely devoid of rocks, and observations suggest a rock abundance of only a few percent [14]; we use 3% here. In each case, the average traverse rates are taken over all traverse sols (up to *Spirit*'s sol 960 and *Opportunity*'s sol 940) that included autonomous navigation. Figs. 8 and 9 show histogram distributions of autonomous navigation traverse rates for both rovers over the course of their missions through mid-September 2006.

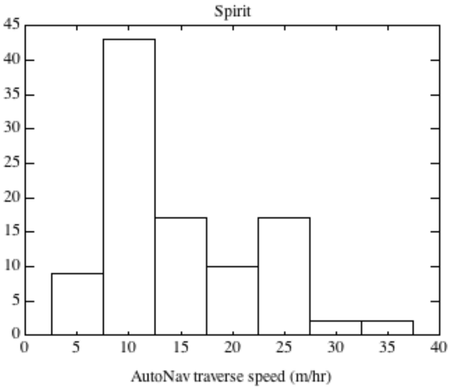


Fig. 8. Navigation speeds during 100 autonomous traverses for *Spirit*.

The histograms show that most of *Spirit*'s autonomous traverses were executed at relatively slower rates than most of *Opportunity*'s. Both rovers executed relatively few autonomous traverses at rates greater than about 30 m/hr. *Spirit*'s autonomous traverses were generally slower across the more challenging terrain at Gusev crater, which presented more geometric hazards for the rover to avoid. Thus far, *Opportunity* has executed two-thirds fewer traverses in autonomous mode than *Spirit*. This too is indicative of the lesser challenging terrain at the Meridiani Planum site, which was relatively flat and open, and thus obviated the need for autonomous avoidance of non-traversable hazards during many traverses. Among the most challenging terrains encountered, relative to those involved in Earth testing, were surfaces of substantial slope (over 50% of the rovers' 45° tilt stability limit), surfaces of softness sufficient to induce high wheel slip, and surfaces soft enough to yield to the rover's ground pressure, embed the wheels and resist motion.

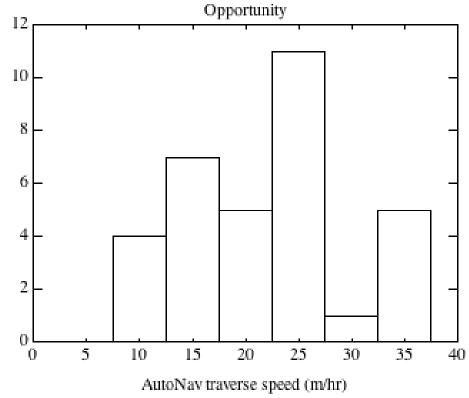


Fig. 9. Navigation speeds during 33 autonomous traverses for *Opportunity*.

B. Science Target Approach

In addition to traversing from place to place, science rovers must deploy instruments in contact with or in proximity to reachable rocks and soil specimens. An instrument positioning system capable of precision placement of *in situ* instruments from mobile platforms is essential [15]. *Spirit* and *Opportunity* perform this function using a 5 degree-of-freedom (DOF) robotic arm known as the Instrument Deployment Device (IDD). It is mounted in a frontal area beneath the rovers' solar panel. Its end-effector is a rotary turret to which science instruments are mounted, and the remaining 4 DOFs are used to place the instruments onto science targets within the arm's kinematic work volume ~ 0.14 m^3). The instruments mounted on the IDD end-effector turret include a microscopic imager (MI) to capture close-up images, a Moessbauer spectrometer (MB) to detect iron-bearing minerals, an Alpha-Particle-X-Ray Spectrometer (APXS) to determine the elemental chemistry of surface materials, and a Rock Abrasion Tool (RAT) for exposing fresh material beneath weathered rock surface layers via controlled-force loading and physical abrasive action. The arm is also used to position the spectrometers for placement onto an instrument calibration target and science-related magnets mounted at different locations on the rover body.

Rover mobility is used to approach a position offset from a science target such that the target is within the work volume of the IDD. A successful target approach is typically followed by placement of instruments onto the target using the IDD. Controlled placement of instruments in contact with science targets is facilitated by contact sensor feedback. Redundant sets of contact sensors on each instrument provide tactile feedback used by software to halt arm motion upon expected or unexpected contact. IDD joint and contact sensor telemetry facilitates determination of errors between commanded and actual placement positions.

An approach-traverse is on the order of 10 m and intended to terminate with a specific science target within the IDD work volume. The science target is selected and designated by mission operators in stereo imagery acquired prior to the approach. Each rover was required to be capable of approaching a science target in a single command cycle whenever within 2 m of the target at the start of a sol. Depending on the initial approach distance from a target, complexity of terrain between rover and target, and other considerations, target-approaches do not always succeed on first attempts. On occasion, more than 1 sol is needed to reach certain targets, particularly when approach distances are longer than 2 m, and since images of the IDD work volume at the rover's final location are required (until recent software upgrades) before IDD deployment is permitted.

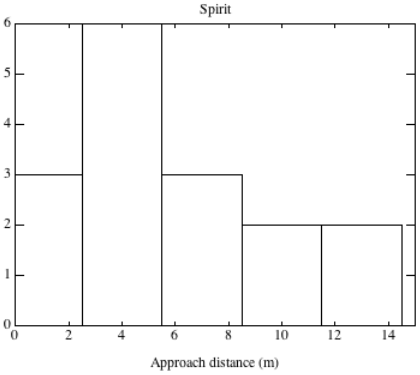


Fig. 10. Frequencies of science target approach distances executed by *Spirit* among a sample set of 16 approach-traverses.

Figs. 10 and 11 show histograms of the science target approach distances for both rovers. Most approach distances have been less than 6 m and several approach-traverses over 10 m have been executed by each rover thus far. Out of 16 target approaches considered for *Spirit*, 14 (88%) were successfully executed in a single sol. Out of 18 considered for *Opportunity*, 13 (72%) were successfully executed in a single sol. All other target approaches were completed within 2 sols. The longest single-sol target approaches among those considered here were 15 m and 11.8 m, respectively, for *Spirit* (sol 685) and *Opportunity* (sol 803).

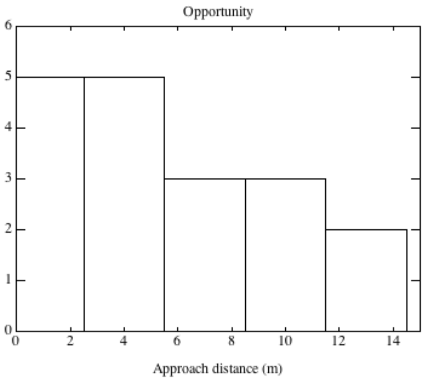


Fig. 11. Frequencies of science target approach distances executed by *Opportunity* among a sample set of 18 approach-traverses.

C. Robotic Arm Instrument Placement

Instrument placement is achieved autonomously (including switching from one instrument to another) by realizing a combination of kinematic configurations that are pre-taught and/or newly commanded. Cartesian and joint-space motions are determined via onboard calculation of inverse kinematics and position error compensation. Original MER operational guidelines required human confirmation of the rover position prior to each IDD use, making each approach and instrument placement take at least two sols, but software upgrades mentioned earlier will make approach and placement possible in the same sol. The requirements on placement performance hold in either case.

While human operator assisted techniques have been employed to achieve position accuracies as good as 0.8 mm as well as improvements in repeatability performance [16], we consider average performance over a large sampling of instrument placements executed without operator assistance (beyond designation of placement targets). Figs. 12 and 13 shows sample histories of instrument placement errors for each rover's IDD and individual arm-mounted instrument. These data are associated with instrument placement activities during Mars surface operations. Across all instruments, the data represent over 1200 placements through sol 944 for *Spirit* and nearly 1200 placements through sol 893 for *Opportunity* on rock, soil, and rover-mounted targets. Placement errors are derived from IDD and instrument contact switch telemetry as well as front stereo Hazard Camera range data evaluations. Instrument placement performance, considering errors in placement of all instruments, reveals average absolute positioning accuracies of 6.81 mm for *Spirit* and 5.84 mm for *Opportunity*.

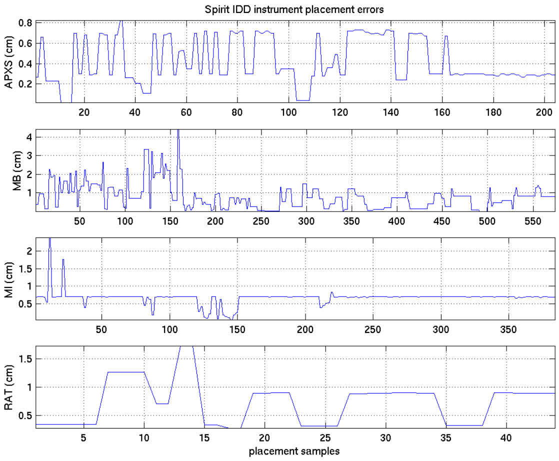


Fig. 12. Robotic arm positioning errors for *Spirit* when placing the APXS, MB, MI, and RAT instruments onto science targets. Data covers instrument placements through sol 944.

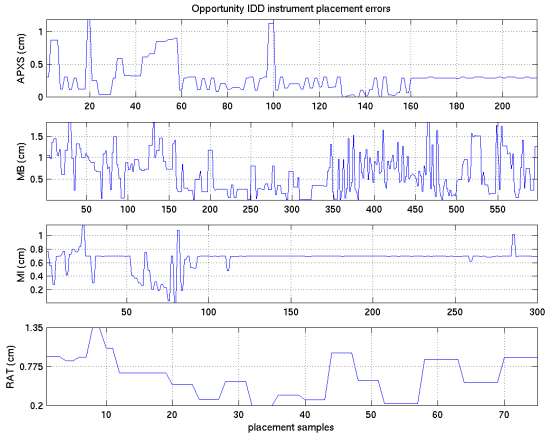


Fig. 13. Robotic arm positioning errors for *Opportunity* when placing the APXS, MB, MI, and RAT instruments onto science targets. Data covers instrument placements through sol 893.

VI. Conclusion

As the space community embarks on future robotic missions to surfaces of other planets or the moon, Earth-based robotic prototypes will become more important for proving and rehearsing surface operations approaches. This paper gives a brief overview of field trials leading up to the MER surface mission that employed two prototypes, *FIDO* and the MER *SSTB*. These field operations tests provided venues for rehearsing, validating, and refining aspects of the mission operations process, tools, and approaches used on MER. They also provide opportunities to train mission personnel on how to use autonomous rovers to conduct remote field-based science and identify technologies that require additional development and/or evaluation.

Mobility and robotic arm performance achieved by *Spirit* and *Opportunity* on Mars cannot be directly compared to the

Earth-based field trial performances since, at various stages of the latter, different hardware, software, and environments were used. The Earth-based tests allowed operators to refine approaches to operating the systems during the actual mission. On Mars the mobility and navigation software kept both rovers safe from terrain hazards while making progress toward commanded goals. Both rover robotic arms enabled acquisition of science data leading to conclusive evidence that water once existed on the surface at their respective landing sites [7, 8]. The performance data summarized herein is based on operations through mid-September 2006. Since then, new autonomy software enhancements expected to yield performance improvements were uplinked but not sufficiently exercised to yield reportable results yet. They include a global path planner, visual target tracking to further automate approach-traverses, and autonomous deployment of the IDD after an approach-traverse, without the additional command cycle that was previously required.

The creativity and expertise of human mission operators has kept the rovers operating despite slowly degrading or aging hardware components. Robotic execution errors reported by mobility or navigation and robotic arm software were often due, to some extent, to some form of human error; that is, command sequence/sequencing errors, unaccounted for system operational behaviors not experienced during Earth testing, or isolated shortfalls of planning tools or processes. All of these were later rectified as lessons-learned for the MER mission, still ongoing at the time of this writing.

While all mission scenarios cannot be anticipated and tested for, robust and fault tolerant system design coupled with safe mission operations can ensure successful surface missions. Thus far, all novel situations encountered during the MER surface mission, that were not thoroughly tested for during field trials, have been successfully overcome thanks to the training and expertise of the mission operations team along with the flexibility of rover system design.

ACKNOWLEDGMENT

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Research and field activities were supported by the Mars Exploration Program and Mars Technology Program, Science Mission Directorate.

Outstanding contributions of the FIDO rover engineering team, MER Mobility/Navigation Working Group and MER field scientists Robert Anderson and Albert Haldemann are acknowledged. Collaboration and support provided by the MER Project and personnel is gratefully acknowledged, with special thanks to John Callas, Jennifer Trosper, and Jacob Matijevic of JPL and MER Principal Investigators, Steve Squyres of Cornell University and Ray Arvidson of Washington University, St. Louis. Expert Internet and satellite networking assistance from Claudia De Luna, Kathleen Walsh, William Adler of JPL and Thomas Nielson and Andre Finlinson of Lyman Bros., Inc. is acknowledged. Major credit goes to the MER Mission Operations team with special thanks to the Mobility/ IDD Analysts for performance assessments reported herein.

REFERENCES

[1] E. Tunstel, “Autonomous mobility software validation challenges for planetary surface missions,” *Intl. Conf. on Space Mission Challenges for Information Technology*, Pasadena, CA, July, 2003, pp. 167-173.

[2] A. H. Mishkin, D. Limonadi, S. L. Laubach and D. S. Bass, “Working the Martian night shift: The MER surface operations process,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, 2006, pp. 46-53.

[3] E. Tunstel, T. Huntsberger, H. Aghazarian, et al, “FIDO rover field trials as rehearsal for the 2003 Mars Exploration Rover mission," *9th Intl. Symp. on Robotics & Applications*, *WAC*, Orlando, FL, 2002.

[4] E. Tunstel, T. Huntsberger, and E. Baumgartner, “Earth-based rover field testing for exploration missions on Mars," *10th Intl. Symp. on Robotics & Applications*, *WAC*, Seville, Spain, 2004, pp. 307-312.

[5] P. G. Backes, J. S. Norris, M. W. Powell, et al, “The Science Activity Planner for the Mars Exploration Rover mission: FIDO Field Test Results,” *Proc. IEEE Aerospace Conf.*, Big Sky, MT, March 2003.

[6] S. Maxwell, B. Cooper, F. Hartman, J. Wright and J. Yen, “The design and architecture of the Rover Sequencing and Visualization Program (RSVP),” *8th Intl. Conf. on Space Operations*, Montreal, Canada, 2004.

[7] *Science*, Special Issue: Spirit at Gusev Crater, vol. 305, 5685, AAAS, Aug. 2004, pp. 793-845.

[8] *Science*, Special Issue: Opportunity at Meridiani Planum, vol. 306, 5702, AAAS, Dec. 2004, pp. 1697-1756.

[9] E. Tunstel, “Performance metrics for operational Mars rovers,” in R. Madhavan and E. Messina (Eds.), *2006 Performance Metrics for Intelligent Systems Workshop* (PerMIS'06), NIST Special Publication 1062, Gaithersburg, MD, USA, August 2006, pp. 69-76.

[10] C. Leger, A. Trebi-Ollennu, J. R. Wright, S. A. Maxwell, et al, “Mars Exploration Rover surface operations: Driving Spirit at Gusev Crater,” *Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics*, Waikoloa HI, pp. 1815-1822.

[11] J. Biesiadecki, E. Baumgartner, R. Bonitz, et al, “Mars Exploration Rover surface operations: Driving Opportunity at Meridiani Planum,” *Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics*, Waikoloa HI, pp. 1823–1830.

[12] *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, 2006, Volume dedicated to Mars Exploration Rovers, A. Trebi-Ollennu (Guest Ed.).

[13] M. Maimone, J. Biesiadecki, E. Tunstel, Y. Cheng and C. Leger, “Surface navigation and mobility intelligence on the Mars Exploration Rovers,” in A. Howard and E. Tunstel (Eds.), *Intelligence for Space Robotics*, TSI Press, San Antonio, TX, pp. 45-69.

[14] M. P. Golombek, R. E. Arvidson, J. F. Bell III, et al, “Assessment of Mars Exploration Rover landing site predictions,” *Proc. 36th Lunar and Planetary Science*,” League City, TX, March 2005, Paper #1542.

[15] A. Trebi-Ollennu, E. T. Baumgartner, C. Leger and R. G. Bonitz, “Robotic arm in-situ operations for the Mars Exploration Rovers surface mission,” *Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics*, Waikoloa, HI, 2005, pp. 1799-1806.

[16] E. T. Baumgartner, R. G. Bonitz, J. P. Melko, et al, “The Mars Exploration Rover instrument positioning system,” *Proc. IEEE Aerospace Conference*, Big Sky, MT, March 2005, pp. 1-19.

Planning to Fail - Reliability as a Design Parameter for Planetary Rover Missions

S. Stancliff, J. Dolan
The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA, USA
{sbs,jmd}@cs.cmu.edu

A. Trebi-Ollennu
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA, USA
Ashitey.Trebi-Ollennu@jpl.nasa.gov

***Abstract*—**The Mars Exploration Rovers (MER) have been operating on Mars for more than three years. The extremely high reliability demonstrated by these rovers is a great success story in robotic design. This reliability comes at a high cost, however, both in the initial cost of developing the rovers and in the ongoing operational costs for their mission extensions. If it were possible to design rovers with reliability more in line with their mission requirements (in the case of MER, 90 days), considerable cost reductions could be achieved. This will be even more important for future planetary robotic missions due to greatly increased mission durations.

In this paper we present an overview of our ongoing research in the area of predicting robot mission reliability, and we show how a mission designer can trade off reliability against costs in order to find an optimal reliability target for a given robotic mission. Our results show that for a given mission there is an optimal reliability range with respect to cost and that having rovers with reliability that is too low or too high is suboptimal from an economic standpoint. This suggests that a better cost-reliability tradeoff can be obtained by "planning to fail" by designing rovers which have lower reliability than current legacy designs.

***Keywords*:** *planetary rovers, mission design, mission cost, reliability, failure, risk.*

I. Introduction

In the near future, NASA intends to send rovers to Mars for missions lasting an order of magnitude longer than the intended duration of the Mars Exploration Rovers (MER) mission. If these future rovers follow legacy designs, then increasing the mission duration by an order of magnitude will require that the rovers be built using components with failure rates an order of magnitude lower. Since NASA rovers already make use of some of the most reliable components available, it is doubtful whether components with order of magnitude lower failure rates are available, let alone affordable.

In order to increase rover mission durations without incurring exponential increases in rover costs, it is necessary to consider risk not simply as something to be minimized to the greatest extent possible, but instead as a quantitative design factor to be traded off against other design factors in order to seek an optimal mission configuration.

In the mobile robotics literature there is little formal discussion of reliability and failure. When reliability is mentioned, it is usually qualitatively, and in passing. Reference [1], for example, mentions intermittent hardware failures as an explanation for gaps in experimental data but makes no attempt at characterizing the failures.

A handful of prior papers make use of reliability engineering for analysis of mobile robot failure rates. Reference [2] provides an overview of robot failure rates at the system level (i.e., robot model X failed Y times in Z hours of operation) and also breaks down failures according to the subsystem that failed (actuators, control system, power, and communications). Reference [3] extends the work in [2] both by the inclusion of additional failure data of the same type and also by addition of new categories of failure—those due to human error. Reference [4] provides a detailed analysis of failures experienced by some of the robots used in searching the World Trade Center wreckage in 2001. Reference [5] provides failure data for robots used in long-term experiments as museum guides.

What these papers have in common is that they use reliability engineering tools in the assessment of existing robots. Our work differs in that it addresses how to use reliability engineering tools for designing robots and robotic missions.

In earlier work we have developed methods for using reliability engineering tools to predict the probability of a robot failing during a mission [6], and we have used these tools to compare the performance of different robot and robot team configurations [7]. The only known work preceding ours in the area of predicting mobile robot team reliability is [8]. That paper's methods are similar to ours in that they are based in the reliability literature, but that work has a narrow focus on teams of robots with cannibalistic repair capability. In contrast, we are developing a general methodology that can be applied to a wide variety of robot teams and missions.

The work presented in this paper differs from our earlier work by addressing the relationship between robot reliability and overall mission cost, and demonstrating how this relationship can be used to identify an optimal reliability level which minimizes mission cost.

II. Example mission scenario

A. Missions and Tasks

Consider a planetary exploration mission where a team of rovers is tasked to install a solar panel array for a measurement and observation outpost. The mission consists of carrying 50 solar panels from the landing site to the outpost and then assembling them. The size of the solar panels is such that two rovers are needed to carry and assemble one panel.

For the purposes of the reliability analysis, the task of assembling a solar panel is broken down into three subtasks:

- Transit to the outpost,
- Assemble the panel, and
- Return to the landing site.

We assume that failure occurs only at the end of a subtask. This allows us to avoid dealing with partially completed subtasks. This simplification does not limit the resolution of the representation because tasks can be restated into smaller subtasks if needed.

B. Rovers and Components

For this analysis we assume that the rovers on the team are identical. The rovers are considered to be made up of several subsystems that are independent from the standpoint of reliability. The specific partitioning is not important to the methodology, but for the analyses in this paper the rovers are divided into the subsystems listed in Table 1.

The subsystem reliabilities listed in Table 1 were calculated from the failure rates of the major components in each subsystem. An example component breakdown for the power module is shown in Table 2. Due to the limited amount of failure data available for planetary rovers, the failure rates in Table 2 were derived from the RAC databooks ([9]) which are commonly used for reliability prediction in aerospace and military applications. Additional details on the calculation of subsystem failure and the combining of component failure rates can be found in [10].

We assume that the failure of any single subsystem leads to failure of the entire rover. For the current example mission this is a reasonable assumption, since all of the subsystems must be functioning in order to complete the mission subtasks.

The probability of a subsystem failing during a task is found using standard reliability engineering methods assuming a constant failure rate. Two inputs determine the module failure probability: the module failure rate and the length of time for which the module is operated during the task. The durations shown in Table 3 were assigned using reasonable assumptions about the relative durations of different tasks and the relative usage of different modules. During the transit task, the panels are assumed to be locked in a fixed position not requiring manipulator actuation.

TABLE 1
ROVER SUBSYSTEMS AND RELIABILITIES

Subsystem	MTTF (h)
Power	4202
Computation&Sensing	4769
Mobility	19724
Communications	11876
Manipulator	13793

TABLE 2
COMPONENTS COMPRISING POWER SUBSYSTEM

Component	Quantity	Failure Rate (1/h)
Battery	2	2.10x10^{-7}
Battery control board	2	4.00x10^{-7}
Mission clock	1	1.00x10^{-7}
Power distribution unit	1	1.70x10^{-6}
Power control unit	1	1.90x10^{-7}
Shunt limiter	1	1.14x10^{-5}
Electrical heater	2	3.00x10^{-6}
Radioisotope heater	2	1.36x10^{-5}
Thermal switch	2	9.50x10^{-5}

The probability of survival for a subsystem for a given task is given by the equation

$$P = e^{-t\lambda}, \qquad (1)$$

where t is the amount of time that the subsystem is used during the task and λ is the failure rate for the subsystem.

Using (1) and the data from Tables 1 and 3, we calculated the probability that each subsystem will survive each task. These probabilities are shown in Table 4.

TABLE 3
SUBSYSTEM USAGE BY TASK IN HOURS

Subsystem	Transit	Assemble	Return
Power	6	8	6
Computation&Sensing	6	4	6
Mobility	6	8	6
Communications	2	4	2
Manipulator	0	8	0

III. Approach

The experiments in this paper make use of the method described in [5] for predicting probability of mission completion. In this method, the mission is represented using a state machine that is simulated stochastically.

The simulation is repeated many times, with the average score of all trials giving the overall probability of mission completion. The results of the simulations were verified by hand calculation for a few simple cases.

TABLE 4
SUBSYSTEM PROBABILITY OF SURVIVAL BY TASK

Subsystem	Transit	Assemble	Return
Power	99.86%	99.81%	99.86%
Computation&Sensing	99.87%	99.92%	99.87%
Mobility	99.97%	99.96%	99.97%
Communications	99.98%	99.97%	99.98%
Manipulator	100%	99.94%	100%

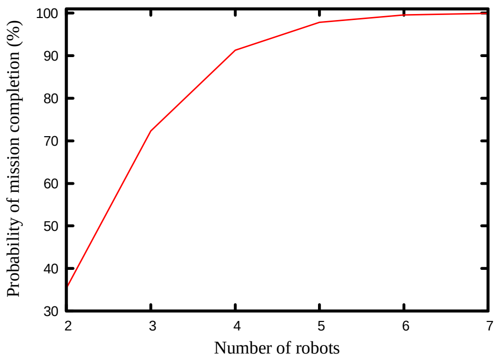


Fig. 1. Mission Reliability as a Function of Team Size

A. Relationship Between Team Size and Mission Success

Using this method, we first examine the relationship between the number of rovers on the team and the probability of completing the mission. Figure 1 compares teams of two to six rovers, each composed of the baseline components with the subsystem reliabilities shown in Table 1. This analysis can be used to determine a minimal team size for a required probability of mission success. For instance, if we set the required probability of mission success at 99.5% then Figure 1 shows that the team must consist of at least six rovers.

B. Relationship Between Component Reliability and Mission Success

Figure 1 shows that a six-rover team exceeds the mission reliability requirement. In such a case, a mission designer may wish to choose lower-reliability components in order to decrease mission costs. The same simulations used to create Figure 1 can be used to determine the minimum component reliabilities required to meet a particular mission reliability requirement. Figure 2 compares six-rover teams using components with reliabilities which vary from 60% to 100% of the values in Table 1. From this we find that we can achieve the 99.5% goal by using components with 95% of the reliabilities shown in Table 1.

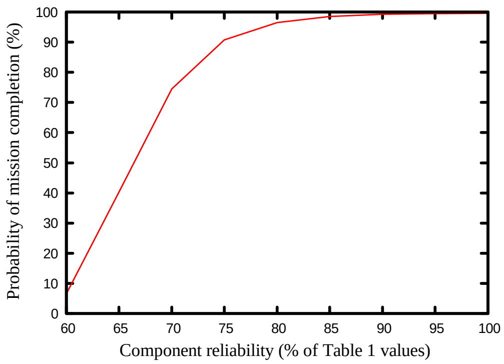


Fig. 2. Mission Reliability as a Function of Component Reliability

We have now determined that the smallest team with the lowest-reliability components which can achieve the design goal of 99.5% probability of mission success is a six rover team with the component reliabilities shown in Table 5. We use this team as the baseline for the comparisons that follow.

C. Relationship Between Component Reliability and Cost

The reliability of the rovers is related to the overall mission cost in two ways. First, there is the increased cost associated with higher-reliability rovers. Second, there is the increased expected value of the mission when using higher-reliability rovers due to a higher probability of mission success.

1) Cost of Reliability: In choosing components from which to build rovers, a designer would usually make choices among a small number of alternative components, each providing a certain reliability for a certain cost. However, in the early stages of design the mission designer may not have complete information about available components. In this case, it is useful to have a parametric model of the cost–reliability relationship. Reference [11] provides a general model for this relationship, which is given as

$$c = \exp\left\{ (1-f)\cdot\frac{(R_i - R_{\min})}{(R_{\max} - R_i)} \right\} \quad (2)$$

where R_i is a reliability of interest between R_{min} and R_{max} ; c is the relative cost of R_i compared to R_{min} ; f is the feasibility of reliability improvement (a number between 0 and 1); and c is the resultant relative cost of $\mathrm{R_i}$ with respect to $\mathrm{R_{min}}$.

This equation can be used to calculate the relative cost of the components used by the six-rover teams with differing component reliabilities. These costs are plotted in Figure 3 as a percentage of the baseline team cost, using $\mathrm{R_{min}}$=0, $\mathrm{R_{max}}$=1 and f=0.95. We examine the effect of changing the feasibility constant later in this paper.

Launch costs are also affected by rover reliability. More-reliable rovers will weigh more, due to increased size of more-reliable components and due to increased component redundancy. We have not found a model for the reliability–weight relationship in the literature. As an initial approximation for launch costs we assume that the relationship between weight and reliability is directly linear and that the relationship between launch costs and weight is also directly linear.

TABLE 5
COMPONENT RELIABILITIES GIVING 99.5% PROBABILITY OF SUCCESS FOR SIX-ROBOT TEAM

Subsystem	**MTTF (h)**
Power	3992
Computation&Sensing	4531
Mobility	18738
Communications	11282
Manipulator	13103

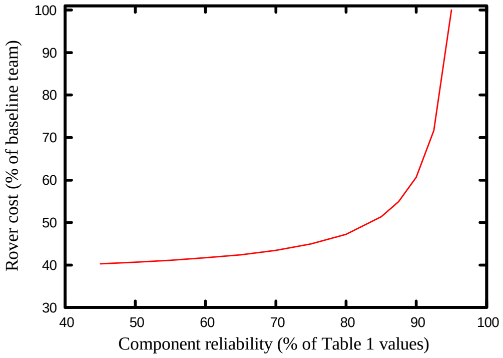


Fig. 3. Relative Cost of Rovers as a Function of Component Reliability

2) Expected Value of Mission: Any mission must have some inherent value to it. For some missions there will be an obvious economic or strategic value to which a dollar amount can be assigned. For a mission that lacks such an obvious dollar value, the cost of the baseline mission itself can be used as a lower bound for this inherent mission value, since the sponsor presumably expects some return on the investment.

Multiplying the probability of mission success by the inherent value of the mission gives an expected value for a given team configuration. For example, the relationship between component reliability and expected mission value is given by Figure 2, with the vertical axis relabeled as "expected value as percent of inherent value".

D. Overall Mission Cost–Reliability Relationship

Taking the expected mission value calculated above and subtracting the rover development and launch costs gives us an estimate of the net expected gain for the mission. We ignore operating costs here since we expect them to be roughly constant with respect to rover reliability (probably slightly higher for lower-reliability rovers due to the increased need for intervention).

In order to combine these costs meaningfully, we assign real dollar values to the various costs for the baseline team. These values are estimated from the costs of the MER mission, along with the assumption that the rovers for this mission would be somewhat cheaper and smaller than the MER rovers due to advances in technology and also because they are single-purpose machines. The values we assigned for the baseline team are shown in Table 6. Figure 4 then plots these component costs and values as well as the net expected gain as a function of rover component reliability.

IV. Conclusions

The most significant thing revealed by Figure 4 is that there is clearly an optimal reliability range with respect to the expected gain of the mission, and that this optimal reliability is significantly lower than the reliability of the baseline (legacy) design.

TABLE 6
BASELINE TEAM COSTS AND REWARDS

Item	Cost ($ Millions)
Robot cost (entire team)	150
Launch cost (entire team)	300
Inherent value of mission	450

The shape of the expected gain curve shows that for low-reliability rovers the cost of failure drives the expected gain value down, while for very high-reliability rovers the high cost of the rovers themselves drives the expected gain down. The optimal reliability range therefore lies in a medium-reliability region where neither of these costs is as high.

In order to evaluate the effects of some of our assumptions on these conclusions, we have repeated the above analysis for different values of the feasibility constant (since this value was arbitrary) and of the mission inherent value (since we used a lower-bound estimate for this value). These results are shown in Figures 5 and 6. These figures show that while the shape of the expected gain curve changes somewhat with these parameters, the overall trends remain the same, and both figures support the argument that the optimal range for mission reliability is at a lower level than we would intuitively consider to be the case.

While we expect that these curves will vary for different missions, we expect that the general trends will hold, indicating that it can be economically wiser to "plan to fail" by building rovers which have lower reliability than current legacy designs.

Acknowledgement

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and funded through the Director's Research and Development Fund Program.

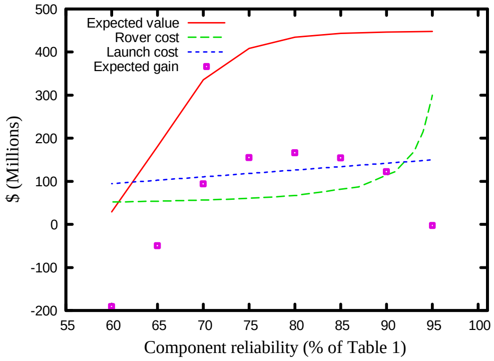


Fig. 4. Net Expected Gain with f=0.95, value = $450M

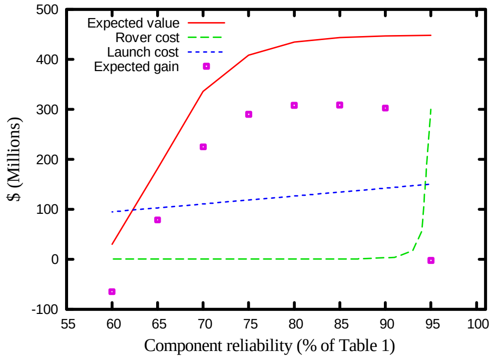


Fig. 5. Net Expected Gain with f=0.5, value=$450M

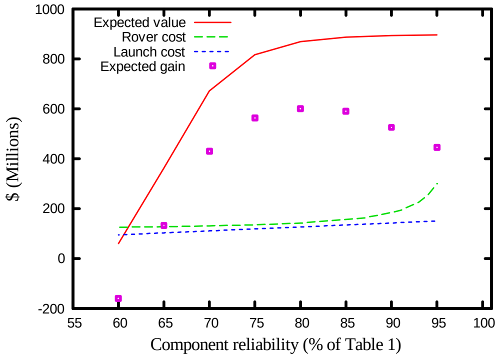


Fig. 6. Net Expected Gain with f=0.95, value= $900M

References

[1] R. Gockley, J. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. Schultz and J. Wang, "Designing robots for long-term social interaction," *Proc. 2005 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2005)*, pp. 1338–1343.

[2] J. Carlson and R. Murphy, "Reliability analysis of mobile robots," *Proc. 2003 IEEE Int. Conf. Robotics and Automation (ICRA 2003)*, pp. 274–281.

[3] J. Carlson, R. Murphy and A. Nelson, "Follow-up analysis of mobile robot failures," *Proc. 2004 IEEE Int. Conf. Robotics and Automation (ICRA 2004)*, pp. 4987–4994.

[4] M. Micire, "Analysis of the robotic-assisted search and rescue response to the World Trade Center disaster," M.S. thesis, University of South Florida, 2002.

[5] I. Nourbakhsh, C. Kunz and T. Willeke, "The mobot museum robot installations: a five year experiment," *Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2003)*, pp. 3636–3641.

[6] S. Stancliff, J. Dolan, and A. Trebi-Ollennu, "Mission reliability estimation for repairable robot teams," *Int. J. Advanced Robotic Systems*, Vol. 3, No. 2, Jun., 2006, pp. 155–164.

[7] S. Stancliff, J. Dolan, and A. Trebi-Ollennu, "Mission reliability estimation for multirobot team design," *Proc. 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2006)*, pp. 2206–2211.

[8] C. Bererton and P. Khosla, "An analysis of cooperative repair Capabilities in a team of robots," *Proc. 2002 IEEE Int. Conf. Robotics and Automation (ICRA 2002)*, pp. 476–486.

[9] RAC Automated Databook, Version 2.20, IIT Research Institute, 1999.

[10] S. Stancliff, J. Dolan, and A. Trebi-Ollennu, "Towards a predictive model of mobile robot reliability," Carnegie Mellon University technical report CMU-RI-TR-05-38, 2005.

[11] A. Mettas, "Reliability allocation and optimization for complex systems," *Proc. IEEE Annual Reliability and Maintainability Symp.*, 2000, pp. 216–221.

A Decision Space Compression Approach for Model Based Parallel Computing Processes

Robert J. Bonneau
George Ramseyer
AFRL /IFTC
Rome, NY 13441. U.S.A.

Abstract— Currently there are many DOD applications where warfighters are asked to make critical decisions based on environmental conditions that are highly complex and where there is incomplete knowledge of the local conditions. An example of such a situation is that of the theater commander who must deploy his C2/ISR assets such as communications and sensing platforms without complete knowledge of the local electromagnetic environment and its effect on his ability to maintain good information exchange and reconnaissance data for his forces. This type of situation falls into a broad class of problems where decision theory and complex physical models must interact for optimal performance such as investment analysis, weather prediction, and organizational dynamics. Such problems have been cast in the mathematical framework of "partial observability" where only some components of the environment are known. We thus we need to model the uncertainty of the environment and weigh our actions accordingly. The approach conventionally used for such optimization is a Partially Observable Markov Decision Process (POMPD) where we can model both our situational knowns and unknowns and come up with the best actions to take based on our model of what we know and do not know. We propose to develop a distributed computational framework that manages the complexity of such a process for large system optimization and provide an approach to parallelize and maintain operation for the system as more information and updates to our underlying environmental models change.

Keywords: *Partially observable Markov Decision Process, Resource Allocation*

I. INTRODUCTION

The problem of parallelization and complexity reduction in POMDPs is a relatively current topic given that the conventional approach is to assume a static model for our physical world and then compute our optimal decision policy based. Once the decision policy is computed, the computational work is over until the underlying system variables are changed and the process starts again. The problem with this approach is that there are many different policies of interest for a particular environment and the timescale are short for our underlying model assumptions. Additionally, most polices that involve up to 1000 environmental observations can take many hours if not days to compute on a single CPU. The ability to have perfectly

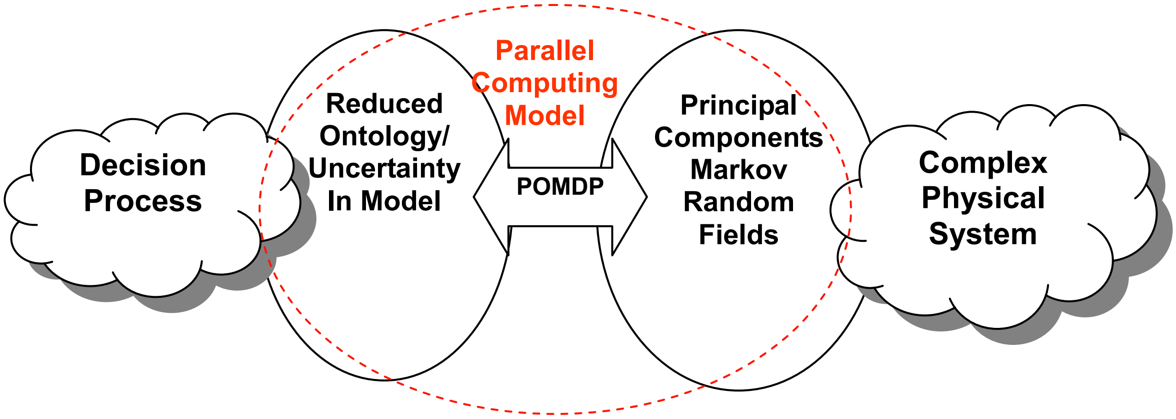

Decision Process
Reduced Ontology/ Uncertainty In Model
Parallel Computing Model
POMDP
Principal Components Markov Random Fields
Complex Physical System

Fig. 1. Process based query of information about an environment

optimal decisions with uncertain information is intractable but even approximate solutions to POMDPs have been judged NP hard. As a result we develop a parallel computing model to manage this scenario by compressing our physical model using a subdivision strategy of Markov random field modeling in conjunction with principal components analysis. Our Markov random field approach allows us to subdivide our observation space to parallel subcomponents and our principal components analysis enables us to reduce the size of our overall physical model such that we can predict our optimal action for changing and highly complex scenarios. This system is shown in Figure 1.

The next challenge of our approach to map our mathematical method on to a software architecture. We will use a multi-user publish and subscribe services architecture for our implementation. The approach is designed to enable multiple users to enter decision policy requests into our computing model and have the model optimize over the complex physical system as well as multiple users. Decision policy requests will be entered into our POMDP process as XML based schema. These schema will be translated by the POMDP process into physical process requests as a function of optimization criteria and then returned to the user after optimization. Because our architecture enables parallel execution and dynamic updating of user request policies, we are able to factor in multiple users with potential influencing dependencies in our environment and build a global decision policy over our physical model as user requests are added. Similar policy requests that enter the system after one user initially enters a policy are not recomputed but accessed in a lookup table fashion. Eventually as a global policy is reached, the amount of computation drops off dramatically as policies are accessed through lookup functions rather than recomputing the entire scenario. As underlying model data change these changes are then reflected in only those parts of the global policy that must be recomputed. Thus users that enter the system after some initial period will have relatively quick turn around to their policy requests and any new updates to the system will be computed as background processes in non real time. The computational load is governed by the fidelity needed for each user requests, the uncertainty about our models, and the rate at which the underlying models change and therefore influence our statistics. The system analogy for this process is a web-based search engine where the rank of relevant pages is pre-computed and users access the pages through lookup tables that are connected to the most recent updates. The software structure is shown below in Figure 2.

II. POMDP METHOD

POMDPs are a derivative of Markov decision processes that focus on complex scenarios that do not have complete knowledge of all states and all relevant a-priori information. We first wish to describe requirements and the complexity of adding more tasks to a single high bandwidth process and a methodology for transitioning to collaborative distributed computing processes. In order to evaluate the cost of this action our POMDP process consists of defining states s1, and s2 which consist of the belief b that an event such as a target being present or data from a communication process being available, actions a1, and a2 which consist of adding more bandwidth to our process vs. adding spatially distributed processes, and observations z1 and z2 which consist of observing whether or not our belief that an event occurred is true.

Using this approach we can assign our reward function: $r(a_i,s_{i)})$ for taking a particular action, and event probabilities $p(z_i|s_i)$ for predicting the consequence of our action. Additionally we must assign our transition probabilities between states $p(s_i|s_i,a_i.)$ of performance. We can now show our Markov state diagram for two sets of actions and a hierarchical valuation framework as is shown in Figure 3.

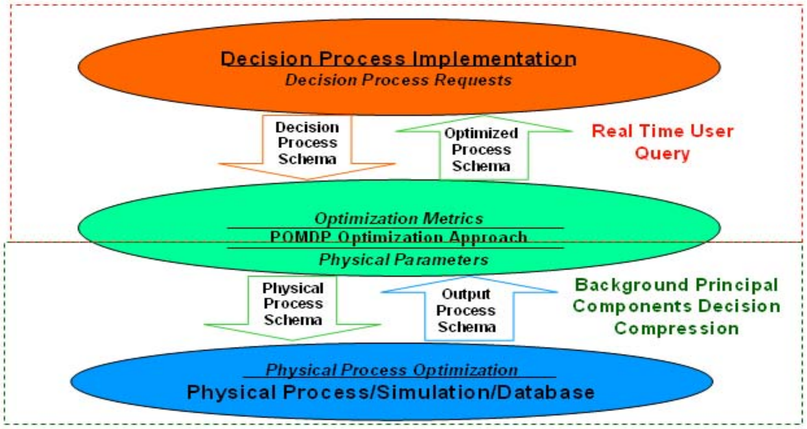

Decision Process Implementation
Decision Process Requests
Decision Process Schema
Optimized Process Schema
Real Time User Query
Optimization Metrics
POMDP Optimization Approach
Physical Parameters
Physical Process Schema
Output Process Schema
Background Principal Components Decision Compression
Physical Process Optimization
Physical Process/Simulation/Database

Fig. 2 Optimization software approach

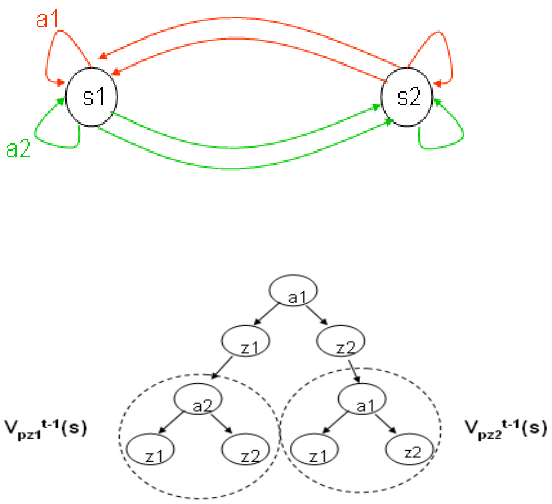


Fig. 3ab: Markov States and Valuation Framework

We now seek to compute the value of each process tree over set of beliefs b by taking expected value

$$V_p(b) = E(V_p(b)) = \sum_{|S|} p(s)V_p(s) \qquad (1)$$

The expectation is a linear function and can be expressed as a vector α. We then compute the value function $V_p^{t-1}(s)$ using a Bayesian recursive filter that steps up and down the tree in Equation 1 with

$$V_p^t(s) = \max R(s, a) + \gamma \sum_{s \in |S|} p(s'|a, s) \sum_{|Z|} p(z|s') V_{p_z}^{t-1}(s') \qquad (2)$$

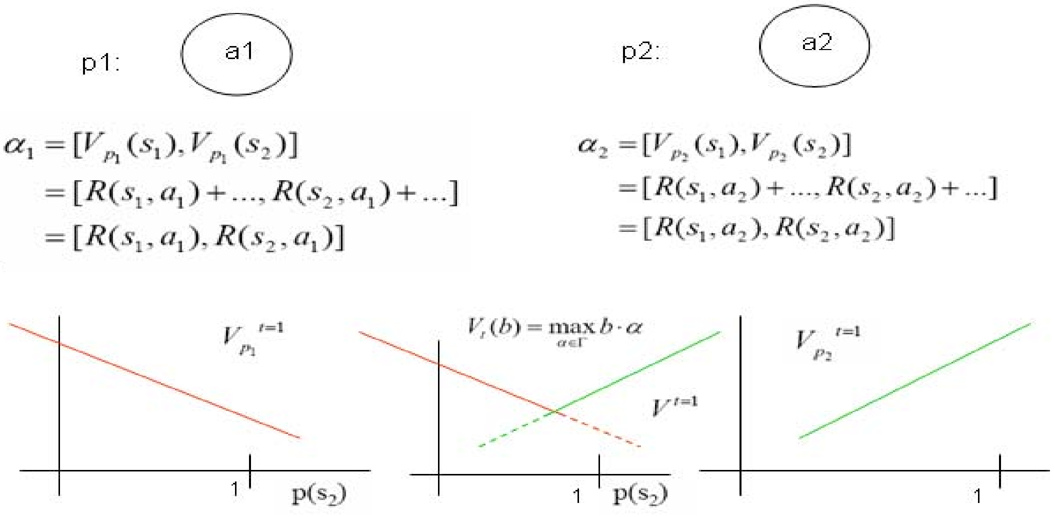
We denote γ as the weighting factor from our previous state estimate s' to our current state s. This form of equation is known as Bayesian filtering or Bayesian recursive filtering and other forms of such filters include the Kalman filter. Thus Equation 2 shows how the POMDP process takes the integrated value of our observation of our previous state $p(z|s')V_{p_z}^{t-1}(s')$ and weights this with $p(s'|a, s)$ the effect of our action on having been in the previous state s' to now being in our current state s. The reward function is a related function that allows us to guide our valuation process using statistical metrics of our process as it proceeds. We will discuss this further in section VI. As is shown in Figure 3a/b for probability state interval p1 we have an action a1 with corresponding vector α and probability state interval p2 with action a2. We now graph the function

$$V_t(b) = \max_{\alpha \in \Gamma} b\alpha \qquad (3)$$

for each action over the states s1,s2. This graph then shows us the value distributing our process to more nodes vs. the costs of adding more tasks and bandwidth to our existing process such as computing time, communication overhead, and priority of tasking a particular platform.

III. Likelihood Observation Space

We now define the method for observing the underlying factors shown as a sequence of variables $y_1 \ldots y_n$ that are the *influencing factors* of adding/subtracting additional bandwidth or platforms including processing overhead, communications, algorithm complexity, modulation scheme, and frequency allocation. Thus we can pose these underlying system



$$\alpha_1 = [V_{p_1}(s_1), V_{p_1}(s_2)]$$
$$= [R(s_1, a_1) + \ldots, R(s_2, a_1) + \ldots]$$
$$= [R(s_1, a_1), R(s_2, a_1)]$$

$$\alpha_2 = [V_{p_2}(s_1), V_{p_2}(s_2)]$$
$$= [R(s_1, a_2) + \ldots, R(s_2, a_2) + \ldots]$$
$$= [R(s_1, a_2), R(s_2, a_2)]$$

Fig. 4. Iterative process valuation algorithm

$$L(y_1, y_2, y_3, \ldots, y_N) = \frac{f_{Y1,Y2,\ldots YN}(y_1, y_2, y_3, \ldots, y_N \mid H_1)}{f_{Y1,Y2,\ldots,YN}(y_1, y_2, y_3, \ldots, y_N \mid H_0)} \tag{4.1}$$

Our observations then consist of evaluating multi-variable likelihood ratios in order to determine whether our algorithm objectives are being met. Thus in a target tracking example, if our algorithm process can be evaluated such that we have a hypothesis of target present we have:

$L(y_1, y_2, y_3, \ldots, y_N \mid H_0)$ H$_0$ – state not present – corresponds to Z$_0$ (4.2)

and target absent we have

$L(y_1, y_2, y_3, \ldots, y_N \mid H_1)$ H$_1$ – state not present – corresponds to Z$_1$ (4.3)

Additionally, as is shown in Figure 4 the addition of a-priori information into the likelihood estimation process allows us to examine a wider availability of outcomes of our decision process and reduce the uncertainty of our process. This also decreases the computational complexity of our process.

IV. Markov Random Field Basis ORDERING

Our Markov model begins by nested basis set structure is shown in dyadic form in Figure 5. The orthogonal basis functions $\phi(\omega)$ and $\psi(\omega)$ are the high and lowpass filters respectively and divide the signal in frequency by one half with each stage of the decomposition. Correspondingly the signal is downsampled by a factor of 2 at each stepA Kth order model defined on the multiresolution structure is defined in either 1 dimension with $t \in \{1, 2, \ldots, K(T+1)\}$ Such a structure for 1D signals takes the form of a binary tree structure. To represent this random field we define a given node in the binary tree structure as s, its parent node as $s\bar{\gamma}$ where $\bar{\gamma}$ shifts the basiscoefficients from parent $s\bar{\gamma}$ to child s shown in Figure 5. This set of points is denoted as Γ_s and it is the union of 2 mutually exclusive subsets. Now if we have the random variable Z representing the current state of any Γ_s at any stage of the tree in 1 dimension then we insert our local scale iterative relationship, the basic probabilistic Markov relationship is defined as

$$p_{Z_t, t \in \Gamma_{s\alpha_i} | Z_T, T \in \Gamma_s}(Z_t, t \in \Gamma_{s\alpha} | Z_T, T \in \Gamma_s) =$$

$$p_{Z_t, t \in \Gamma_{s\alpha_i} | Z_T, T \in \Gamma_{s,i}}(Z_t, t \in \Gamma_{s\alpha} | Z_T, T \in \Gamma_{s,i}) \tag{5.0}$$

Using the above multiresolution Markov structure we must develop an ordering structure for the basis coefficients before they are placed in vectors for insertion into the covariance matrix for principal components analysis. . They key is to order the coefficients so that the natural Markov structure of relevant features of the data detected. Since the Markov increments are between a scales and we wish to have a continuous basis progressing from one scale to the next we must both examine coefficients with an upward as well as downward sweep of the binary and quadtree structures.

V. Principal Components order Reduction and reward process

There are three sets of constraints that must be included in order to adequately characterize the information content of our process and resulting reward functions. Our

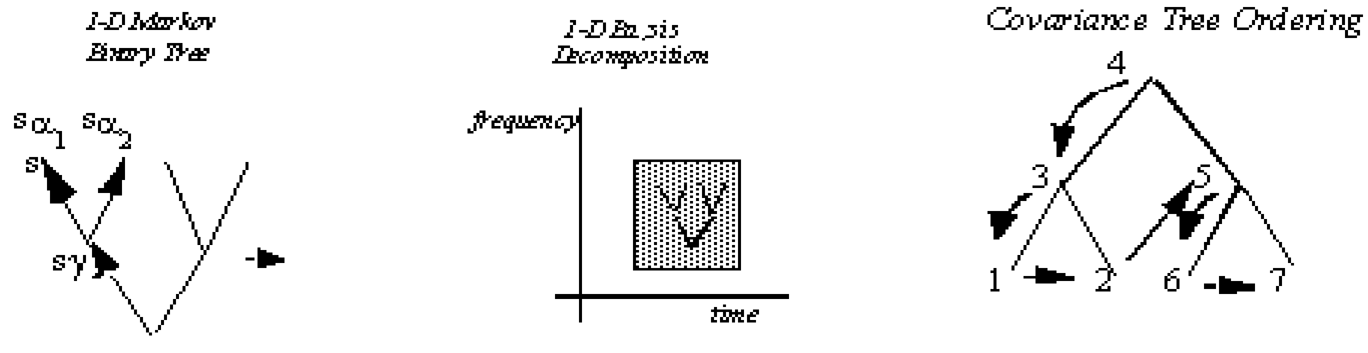


Fig. 5. Markov random field structure

rewards are characterized by computing overall computational algorithmic complexity. This has been characterized by Kolmogorov as the length of the shortest computer program that describes an object. More importantly complexity can be bounded by the Shannon capacity for random objects or sequences as follows

$$C_I = \max_{p(X)} I(X, Y) \qquad (6)$$

Since we are using a distributed computing methodology we will start our complexity metric based on the variance of the underlying density functions that characterize the complexity of adding or subtracting additional nodes or bandwidth. For each of our quantities y_i , these are usually composed of vectors of information that comprise our density functions. Thus if y_i is a vector $[s_1,\ldots\ldots\ldots,s_n]$ of samples projected onto our Markov basis sets characterizing the outputs of one of our node processes, then we can compute a covariance estimate of the vectors to characterizes the variance or entropy and stationarity of our distribution with an undistorted copy of this vector x_i with

$$\Gamma_e(x_i, y_i) = \langle x_i, y_i \rangle \qquad (7)$$

Similarly we can compute the mutual information between sample vectors sets if this information is described as

$$\Gamma_m(y_i, y_j) = \langle y_i, y_j \rangle \qquad (8)$$

If we determine the variance of the two random variables then we can expression 8 as the Fisher or mutual information between the two random variables. The inverse of this function is then the Cramer Rao bound which has been used extensively to describe spatial/spectral localization.

Thus from expressions 7 and 8 we can compute the eigenvalues (principal components) of these covariances over all density functions and samples of each random variables.

$$\hat{\lambda}_e = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{e_{ij}} \qquad (9.1)$$

$$\hat{\lambda}_m = \sum_{i=1}^{n} \sum_{j=1}^{k} \lambda_{m_{ij}} \qquad (9.2)$$

Thus we wish to set our reward functions such that the ratio

$$R(r, s) = \sup_{|S|} \frac{\hat{\lambda}_m}{\hat{\lambda}_e} \qquad (10)$$

is maximized such that our mutual information is maximized and our entropy and overall complexity is minimized until our threshold criteria of detection is met over all the possible states of our system. Figure 6 shows our ability to quantize and compress the raw information in our distributed process before evaluating the mutual information and entropy thereby reducing the computational complexity and reducing the overall complexity of our POMDP process. Our system block diagram is shown below.

Thus our reward function rewards the maximum mutual information of our process, with the minimum entropy and lowest overall computational complexity. We can see this process implemented in our process below where as we reduce the entropy in our scene we aggregate

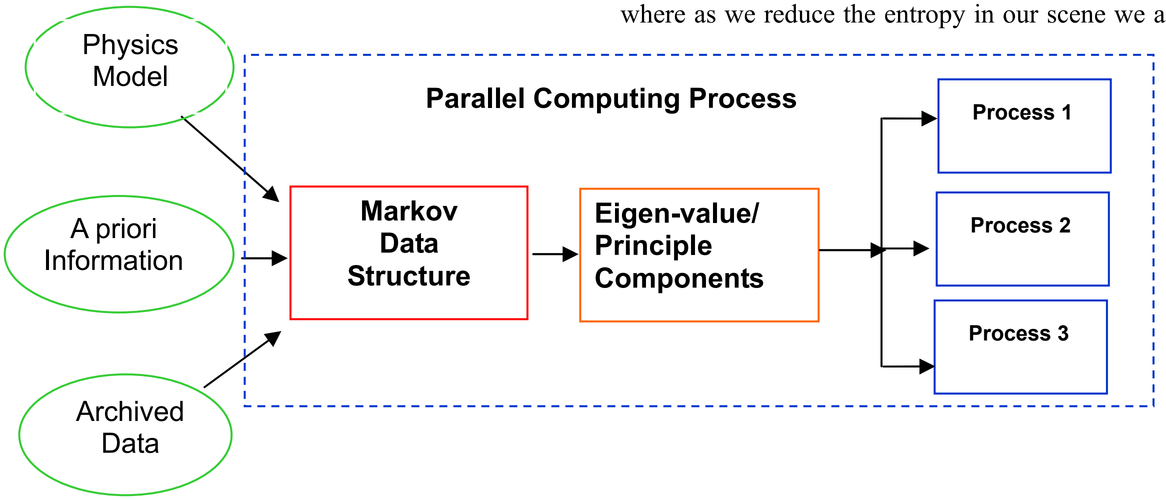


Fig. 6. Parallelization process

the number of levels and branches of our decision process and improve our estimate of our scenario based on what we already learned. This information might include building model information and the corresponding expected scattering profile. With more added information the algorithm selection process can occur more efficiently than without. Thus by using and updating our a-priori information our uncertainty is reduced our partially observable process becomes more observable. This enables us to go from highly complex estimates of our scenario to very compact or compressed estimates about the state of our scenario. This compressed state is known as our compressed belief space. This compressed belief space then can be updated with new information much more quickly given what we already know about a scenario. Our entire process is shown below.

V. DISTRIBUTED EMBEDDED EXAMPLE SCENARIO COMPUTING MODEL AND TESTING

There has been much work in distributed computing, distributed communications and distributed sensing. Our architecture will make use of a processing engine that will enable users making requests to optimize the electromagnetic environment for their application. We will describe how to optimize such an environment from a knowledge based search perspective given a-priori information taken by past sensor and communications platforms and updating with live sensor and communication information only when this information meets the users search requirements and does not duplicate existing stored data that has been previously collected.

Our goal is thus to observe our environment with new information to reduce our overall uncertainty and improve our a-priori knowledge database without duplicating information we have already retrieved. We therefore optimize our use of sensing and communications assets for new information and draw existing information from databases to improve the performance of our reconnaissance search procedure. This process of exploiting a-priori information dramatically reduces the computational overhead of our ISR search algorithms and improves their accuracy. We will also distribute the optimization among multiple user's requirements or processes to make optimal use of the platforms for all users. In communications and sensing the act of distributing the process allows us to trade spatial diversity for bandwidth by increasing the spatial sampling of our environment and therefore creating more parallel communications or sensing channels.

VI. CONCLUSION

Thus our actions consist of ISR operations, our states consist of ISR truth information, and our observations consist of detection our communication. Our algorithms are incoherent such as video data, coherent, such as synthetic aperture radar, and partially coherent such as distributed platform observation of the space in question with multiple electromagnetic sensors. Our model is a finite difference time domain and ray tracing wave propagation model and we wish to update our site specific clutter models with new information from our system. We can use his approach to improve the overall computational efficiency of analysis of large scale ISR operations.

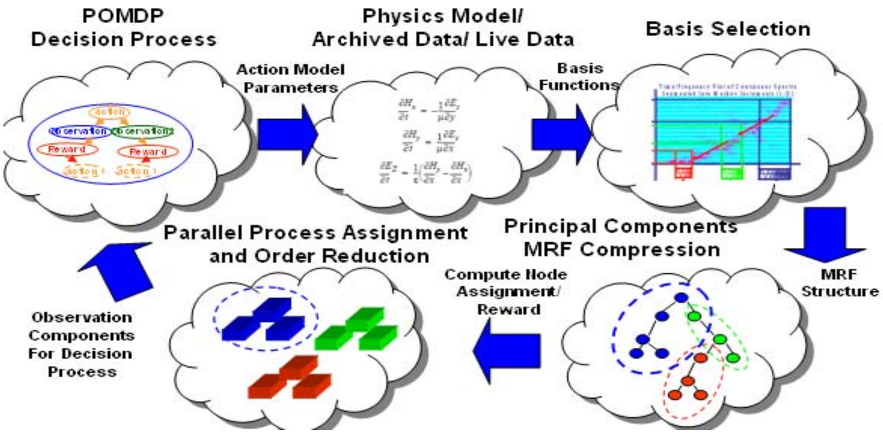


Fig. 7 Overall process block diagram

REFERENCES

[1] Bonneau, R.J., **A Waveform Strategy for Detection of Targets in Multiplicative Clutter** IEEE Radar Conference, April 2006 Page(s):84 – 93.

[2] Bonneau, R.J., **A Rate Distortion Method for Waveform Design in RF Image Formation**
Applied Imagery and Pattern Recognition Workshop, 2005. Proceedings. 34th
19-21 Oct. 2005 Page(s):63 – 69

[3] Capraro, G.T.; Capraro, C.T.; Wicks, M.C.; Liuzzi, R.A., **Artificial intelligence and waveform diversity**, Integration of Knowledge Intensive Multi-Agent Systems, 2003. International Conference on
30 Sept.-4 Oct. 2003 Page(s):270 – 274

[4] Ying He; Chong, K.P., **Sensor scheduling for target tracking in sensor networks**, Decision and Control, 2004. CDC. 43rd IEEE Conference on, Volume 1, 14-17 Dec. 2004 Page(s):743 - 748 Vol.1

Physically-Proximal Human-Robot Collaboration for Air and Space Applications

Ella M. Atkins
University of Michigan
Aerospace Engineering Dept., 1320 Beal Ave.
Ann Arbor, MI 48109
ematkins@umich.edu

***Abstract*—**In Aerospace applications, human safety is of paramount importance given harsh environmental conditions that require persistent electromechanical life support. The resulting inherent proximity between humans and "robotic support" requires effective communication and collaboration in emerging systems where the robot is not strictly a "tool" for a human operator/pilot to command. This paper investigates the challenges of human-robot collaboration in the context of two critical Aerospace applications, airspace management and planetary surface exploration. We first present a spectrum of alternative air traffic management designs ranging from centralized to fully-decentralized. Discussion focuses on roles of human versus synthetic decision-makers, associated efficiency bounds, and metrics for quantifying performance and safety. Next, a space exploration scenario is investigated in which robots and human astronauts are both modeled as "agents" with specific skills and resources available for tasking by a (computerized) planner. Emphasis is placed on real-time reconfiguration when astronauts purposely deviate from their default plan or are in need of assistance, accounting for astronaut-initiated activities while proactively enhancing astronaut safety.

***Keywords*:** *human-robot interaction, airspace management, space robotics*

I. Introduction

A "robot" is a machine that performs complicated tasks, typically a mechanism guided by automatic controls [1]. Deployed robotic platforms range from heavy industry systems on the assembly line to disaster response teams and small-scale toys or "battlebots" intended for amusement. In the context of Aerospace, a "robotic" system generally translates to a highly-automated aircraft or spacecraft. Drawing a closer physical analogy to systems we call "robots" on Earth, space robots also include planetary surface rovers and space-based manipulators. Aerospace robotic systems share a common attribute: they operate in environments that cannot sustain human life without mechanical assistance. As a consequence, vehicles are designed with the pervasive goal of maintaining safety for human occupants or collaborators as well as maintaining high levels of efficiency and productivity.

Early air and space vehicles were designed to convert human directives into actuator commands. Pilot controls were mechanically linked to control surfaces and engines, and spacecraft computers expected the uplink of detailed instructions such as thruster burn sequences or pointing and data acquisition commands. Such systems enabled humans to physically and virtually escape the Earth's surface. However, their labor-intensive operation required substantial human support just to execute the suite of "reflexive" functions analogous to the breathing, walking, and talking humans constantly perform with minimal cognitive overhead.

Once sufficiently-mature computing and sensing technologies were available, Aerospace automation efforts focused on guidance, navigation, and control (GNC), enabling the Aerospace vehicle to stabilize itself and achieve explicit spatiotemporal objectives such as "go to waypoint ***x*** at time *t*" for an aircraft or "over time interval *ti* point in direction *y*" for a spacecraft. More recently, advanced trajectory optimization and guidance laws have increased autonomy level a step further, minimizing operator overhead through translation of surveillance/science targets or waypoints to continuous-time motions [2][3][4]. Additionally, mission planning tools for air and space systems have enabled the translation of goals to activity sequences optimized over available resources and scheduled on timelines [5].

Researchers are beginning to study the interaction of multiple vehicles in the context of collaborative task/trajectory planning and cooperative control [6][7]. However, such systems currently assume a level of homogeneity in physical capabilities and response logic difficult to achieve with mixed human (human-piloted) and robot teams. Conversely, researchers specifically studying human-robot interaction (HRI) have made significant advances to facilitate human direction of robotic teams for air and space, with emphasis on the presentation of sensor data and specification of commands in an intuitive manner that focuses operator attention and maximizes human situational awareness [8]. However, in typical HRI systems the role of the "robot" is subordinate to the role of the human. Collaboration is seen through the eyes of the human operator or companion issuing a command to be directly and deterministically translated into action by the robot. As a result, the burden of inferring intent and formulating high-level goals is placed on the human, both for the human's activities and those of their robotic companion(s).

We propose an alternate model by which robotic companions are assigned the responsibility of maintaining and adapting their intent and goals. This is not a universally new idea, but it is new to air and space systems substantially constrained by safety thus high levels of conservatism with respect to automation. The goal is to increase system-level efficiency through increased levels of autonomy while maintaining a level of safety comparable to or exceeding the level of safety available in human-directed systems. Given high bandwidth and computational capacity possible in future networked air and space systems, we anticipate a level of performance that substantially exceeds that possible for a system constrained by human direction of physically complex, operationally diverse robotic vehicles and companions.

This paper introduces two Aerospace problem domains that illustrate the challenges associated with operating heterogeneous air and space vehicles efficiently in close physical proximity to one another. The first topic, airspace separation assurance, is of paramount importance to long-term aviation safety and efficiency. We demand substantial increases in air traffic capacity (thus density) than possible today, but if aircraft collide both are likely to incur sufficient damage to crash. The second topic is robot collaboration with a suited astronaut in an extraterrestrial exploration scenario. This discussion focuses on a specific strategy to plan human and robot activities but opportunistically respond when human perceptual and reasoning capabilities indicate a preference for deviating from default activity sets.

II. Aircraft Separation Assurance in Complex Airspace

Airspace management is currently performed by human air traffic controllers, augmented by local pairwise deconfliction through algorithms such as the Terminal Collision Avoidance System (TCAS, TCAS-II). This operational paradigm has been deployed long-term, but as density requirements increase, we are beginning to break the ability of both civilian and military air traffic controllers to effectively allocate airspace and manage traffic through geometrically-comprehensible corridors, queues, and minimum spacing directives.

Air travelers experienced record delays in 2007, a trend expected to worsen in the future [9]. Additionally, pressure is building to introduce highly-automated unmanned air vehicle systems (UAS) into civilian airspace. The United States Department of Defense (DoD) fields an increasingly large set of small-scale micro air vehicles (MAVs) through highly-sophisticated high-cost UAS. Efficient management of civilian and military airspace is needed. In a worst-case scenario, active battlespace environments require manned helicopters and fighters to share (low-altitude) airspace with rotary wing and fixed wing MAVs and UAS. Such heterogeneous traffic mixes are not feasible with current human-directed separation protocols, but instead will require increased local cooperation not directed by centralized human controllers. Military personnel have recognized the challenge of increasing traffic density in critical battlespace environments, particularly given the need to mix vehicles with diverse objectives at low altitudes. As described below, concepts such as dynamic (self-organizing) airspace command and control have begun to emerge, with the goal of replacing the current "airspace exclusion zones" used to organize battlespace traffic through human air traffic management.

The concept of dynamic airspace management and mixed vehicle fleets operating in close proximity, however, is just that. Realistic implementation plans for achieving superdensity operations over homogeneous and heterogeneous traffic mixes are at the early conceptual stage. Below, we describe a spectrum of possible airspace organization structures, from current centralized operations to a futuristic fully-decentralized paradigm that will mandate automated flight. Perhaps most realistic near-term solutions are partially-decentralized models, also described below. We also propose metrics for evaluating the efficiency and safety of the different airspace management options.

A. Dynamic Airspace Command and Control (C2)

The air over active military zones can quickly become congested with a variety of air vehicle types with a variety of mission objectives. Consider the Baghdad region, over which operators have ranged from civilian transport (Baghdad airport) to heavy-lift rotorcraft, manned fighters, and a suite of MAVs and UAS. All share the same low-altitude airspace, and traffic is necessarily dense during peak periods of activity. The manned fighter is fast and maneuverable but cannot easily identify the MAVs and UAS in time to avoid them. The MAVs and UAS are typically quite maneuverable but are very slow thus cannot evade a fighter moving rapidly in an unpredictable fashion. Flight envelopes differ: rotorcraft can hover but fixed-wing aircraft will stall below platform-dependent minimum airspeeds. Such traffic cannot all be organized into queues or cooperative formations due to the diversity of their flight envelopes and their missions. The result is that traffic densities are currently kept artificially low, and exclusions zones are manually generated to separate disparate operator types. A benchmark future military goal is to maintain persistent surveillance of critical areas (by MAVs and UAS) while simultaneously supporting rapid response to opportunistically-identified targets (by fighters). Achievement of this goal inherently requires mixing fighters and unmanned aircraft in the same airspace, which in turn requires resolution of the challenge to deconflict slow and fast-moving traffic when the fast-moving (manned aircraft) traffic has the highest operational priority (e.g., delivering a critical munition).

B. Airspace Organization and Deconfliction Modes

To adequately address the formidable challenges of civilian and military airspace management, the spectrum of management options must first be understood and assessed. Figure 1 illustrates this spectrum, ranging from the current "centralized" paradigm on the left to the maximally-automated "fully-decentralized" paradigm on the right. Multiple management models exist within each "class" of airspace management. The level of technological challenge typically increases moving from left to right on this

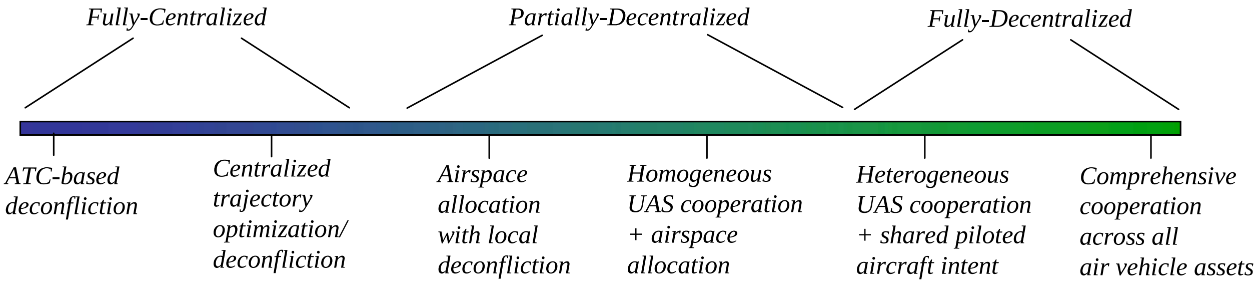

Fully-Centralized
Partially-Decentralized
Fully-Decentralized
ATC-based deconfliction
Centralized trajectory optimization/ deconfliction
Airspace allocation with local deconfliction
Homogeneous UAS cooperation + airspace allocation
Heterogeneous UAS cooperation + shared piloted aircraft intent
Comprehensive cooperation across all air vehicle assets

Fig. 1. Airspace Separation Management Architectural Spectrum.

spectrum, although different challenges are present for capable partially-decentralized versus fully-decentralized operational paradigms, as will be discussed below. The following discussion of modes and metrics is applicable to homogeneous commercial transport queues. It is also more general, encompassing mixed fleets of traffic to be managed. Traffic may be distinguished and grouped by mission (transport/transit, surveillance, special-purpose), pilot class (manned vs. unmanned), and aircraft performance envelope. Vehicles with similar missions (e.g., transiting in a common direction) and overlapping performance characteristics (e.g.., common cruise airspeed) may be densely packed, more densely with an autopilot than with direct human pilot inputs.

C. Centralized Separation Management

Since air travel became an accepted means of transportation, a centralized air traffic management system has been in place. In its current instantiation, [human] air traffic controllers and [human] pilots communicate verbally to share flight plans and requests for redirection most commonly due to pilot preference, adverse weather, or vectoring around other traffic. In open (low-density) airspace, the current system works well for transport, military, and recreational pilots. Voice communication, however, is not a feasible option for unmanned aircraft and is a low-throughput often ambiguous form of communication even for manned aircraft. As a result, although most directives are still verbally communicated, air-ground datalinks are beginning to provide pilots with improved data (e.g., weather) and to provide air traffic controllers with precise flight trajectory histories and entered flight plans.

The Figure 1 graph shows Air Traffic Controller (ATC) based deconfliction as the “entry-level” form of airspace management. This is the historical model relying on radar and verbal communication, with extensions to better distribute workload [10] or better represent data [11]. Although still centralized operationally, modern aircraft flight management systems are capable of computing and maintaining fuel/time-optimal flight plans onboard [12][13], securing an initial clearance then ATC approval for flight plan revisions to ensure the centralized air traffic database maintains an updated model of airborne aircraft position and intent. This centralized structure enables aircraft departures to be timed in a manner that greatly reduces the number of deconfliction maneuvers required, provided aircraft are able to follow their scheduled flight plans. Adverse weather, unexpected flight plan deviations, and overly-dense peak period traffic compromise efficient flows. In civilian aviation, throughput constraints given adverse weather and/or peak operational periods at major urban airports are the primary drivers to move beyond the current human-managed centralized air traffic paradigm. Military aviation experiences analogous weather and peak operation issues plus additional challenges related to an adversarial environment and mixed fleet, as discussed previously.

By its nature, centralized air traffic control requires long-range communication. Response latencies are inherent particularly due to delays associated with verbal communication and pilot/controller translation of presented data to appropriate commands or reactions. To minimize this delay, traffic is organized in queues, and flight plans are structured as a 4-D (position and time) waypoint sequence connected by constant trim state (constant climb rate, turn rate, airspeed) segments. Although intuitive, this structure is rigid and limited in extensibility by human geometric reasoning constraints.

D. Partially-Decentralized Separation Management

Over the last decade, the air transportation system has made its first incremental step into partially-decentralized operations through the use of onboard collision avoidance with TCAS and more recently an improved TCAS-II. As a supplement to the otherwise centralized air traffic control system, TCAS both warns the pilot of potential near-misses/collisions and also proposes a conflict resolution strategy. Research in local conflict resolution has made substantial theoretical and practical progress [14], and alternative concepts for efficiently routing diverse vehicle classes are being studied [15]. With the assumption that other aircraft will act unpredictably, however, the number of aircraft that can be successfully deconflicted in a common local volume is highly constrained.

As depicted in Figure 1, the “next step” toward higher-density, partially decentralized airspace management, most applicable near-term to military applications, is to group unmanned vehicles with similar missions and common performance envelopes into locally-networked cooperative teams. Such a team could be presumed to internally self-deconflict, such that the air traffic controller could treat the entire multi-vehicle team as one entity for airspace management purposes. Then, rather than allocate a static airspace volume to this team, it would fly as a dynamic "occupied airspace" volume that moved with the team and would be deconflicted dynamically with other aircraft. Researchers have proposed a variety of formation-based and consensus-based protocols for organizing unmanned aircraft in a manner that locally deconflicts the networked team [6][7]. The remaining challenge is to group these teams with other aircraft that either do not or cannot join the team. Note that a piloted aircraft is an example of a vehicle that would not typically join a cooperative control team, since joining the team would require the pilot to fly hands-off to meet the stringent inner-loop requirements imposed for stable cooperative control systems. Vehicles with dissimilar missions and/or dynamic envelopes may also be unable to cooperate. Given these substantial cooperation constraints, the best we can hope to achieve near-term is partial decentralization, with air traffic control (human or computerized) deconflicting and allocating airspace for the set of disparate teams and individual vehicles. As a result, the delays associated with centralized air traffic management are inherited by partially-decentralized systems, although density may be substantially improved for those vehicle groups able to cooperate.

E. Fully-Decentralized Separation Management

A fully-decentralized air traffic management would not rely on contact with a central authority, computerized or otherwise. As an ultimate goal to streamline air transportation and military battlespace, the challenge then is to guarantee that all air vehicles are able to communicate and reason in a manner that guarantees no collisions will occur. Unambiguous communication of intent is an incremental step beyond partial decentralization that will enable vehicles with different performance envelopes and/or goals to cooperate rather than be separated by a central authority. The mathematical goal of separation assurance is to constrain the four-dimensional space-time in which each aircraft operates such that there are no potential conflict regions. This space can be contracted to its minimal extent through team-based cooperative control as discussed above. For vehicles unable or unwilling to join these teams, however, occupation space over time can still be contracted substantially through shared intent. For an unmanned (automated) vehicle or highly-automated manned vehicle, this intent could translate to a precisely-followed future trajectory (e.g., transport aircraft landing on parallel runways).

For piloted aircraft such as fighters, no knowledge of future intent translates rapidly to no knowledge of future spatio-temporal position due to the ultra-high maneuverability and speed of these aircraft. Communication of intent, while not taking the "stick out of the pilot's hand", would then be a valuable tool to constrain the general direction and speed of travel to nearby aircraft. Consider a battlespace environment populated with surveillance UAS but with an occasional munition delivery by a manned fighter. The pilot is unlikely to follow an automated delivery trajectory at least near-term, but their intent can be known minutes in advance, enabling sufficient time for UAS to clear the delivery area. The alternatives are for the slow-moving UAS to be avoided by the higher-priority fighter, not the ideal scenario, or for the space to be manually cleared by a central controller, requiring far more overhead and time than would properly-executed UAS avoidance of the anticipated narrow delivery corridor.

Airspace management through shared intent is the highest-level stage of airspace management at which human pilots can remain in the loop. The final option, the most sophisticated but also the most technologically challenging, is for all aircraft, of all types and with all mission objectives, to self-organize in a maximally efficient and safe manner. Certainly teams would form, and altitude could be used to separate aircraft of different classes and traveling different directions. But, the hard problem is to truly maximize density and heterogeneity, given that vehicles will want to transit when they want to transit and how they want to transit. The building blocks are beginning to be formed: cooperative control, shared intent, dynamic airspace allocation when needed. For truly decentralized operation these strategies must be merged into a complete and correct architecture and augmented for all vehicles certified to occupy civilian and military airspace.

F. Airspace Management Evaluation Metrics

As we seek to increase airspace density while maintaining flexibility and safety, clear performance metrics are essential to assess tradeoffs between the different architectural as well as algorithmic options. Safety is of paramount importance, which for separation assurance can be quantitatively translated into probability of a near-miss and/or collision. This probability, in turn, is a function of airspace density and disparity (heterogeneity) between nearby vehicles. Disparity is quantified in terms of their comparative performance envelopes, onboard "intelligence" and sensing abilities (for piloted and autonomous systems), and mission objectives. Disparity is minimized for UAS of the same type with compatible objectives, the criteria for a cooperative team.

Predictability (determinism) also plays a major role in assessing the likelihood of a near-miss or collision. Any aircraft formation or transport aircraft in a busy queue relies on predictability today. The notions of cooperative control, shared intent, and locally-negotiated airspace use require predictability in a more challenging but analogous manner. As strategies evolve, both mathematical proof of algorithm correctness and reliability guarantees for software and networks will be essential.

Finally, the primary reason to push for new airspace management protocols is to improve efficiency, in terms of throughput, which translates to maximizing density, and in terms of efficiency (flight time, fuel use) for each vehicle. In terms of metrics, optimizing efficiency tends to require tradeoffs with optimizing safety. In an efficient air transportation system, density will be maximized, all aircraft will be able to fly their most efficient individual routes at the best time. Necessarily, individual optimality is relaxed to satisfy safety constraints. The quantitative goal, then, is to minimize the compromise in efficiency (and goal achievement) necessary to satisfy safety constraints over the air vehicle network.

It is important to remember that the reason we fly is to ultimately achieve a set of mission objectives. Transport aircraft passengers demand comfort, efficiency, and a "sense of safety" – not necessarily perceived even if safety is mathematically guaranteed during flight through a seemingly-chaotic group of UAS. Manned aircraft pilots still want control of their aircraft – this raises debates about the ultimate roles of pilot vs. automation, particularly during high-stress emergency situations such as damage or failure scenarios that compromise performance [16]. In the context of separation assurance, however, a manually-piloted aircraft at best constrains cooperation to "shared intent". At worst a pilot may demand complete operational freedom (e.g., to deliver a munition), requiring complete evacuation of the nearby airspace. Qualitative metrics will likely be required to assess level of pilot, operator, and passenger acceptance of alternative airspace management protocols. It is anticipated that a more "autonomous", less-centralized system will need to slowly evolve rather than be selected near-term, moving continuously along the Figure 1 spectrum rather than with discrete "jumps". This requirement, in turn, mandates smooth transitions between operational paradigms, metrics for which will evolve as concrete algorithms are readied for large-scale deployment.

III. Robot Collaboration with a Suited Astronaut

Extraterrestrial exploration currently shares few of the "application" attributes of dense airspace management – Moon and Mars will be sparsely populated for quite some time thus "traffic jams" are improbable. However, common questions emerge when modeling and quantifying the collaboration between humans and robots, the answers to which for the more "mature" airspace management applications may have relevance to emerging extraterrestrial exploration applications. How can the robot and human "agents" communicate and coordinate their activities? How are disparate goals and preferences reconciled? How do we manage, direct, and accommodate the preferences of human and robotic explorers while maximizing the safety and efficiency at which objectives are pursued and achieved?

Manned exploration missions have thusfar been modeled through the "eyes" of the mission controller and astronaut explorer as explicit mission directors. Robots provide sensor data but are explicitly tasked with accompanying and supporting their human companions. Thus, although capable planner/scheduler systems accurately assign activities to timelines, human oversight is pervasive – not through the joystick but through the "micromanagement" of goals. Such management is warranted in some cases, given practical limitations in robotic perceptual and reasoning capabilities to-date. However, given that robotic technology will continue to mature but astronaut perception will still be limited by life support systems (a spacesuit), the assumption of "one-way" command issuance may not be optimal.

We propose an alternate viewpoint for human-robot collaboration in which both human and robot are subject to explicit task assignment based on mission goals, but also in which both are able to redirect their activities independently, so long as they share their intent with mission control, likely itself a collaborative human/computational system. In this manner, human and robotic agents are both able to "decentrally" request assistance in the context of task completion of safety, and both are able to act opportunistically based on their observations.

An early prototype of a "collaborative" rather than "human-directed" planetary surface exploration architecture is described in [17]. As a "first step" to collaboration, this system presumes a centralized planning/scheduling system analogous to centralized air traffic coordination but in a fully-computerized (automated) instantiation. Task allocation and scheduling among an agent team is not new, although task allocation for humans as well as robotic systems is somewhat unconventional. Reacting to unexpected events, also supported by this architecture, is also not new. However, typically the unexpected events take the form of low-probability dynamicism in the environment, component or system failures, and actions by non-cooperative agents. As a partially-decentralized system, our human-robot agents are allowed, and in fact encouraged, to seize opportunities and ask for assistance. Thus, they freely deviate from their plans, not because they are unable to achieve the tasks assigned to them, but because they act on their "individual" initiative to improve the mission. In known environments, such "unexpected deviations" would likely be infrequent and perhaps even discouraged given the ability of the centralized agent to optimize globally rather than locally. However, extraterrestrial exploration is "exploration" because the environment is not known, thus local agents, human or robotic, will be at the forefront – able to seize data collection, sample retrieval, or even energy collection opportunities as transient events in weather and wind or in resource usage [spikes] prompt deviant action.

Below, the architecture and early prototype testing from [17] are summarized in the context of human-robot collaboration during extraterrestrial missions. Although the simulation and hardware-based experiments were necessarily limited in extent, it both demonstrated the utility of collaboration and indicated areas where additional feedback regarding resource consumption and task timings are required.

A. Architecture Overview

Figure 2 illustrates the components of the "reference" human-robot collaboration architecture we have devised to study the ability of human and robotic systems to "equally" direct their individual and cooperative activities. Analogous to a mission control or base station directing and monitoring a human/robot extravehicular activity (EVA) team, a centralized planning and execution system has been implemented. The planner directs the activities of each mobile "agent" based on agent capabilities, goals, and feedback from the agents (e.g., location, available energy). The planner also responds to directives issued by astronaut or robotic team members. Such directives range from "I need help" to "I'm doing another task" to "Please ask a rover (or astronaut) to execute a new task" (as defined in the communication).

As shown in Figure 2, our architecture is composed of three interacting components: a team planner, a coordination executive, and the mobile agents deployed in the field (or simulated to facilitate experiments with many rovers). The centralized planning module takes the current state of the world and computes a plan for all mobile agents so that the maximum number of high-priority tasks can be accomplished by the team in a minimum amount of time. The implemented design-to-time algorithm [17] can be configured to react quickly and possibly suboptimally or to consume the additional time needed for plan optimization.

The execution module controls the flow of information between planner and distributed rover-astronaut team, dispatching actions in real-time in accordance with the nominal policy and any dynamic updates. This "coordination executive" gathers and processes available state information from the rovers/astronauts to enable detection of off-nominal events. Off-nominal events are managed by either adjusting the policy (e.g., task execution timings) or by notifying the planner that plan modifications are required. Rovers are assumed to understand high-level directives (task descriptions) and either to correctly execute them or return an annotated error message indicating the task was not successfully accomplished. Our implementation is configured such that the simple simulated agents can be replaced with "real" robotic and astronaut agents. Specifics of the time-controlled planner/scheduler algorithm and coordination executive are found in [17].

B. Summary of Test Results and Observations

Each rover was simulated or interfaced (in hardware) with a single process executed as navigation, task execution, message processing, and update generation threads. In simulation, navigation included Gaussian perturbations to provide realistic deviations from the expected execution profile. Each astronaut was simulated in an analogous manner. A case with three rovers and one astronaut is shown in Figures 3 and 4. Each figure illustrates task schedules for each agent "resource" placed on a timeline. The grey regions represent traversals, while blue, yellow, green regions represent tasks such as taking pictures, acquiring samples, or measuring environmental conditions at a particular site (waypoint). Figure 3 shows a comparison between planned (lower) and actual (upper) task timelines. These deviations are not substantial and were the result of the simulated deviations from expected task execution times. With a real rover in difficult terrain, we observed substantially more discrepancy between expected (planned) and actual traversal times, illustrating a challenge for coordination when multi-agent (cooperative) activities are planned.

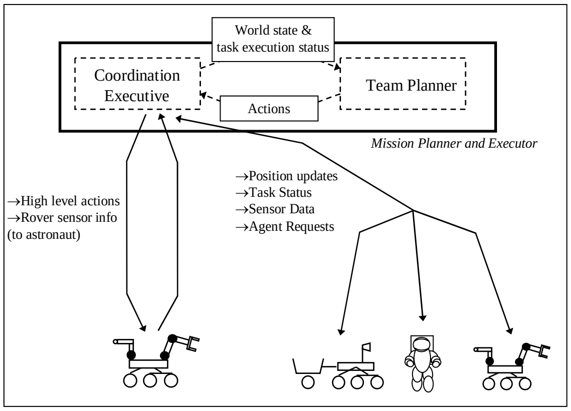


Fig. 2. Planetary Exploration Planning & Collaboration [17].

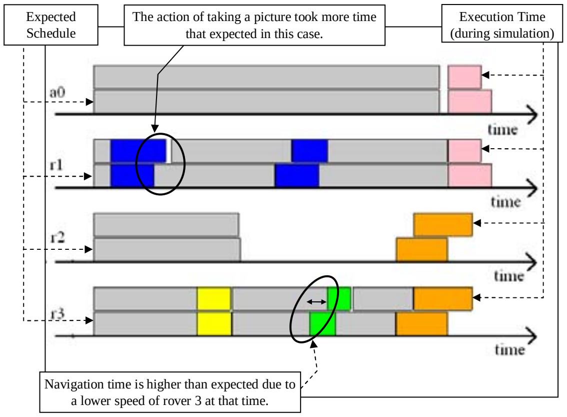


Fig. 3. Planned vs. Actual Execution Times (Simulation) [17].

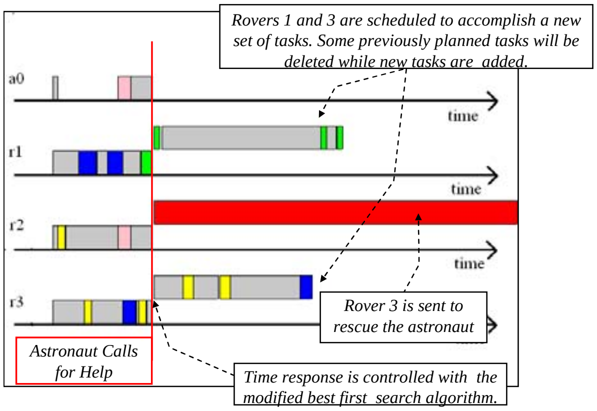


Fig. 4. Contingency Response to Astronaut Request for Help [17].

Figure 4 illustrates real-time replanning for a case where the astronaut requires assistance (e.g., injury, low on oxygen, etc.). In response, a rover (r2) is redirected to assist the astronaut (e.g., provide a spare oxygen supply). If the remaining agents are undisturbed, they could complete their originally-scheduled activities, but the activities of the astronaut and rover r2 would be ignored. Instead, the planner reschedules activities for rovers r1 and r3, directing them to complete as many high-priority tasks abandoned by r2 and the astronaut as well as their own high-priority tasks.

To identify additional challenges associated with human-robot collaboration, we performed tests with a "real" 6-wheel robot and human "astronaut". With no manipulator, the rover's "skills" were to store samples acquired by the astronaut (requiring cooperation) and take pictures/video of "interesting" sites. The astronaut could collect samples and convey perceptions of the environment to identify interesting sites, but could not carry the samples (and remain productive) or acquire picture/video data.

Overall, once baseline rover navigation and path planning algorithms were in place, tests went smoothly. The primary lesson learned, however, was that discrepancies between actual and predicted rover traversal speed can be substantial, especially in situations where the rover diverts around [unknown] obstacles. Given our collaborative task in which an astronaut collects and stores samples on the rover, either the rover or the astronaut had to wait a substantial time for the other to arrive, depending on the numerical value we set for expected rover traversal speed. This is an important issue to be resolved, likely through intermediate status reports to better synchronize agents over long-term tasks, particularly given unanticipated task execution speed-ups and delays.

D. Human-Robot Collaboration Evaluation Metrics

As with airspace management, metrics are an important means to assess alternative architectural and algorithmic options for extraterrestrial human-robot collaboration. Safety and efficiency are again the primary considerations. Safety can be quantitatively measured by response time of each agent and the planner given critical (dangerous) events. Efficiency can be measured in terms of time and resource use for individual agents and collaborative groups to accomplish tasks. Efficiency is substantially compromised when substantial delay is encountered, as with the astronaut waiting for the rover to arrive. Conversely, performance can be boosted beyond that originally considered possible through opportunistic task insertion by astronauts and rovers.

As with airspace management, any exploration architecture requires acceptance by mission controllers and astronauts as well as quantitative performance evaluation. Astronauts, mission controllers, and mission scientists will all have a strong impact on the ultimate system. We can support this decision through a series of qualitative metrics. The impact on astronauts can be measured through perceived workload and situational awareness when being tasked and during the accomplishment of individual and collaborative tasks as well as communication of opportunistic deviations and requests for assistance. As evidenced by the eventual science community acceptance of onboard data processing for spacecraft, the science community is interested in acquiring the maximum amount of high-quality data. Thus, they would also be interested in the quantitative performance evaluation (objective achievement). The goal of mission control is to accomplish mission objectives while maintaining situational awareness and safety. The key to acceptance will be to field a "collaboration architecture" that is more safe and more informative than less-collaborative alternatives.

IV. Conclusion

This paper has studied human-robot collaboration in the context of airspace management and human-astronaut planetary surface exploration. A spectrum of airspace management paradigms were presented, ranging from the current "centralized" standard to a fully-decentralized futuristic system that would support substantially more dense and disparate operations. Due to legacy and ultra-high safety requirements, aircraft are still manually routed by human controllers with local automated (pairwise) deconfliction. Migration to a more efficient "automated" system faces formidable technological, operational, and psychological challenges. Technologically, we must develop a common representational and communication framework enabling manned and unmanned aircraft of different sizes and designs to share common airspace. Operationally, we must reduce uncertainty to retain or even enhance safety given ultra-high-density traffic, requiring extension of "traffic queue" and "miles in trail" models to safe but minimal wake and maneuverability-based constraints. Psychologically, we must gain the trust of the human pilot, passenger, and operator through incremental implementation and long-term performance excellence. The keys to success are capable and correct management and coordination algorithms and implementations – anything less risks a popularized disaster that could compromise acceptance indefinitely.

Collaborative planetary surface exploration was also presented in the context of enhanced efficiency due to the introduction of "initiative-driven" rather than "human-directed" robotic companions. Our hypothesis is that human initiative is superior for opportunistic plan revision but that computerized offline scheduling is superior to human-directed scheduling (e.g., on a spreadsheet). As such, humans and robots are assigned a default, coordinated plan, but astronauts and robots, based on mission objectives, could dynamically revise their activities without argument and safety-oriented activities (e.g., astronaut rescue) would be efficiently accommodated. Although much work in communication, coordination, and knowledge representation remains, we presented a baseline architecture capable of planning and dynamically accommodating changes due to the environment or to agent-initiated plan deviations.

For both presented air and space robotic domains, a common set of performance metrics emerged that are crucial for assessing efficiency, safety, and robustness due to enhancing robotic systems with initiative and deliberation capabilities versus continuing to deploy exclusively human-directed systems. We believe such unique Aerospace challenges will serve as important drivers for truly collaborative and coordinated human-robot operations. NASA's emphasis on manned space exploration with robotic support will ultimately mandate the study of multiple "viewpoints" in human-robot collaboration, emphasizing novel technological capabilities rather than demonstrating the use of existing technology. Airport congestion and delays have resulted in a call to triple airport capacity (throughput) in the coming decades, and a nearly overwhelming insurgence of unmanned air vehicles has led to pressure for manned and unmanned aircraft to share common airspace. Physically-proximal human-robotic operations are inevitable on Earth, in the air, and in space. Elevating robotic system deliberation, awareness, and response capabilities to "see and act" independent of its human companions must be performed carefully but surely must be performed to support the demands society will place on transportation and exploration systems.

V. References

[1] Merriam-Webster OnLine (http://www.webster.com/).

[2] D. Conkey, G. Dell, G., S. Good, J. Bristow, "EO-1 Formation Flying Using Autocon™", *Proceedings of the IEEE Aerospace Conference,* Vol. 7, 55-61, 2000.

[3] G. Inalhan, M. Tillerson, J. How, "Relative Dynamics and Control of Spacecraft Formations in Eccentric Orbits", *Journal of Guidance, Control, and Dynamics*, Vol . 25, No. 1, January– February 48-59, 2002.

[4] D. Chavez-Clemente and E. Atkins, "Optimization of Tetrahedral Satellite Formations," *Journal of Spacecraft and Rockets*, Vol. 42, No. 5, July-August, 699-710, 2005.

[5] S. Chien, B. Cichy, A. Davies, D. Tran, G. Rabideau, R. Castano, R. Sherwood, D. Mandl, S. Frye, S. Shulman, J. Jones, S. Grosvenor, S., "An Autonomous Earth-Observing sensorWeb," *IEEE Intelligent Systems and Their Applications,* 20:3, 16-24, 2005.

[6] W. Ren and E. Atkins, "Distributed Multi-Vehicle Coordinated Control via Local Information Exchange," *International Journal of Robust and Nonlinear Control*, Vol. 17, Issue 10-11, pp. 1002-1033, July 2007.

[7] W. Ren, R. W. Beard, and E. M. Atkins, "Information Consensus in Multivehicle Cooperative Control: Collective Group Behavior through Local Interaction," *IEEE Control Systems Magazine,* Vol. 27, Issue 2, April, pp. 71-82, 2007.

[8] T. Fong, C. Thorpe, and C. Baur, "Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Remote Driving Tools", *Autonomous Robots* 11(1), July 2001.

[9] D. Q. Wilber, "Air Travel Delays: Bad, Getting Worse," *Washington Post,* Washington, DC, Aug. 1, 2007, Page A01.

[10] A. Yousef, and G. L. Donohue, "Temporal and Spatial Distribution of Airspace Complexity for Air Traffic Controller Workload-Based Sectorization," *4th Aviation Technology, Integration and Operations Forum,* AIAA 2004-6455, Chicago, Illinois, Sept. 20-22, 2004.

[11] B. Sridhar, S. Grabbe, K. Sheth, and K. D. Bilimoria, "Initial Study of Tube Networks for Flexible Airspace Utilization," *Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2006-6768, Keystone, Colorado, Aug. 21-24, 2006.

[12] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics,* Vol. 21, 1998.

[13] Schultz, R. L., "Three-Dimensional Trajectory Optimization for Aircraft," *Journal of Guidance, Control, and Dynamics,* Vol. 20, 1997.

[14] C. Tomlin, I. Mitchell, R. Ghosh, "Safety Verification of Conflict Resolution Maneuvers," IEEE Transactions on Intelligent Transportation Systems, Vol. 2, No. 2, Jun. 2001.

[15] M. Xue and E. M. Atkins, "Noise-Minimum Runway-Independent Aircraft Approach Design for Baltimore-Washington International Airport," *Journal of Aircraft,* American Institute of Aeronautics and Astronautics (AIAA), 43(1):39-51, Jan-Feb 2006.

[16] Y. Tang, E. Atkins, R. Sanner, "Emergency Flight Planning for a Generalized Transport Aircraft with Left Wing Damage," *Proc. Guidance, Navigation, and Control Conference,* AIAA, Hilton Head, SC, Aug. 2007.

[17] M. Ransan and E. Atkins, "Human-Robot Team Task Scheduling for Planetary Surface Missions," *Proc. Infotech@Aerospace Conference,* AIAA, Rohnert Park, CA, May 2007.

Analyzing the Performance of Distributed Algorithms

Robert N. Lass, Evan A. Sultanik and William C. Regli
Drexel University
Department of Computer Science
3141 Chestnut Street
Philadelphia, PA 19104
{urlass,eas28,regli}@cs.drexel.edu

Abstract — A large class of problems in multiagent systems can be solved by distributed constraint optimization (DCOP). Several algorithms have been created to solve these problems, however, no extensive evaluation of current DCOP algorithms on live networks exists in the literature. This paper uses DCOPolis—a framework for comparing *and deploying* DCOP software in heterogeneous environments—to contribute an analysis of two state-of-the-art DCOP algorithms solving a number of different problem types. Then, we use this empirical validation to evaluate the use of both cycle-based runtime and concurrent constraint checks.

I. Introduction

With the proliferation of small, inexpensive computers able to communicate wirelessly, the importance of distributed algorithms will likely grow in the coming years. This makes investigation of performance metrics and evaluation procedures for these types of systems particularly important. Due to the number of factors that influence the performance of a distributed system it is difficult to predict how a system will perform.

As an example, this paper examines metrics used to compare Distributed Constraint OPtimization (DCOP) algorithms. We empirically assesses cycle-based runtime, the primary (theoretical) performance metric in common use, in a number of different live network settings including MANETs.

In the remainder of this paper we first give background on DCOPs and several types of problems we investigate. We then describe a DCOP testbed that we have created to compare algorithms on live networks. We report on a series of experiments run on this testbed using the Adopt [8] and DPOP [11] algorithms. In doing so, we evaluate CBR as a predictor of actual runtime. Finally, we present an analysis of our results that suggests the coefficients to the CBR equation are actually a function of the algorithm and problem domain, which invalidates CBR (and its special-case ccc) as a general metric for comparing DCOP algorithms, but suggest that it may be useful as a metric for predicting asymptotic runtime or even for comparison if the coefficients are know.

II. Distributed Constraint Optimization

A large class of multiagent coordination and distributed resource allocation problems can be modeled as DCOP problems. DCOP has generated a lot of interest in the constraint programming community and a number of algorithms have been developed to solve DCOP problems [8], [6], [11], however, existing metrics for comparing these algorithms do not adequately capture the many intricacies inherent in solving DCOPs on live networks.

This is complicated by the fact that DCOP algorithms are currently implemented in simulation; there is no record in the literature of any significant evaluation of DCOP algorithms on live networks. Furthermore, cycle-based runtime (CBR) metric, for example, has coefficients that are meant to represent network constants, however, no reasonable values for these coefficients are yet known, and the correct values of these coefficients may dictate the ranking of DCOP algorithms. This paper explores when it is useful and when it is not useful to use these metrics.

A. Definitions

A “**DCOP**” is a problem in which a group of agents must distributedly choose values for a set of variables such that the cost of a set of constraints over the variables is either minimized or maximized.

Formally, a DCOP may be represented as a tuple $\langle A, V, \mathcal{D}, f, \alpha, \sigma \rangle$, where:

A is a set of agents;

V is a set of variables, $\{v_1, v_2, \ldots, v_{|V|}\}$;

$\mathcal{D}$ is a set of domains, $\{D_1, D_2, \ldots, D_{|V|}\}$, where each $D \in \mathcal{D}$ is a finite set containing the values to which its associated variable my be assigned;

f is a function

$$f : \bigcup_{S \in \mathcal{P}(V)} \prod_{v_i \in S} (\{v_i\} \times D_i) \to \mathbb{N} \cup \{\infty\}$$

(where “$\mathcal{P}(V)$” denotes the power set of V) that maps every possible variable assignment to a cost. This function can also be thought of as defining constraints between variables;

α is a function $\alpha : V \to A$ mapping variables to their associated agent. $\alpha(v_i) \mapsto a_j$ implies that it is agent a_j’s responsibility to assign the value of variable v_i. Note that it is not necessarily true that α is either an injection or surjection; and

σ is an operator that aggregates all of the individual f costs for all possible variable assignments. This is usually accomplished through summation:

$$\sigma(f) \mapsto \sum_{s \in \bigcup_{S \in \mathcal{P}(V)} \prod_{v_i \in S} (\{v_i\} \times D_i)} f(s).$$

The objective of a DCOP is to have each agent assign values to its associated variables in order to either minimize or maximize $\sigma(f)$.

A "**Context**" is a variable assignment for a DCOP. This can be thought of as a function mapping variables in the DCOP to their current values:

$$t : V \to (D \in \mathcal{D}) \cup \{\emptyset\}.$$

Note that a context is essentially a partial solution and need not contain values for *every* variable in the problem; therefore, $t(v_i) \mapsto \emptyset$ implies that the agent $\alpha(v_i)$ has not yet assigned a value to variable v_i. Given this representation, the "domain" (*i.e.*, the set of input values) of the function f can be thought of as the set of all possible contexts for the DCOP. Therefore, in the remainder of this paper we may use the notion of a context (*i.e.*, the t function) as an input to the f function.

B. Examples of DCOP Problems

1) Graph Coloring: Given a graph $G = \langle N, E \rangle$ and a set of colors C, assign each vertex, $n \in N$, a color, $c \in C$, such that the number of adjacent vertices with the same color is minimized. Graph coloring is a commonly-cited problem used for evaluating DCOP algorithms [8], [6].

DCOP Encoding: For each vertex $n_i \in N$, create a variable in the DCOP $v_i \in V$ with domain $D_i = C$. For each pair of adjacent vertices $\langle n_i, n_j \rangle \in E$, create a constraint of cost 1 if both of the associated variables are assigned the same color: $(\forall c \in C : f(\langle v_i, c \rangle, \langle v_j, c \rangle) \mapsto 1)$. A and α cannot be generically defined for graph coloring; they will depend on the application. Most publicly-available benchmark problem sets create one agent per variable [9].

2) Distributed Multiple Knapsack Problem (DMKP): Given a set of items of varying volume and a set of knapsacks of varying capacity, assign each item to a knapsack such that the amount of overflow is minimized. Let I be the set of items, K be the set of knapsacks, $s : I \to \mathbb{N}$ be a function mapping items to their volume, and $c : K \to \mathbb{N}$ be a function mapping knapsacks to their capacities.

DCOP Encoding: for each $i \in I$ create one variable $v_i \in V$ with associated domain $D_i = K$. Then for all possible context t:

$$f(t) \mapsto \sum_{k \in K} \begin{cases} 0 & r(t,k) \leq c(k), \\ r(t,k) - c(k) & \text{otherwise}, \end{cases}$$

where $r(t,k)$ is a function such that

$$r(t,k) = \sum_{v_i \in t^{-1}(k)} s(i).$$

C. Evaluation

Cycle-based runtime (CBR) [3], a popular and simple metric used by researchers to evaluate DCOP algorithms, is evaluated in this section. The focus was chosen to be on CBR (over other metrics such as non-concurrent constraint checks [7]) since CBR and its special-case ccc are the metrics most often employed in evaluating DCOP algorithms in the literature [8], [11], [6].

III. Experimental Setup

A. Software

The reference implementations for the Adopt and DPOP algorithms (coded by their respective authors) were designed to be run in simulation; although extending the code to be run on a live network was not hard, configuring it for automated batch processing of experiments in such a setting was non-trivial. Therefore, the implementations of these algorithms as provided in the DCOPolis[1] package were used.

DCOPolis was chosen as the testbed for our experiments because it was originally designed as framework for comparing and deploying distributed decision processes in heterogeneous environments. At the time the experiments were performed, DCOPolis had three DCOP algorithms implemented: Adopt, DPOP and a naïve algorithm called Distributed Hill Climbing. Only Adopt and DPOP were used for our experiments.

DCOPolis differentiates itself from existing frameworks and simulators (like FRODO [10] and those used in testing Adopt and OptAPO) in two fundamental ways:

1) DCOPolis was designed to allow for both simulation of DCOPs on a single computer and full deployment of DCOP solvers on many types of live networks, including traditional wired networks and ad-hoc wireless networks; and
2) DCOPolis is able to instantiate a DCOPs and start the solution process completely distributedly. This means that there is no need for configuration files, nor is there any need for a central agent/server that initializes/instantiates the rest of the group.

All of the code is freely available under the GNU public license.

B. Pseudotree Generation

A similarity between Adopt and DPOP is that they both assume the existence of a tree ordering over all of the variables in the problem. The pseudotree has an invariant that for each pair of variables $\langle v_i, v_j \rangle$ that are neighboring in the constraint graph it must be the case that v_i is either an ancestor or descendent of v_j in the pseudotree. The pseudotree also contains a backedge between all pairs of neighbors in the constraint graph that do not have a parent/child relationship in the pseudotree. For each $v \in V$, $\alpha(v)$ must know the relative tree position (*i.e.*, ancestor, descendent, or parent) of each constraint graph neighbor of v. The authors of both Adopt and DPOP assume that the agents would simply elect one

[1] http://dcopolis.sourceforge.net/

agent to create this ordering which is then broadcast to the rest of the group. Since the runtime of both algorithms is highly dependent on the structure of the pseudotree, we ensured that for each problem instance in our experiments the algorithms were given identical pseudotrees.

C. Computing Devices

Five HP-TC1100 tablet PCs with 1Ghz Intel Pentium M processors and 512M of RAM were connected via Ethernet to a Netgear FS108 switch. No machines were connected to the switch other than the ones taking part in the experiment and the switch was not connected to the Internet or any other network. All the machines were running Ubuntu 6.06 Linux with a 2.6.15-27-686 kernel.

D. Problem Datasets

1) Multiagent Task Scheduling: Experiments on the data multiagent task scheduling C_TÆMS dataset referenced in [21] was attempted, however, DPOP was unable to solve any of these problems. This was likely due to the problems' large number of variables and domain sizes. This is analyzed in §V-A.

2) Graph Coloring: The USC Teamcore project has a variety of sample problem data files in their DCOP repository [9] which were used in our analysis. The graph coloring problems were from the "Graph coloring dataset" and range from 8 to 30 variables. In these experiments, a subset of the problems containing 12 and 14 variables was used.

3) Distributed Multiple Knapsack Problem: DCOPolis has a utility for creating random DMKP data files. Twenty-five problem sets were created, consisting of five of each of the following: many small bins (ten), many small objects (twelve); few small bins (three), many large objects; few large bins, many small objects; few small bins, wide variety (high standard deviation) of objects and a wide variety of bins, many small objects. These data files are available from the authors' website.

E. Cycle-based Metrics

In the first publication introducing Distributed Constraint Satisfaction, [23], Yokoo, *et al.* evaluate algorithms by counting the number of cycles needed to determine a solution. The cost of communications is not taken into account, which the authors note and explain by stating that they do not have a standard way to compare communication costs and computational costs.

Cycle-based runtime (CBR) was introduced in [3] as a metric that takes into account the number of constraint checks performed in each cycle as well as the communications latency between cycles. CBR is computed as

$$CBR(m) = L \times m + ccc(m) \times t,$$

where t and L are constants respectively relating to compute time and communications time, m is the number of cycles, and

$$ccc(m) = \sum_{k=0}^{m} \max_{a \in A} cc(a, k),$$

where $cc(a, k)$ is the number of constraint checks performed by agent a in cycle k.

Given the fact that a single host on the network can support multiple agents (and assuming that each host has a single processor), CBR must take into account the number of machines used in the solution of the DCOP. Therefore, we propose a slight modification to CBR that accounts for the distribution of agents on the hosts:

$$ccc(m) = \sum_{k=0}^{m} \max_{h \in H} \sum_{a \in A_h} cc(a, k),$$

where H is the set of all hosts and A_h is the set of agents on host h. In other words, all agents that are running on the same host must compete for time from the single CPU, so these agents are in effect running synchronously during each cycle. Therefore, for all agents that are sharing a host we need to sum over the number of constraint checks during each cycle instead of taking the maximum. Given an experiment where $\max_{h \in H} |A_h| = 1$ (which implies $|H| \geq |A|$), the two equations are equivalent. For the remainder of the paper we shall use this augmented definition of CBR.

IV. Results and Analysis

The results of the graph coloring and the DMKP experiments can be seen in Figure 1 and 2 respectively. In both graphs, Adopt and DPOP both show a linear correlation between runtime and CBR. In Figure 3, the results of running graph coloring problems with large domains and fifty sparsely connected vertices is shown. DPOP was unable to solve many of these problems due to the algorithm running out of memory while trying to construct the massive hypercubes for this problem domain. There are DPOP variants[16], [18] that may scale better, but they are not yet implemented in DCOPolis.

Pearson's linear correlation coefficient was calculated for the runtime and the CBR metric. For each of the datasets except one we were able with 99% certainty to reject the null hypothesis that the distributions were not linearly correlated in favor of the alternate hypothesis that CBR and actual runtime are linearly correlated. Pearson's coefficient has a student's t distribution, which is what we used to test these hypotheses. Our smallest test statistic value was 4.05. The one test for which we failed to reject the null hypothesis was for the DPOP data in Figure 2. It is clear from looking at the graph, however, that the large cluster of DPOP data supports the claim that CBR is a valid metric for predicting actual runtime.

L and t were calculated empirically for each of the domains and algorithms. The average time spent sending and receiving data during each cycle was calculated and used as L. The average runtime per cycle—not counting time required for communication—was used as t. As shown in Table I, these coefficients were quite different between algorithms and problem domains.

V. Conclusions and Future Work

We have shown that CBR is an excellent predictor of asymptotic runtime. We have also shown that the L and t coefficients in the CBR metric are not in fact constant, even

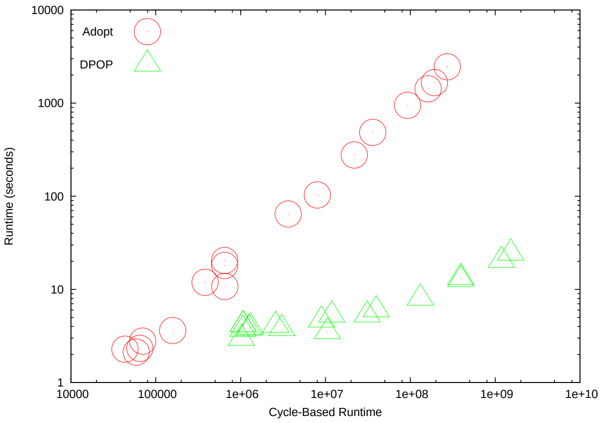


Fig. 1. Actual runtime versus cycle-based runtime for a subset of the USC Teamcore graph coloring problem set. Both Adopt and DPOP exhibit a linear correlation. Both axes are scaled logarithmically in order to reduce clustering around the origin.

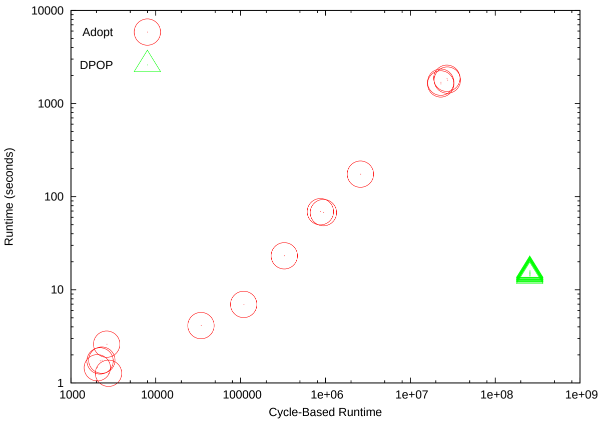


Fig. 2. Actual runtime versus cycle-based runtime for a randomly-generated set of DMK problems. Both the number of knapsacks and number of items were varied. Both Adopt and DPOP exhibit a linear correlation. Both axes are scaled logarithmically in order to reduce clustering around the origin.

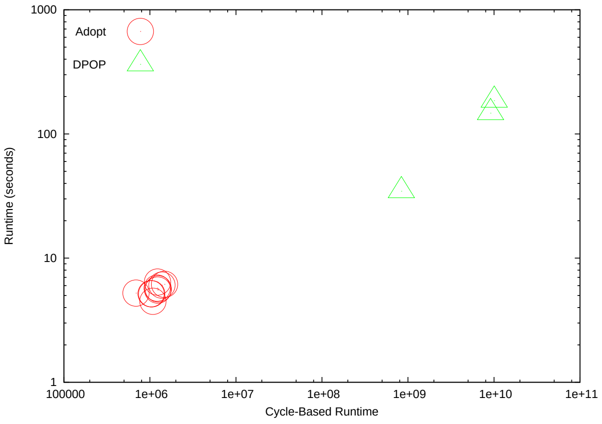


Fig. 3. Actual runtime versus cycle-based runtime for a randomly-generated set of eight-color graph coloring problems with fifty vertices. There are only three DPOP data points; the other seven failed due to a lack of memory. Both Adopt and DPOP exhibit a linear correlation. Both axes are scaled logarithmically in order to reduce clustering around the origin.

PROBLEM DOMAIN	ALGORITHM	L	t
Graph Coloring	Adopt	75.91	60.74
	DPOP	46.4	1985.69
DMKP	Adopt	54.01	68.62
	DPOP	215.78	4299.18

TABLE I
EMPIRICALLY-DETERMINED VALUES FOR THE CBR COEFFICIENTS.

when the network environment is constant. These coefficients are best represented as a function of the algorithm and the problem domain, and it is currently unclear how these can be predicted through traditional simulation. CBR therefore falls short as a metric for comparing algorithms, unless the coefficients for each algorithm are known *a priori*. We have provided a list of these coefficients for a number of different problems. In the future we hope to expand this list and also investigate new metrics such as non-concurrent constraint checks [7].

The runtime of these algorithms is highly dependent on the variable ordering given by the pseudotree. Our next experiments will be to measuring the impact of alternate techniques for generating these trees, such as [2].

DCOPolis supports the use of the Sefirs[2] simulation kernel and MATES network simulator [22], which essentially creates a virtual machine that runs in simulated time. We hope to use our live network data to calibrate these simulations to allows for a comparison of DCOP algorithms empirically in *simulation*, without the need for theoretical comparison metrics like CBR or access to a cluster of computers or testbed like the one created for this paper.

A. A note on comparisons

It is not the authors' intent to directly compare the algorithmic performance of Adopt and DPOP in this paper. The reference implementations for these algorithms (coded by their respective authors) were designed to be run in simulation; although extending the code to be run on a live network was not hard, configuring it for automated batch processing of experiments in such a setting was non-trivial. Therefore, the implementations of these algorithms as provided in the DCOPolis package were used. These implementations were created by authors other than the original algorithm designers, based solely upon the algorithms described in the respective papers. Furthermore, there are other techniques and variations of both Adopt [4], [20], [1] and DPOP [19], [18], [5], [17], [14], [13], [12], [15] that may have performed differently given our experimental datasets.

Although the data in this paper seem to suggest DCOPolis' implementation of DPOP outperforms ADOPT in terms of runtime, they are insufficient to objectively declare DPOP a better algorithm. The favorable runtimes of DPOP may be due to our selection of small problems; larger problems (*e.g.*, coloring problems with large domains and C_TÆMS problems) cannot be run with DCOPolis' implementation of DPOP because the hypercubes DPOP generates require far

[2] http://sefirs.sourceforge.net/

too much memory. DPOP's worst-case memory usage scales exponentially with respect to the average domain size [11], while Adopt scales polynomially [8]. For example, Figure 3 shows a graph of experiments that used randomly generated graph coloring problems of fifty sparsely connected vertices using eight colors. Of the ten experiments, only three completed for the DPOP algorithm; the other eight failed due to the inability to allocate enough memory. All of the Adopt problems finished.

REFERENCES

[1] Syed Ali, Sven Koenig, and Milind Tambe. Preprocessing techniques for accelerating the dcop algorithm adopt. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1041–1048, New York, NY, USA, 2005. ACM Press.

[2] Anton Chechetka and Katia Sycara. A decentralized variable ordering method for distributed constraint optimization. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1307–1308, New York, NY, USA, 2005. ACM Press.

[3] John Davin and Pragnesh Jay Modi. Impact of problem centralization in distributed constraint optimization algorithms. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1057–1063, New York, NY, USA, 2005. ACM Press.

[4] John P. Davin. Algorithmic and domain centralization in distributed constraint optimization problems. Master's Thesis, CMU-CS-05-154, CMU Tech Report, 2005.

[5] Akshat Kumar, Adrian Petcu, and Boi Faltings. H-DPOP: Using hard constraints to prune the search space. In *IJCAI'07 - Distributed Constraint Reasoning workshop, DCR'07*, Jan 2007.

[6] Roger Mailler and Victor Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 438–445, Washington, DC, USA, 2004. IEEE Computer Society.

[7] A. Meisels, E. Kaplansky, I. Razgon, and R. Zivan. Comparing performance of distributed constraints processing algorithms, 2002.

[8] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. An asynchronous complete method for distributed constraint optimization. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 161–168, New York, NY, USA, 2003. ACM Press.

[9] Jonathan P. Pearce. University of southern california DCOP repository, 2007. http://teamcore.usc.edu/dcop/.

[10] Adrian Petcu. Frodo: A framework for open/distributed constraint optimization. Technical Report No. 2006/001 2006/001, Swiss Federal Institute of Technology (EPFL), Lausanne (Switzerland), 2006. http://liawww.epfl.ch/frodo/.

[11] Adrian Petcu and Boi Faltings. A distributed, complete method for multi-agent constraint optimization. In *CP 2004 - Fifth International Workshop on Distributed Constraint Reasoning (DCR2004)*, Toronto, Canada, September 2004.

[12] Adrian Petcu and Boi Faltings. Ls-dpop: A propagation/local search hybrid for distributed optimization. In *CP 2005- LSCS'05: Second International Workshop on Local Search Techniques in Constraint Satisfaction*, Sitges, Spain, October 2005.

[13] Adrian Petcu and Boi Faltings. R-dpop: Optimal solution stability in continuous-time optimization. In *IJCAI 2005 - DCR Workshop (Distributed Constraint Reasoning)*, Edinburgh, Scotland, Aug 2005.

[14] Adrian Petcu and Boi Faltings. S-dpop: Superstabilizing, fault-containing multiagent combinatorial optimization. In *Proceedings of the National Conference on Artificial Intelligence, AAAI-05*, pages 449–454, Pittsburgh, Pennsylvania, USA, July 2005. AAAI.

[15] Adrian Petcu and Boi Faltings. O-dpop: An algorithm for open/distributed constraint optimization. In *Proceedings of the National Conference on Artificial Intelligence, AAAI-06*, pages 703–708, Boston, USA, July 2006.

[16] Adrian Petcu and Boi Faltings. Mb-dpop: A new memory-bounded algorithm for distributed optimization. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI-07*, pages 1452–1457, Hyderabad, India, Jan 2007.

[17] Adrian Petcu, Boi Faltings, and Roger Mailler. Pc-dpop: A new partial centralization algorithm for distributed optimization. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI-07*, pages 167–172, Hyderabad, India, Jan 2007.

[18] Adrian Petcu, Boi Faltings, and David Parkes. M-DPOP: Faithful distributed implementation of efficient social choice problems. *submitted to the Journal of Artificial Intelligence Research (JAIR)*, 2007. submitted.

[19] Adrian Petcu, Boi Faltings, David Parkes, and Wei Xue. BB-M-DPOP: Structural techniques for budget-balance in distributed implementations of efficient social choice. Technical report id: Lia-report-2007-002, Swiss Federal Institute of Technology (EPFL), Lausanne (Switzerland), April 2007.

[20] Marius C. Silaghi and Makoto Yokoo. Nogood based asynchronous distributed optimization (adopt ng). In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1389–1396, New York, NY, USA, 2006. ACM Press.

[21] Evan Sultanik, Pragnesh Jay Modi, and William C. Regli. On modeling multiagent task scheduling as a distributed constraint optimization problem. In *IJCAI*, pages 1531–1536, 2007.

[22] Evan A. Sultanik, Maxim D. Peysakhov, and William C. Regli. Agent transport simulation for dynamic peer-to-peer networks. Technical Report DU-CS-04-02, Drexel University, 2004.

[23] Makoto Yokoo, Edmund H. Durfee, Toru Ishida, and Kazuhiro Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *Knowledge and Data Engineering*, 10(5):673–685, 1998.

An agent structure for evaluating micro-level MAS performance

Christos Dimou, Andreas L. Symeonidis and Pericles A. Mitkas
Aristotle University of Thessaloniki
Thessaloniki, Greece
cdimou@issel.ee.auth.gr, asymeon@issel.ee.auth.gr, mitkas@eng.auth.gr

Abstract — Although the need for well-established engineering approaches in Intelligent Systems (IS) performance evaluation is urging, currently no widely accepted methodology exists, mainly due to lack of consensus on relevant definitions and scope of applicability, multi-disciplinary issues and immaturity of the field of IS. Even existing well-tested evaluation methodologies applied in other domains, such as (traditional) software engineering, prove inadequate to address the unpredictable emerging factors of the behavior of intelligent components. In this paper, we present a generic methodology and associated tools for evaluating the performance of IS, by exploiting the software agent paradigm as a representative modeling concept for intelligent systems. Based on the assessment of observable behavior of agents or multi-agent systems, the proposed methodology provides a concise set of guidelines and representation tools for evaluators to use. The methodology comprises three main tasks, namely metrics selection, monitoring agent activities for appropriate measurements, and aggregation of the conducted measurements. Coupled to this methodology is the *Evaluator Agent Framework*, which aims at the automation of most of the provided steps of the methodology, by providing Graphical User Interfaces for metrics organization and results presentation, as well as a code generating module that produces a skeleton of a monitoring agent. Once this agent is completed with domain-specific code, it is appended to the runtime of a multi-agent system and collects information from observable events and messages. Both the evaluation methodology and the automation framework are tested and demonstrated in *Symbiosis*, a MAS simulation environment for competing groups of autonomous entities.

Keywords: *performance evaluation methodology, autonomous agents, multi-agent systems, automated evaluation*

I. Introduction

Evaluation is an integral part of any complete scientific or engineering methodology. Evaluation methodologies enable researchers to test the quality and applicability of their findings, as well as to set the limits and define the appropriate environmental or intrinsic parameters for optimal performance. The benefits of a well-defined evaluation methodology lead to detection of defects, safety and overall quality of a system. But, mainly, evaluation helps researchers to thoroughly comprehend the internal characteristics and impact of their newly proposed methods and ideas.

Although the need for generalized evaluation methodologies in the field on Intelligent Systems (IS) is indisputable, currently no such effort is known to the authors. This remarkable lack of means for evaluating the performance of intelligent systems may be attributed to a number of reasons. First, it is argued that IS technology has not yet reached a certain degree of maturity. Despite being in the center of attention for more than six decades, it is only recently that Artificial Intelligence (AI) and IS are applied to realistic problems. It is, thus, evident that more experience and time are needed in order to help this field reach the desired maturity level. Second, the research area of IS combines background theory and practices from a number of diverse scientific fields, including artificial intelligence, computational theory, distributed systems, even cognitive psychology and sociology. A coordinated course of action is therefore required, one that will integrate expertise gathered from all the above mentioned areas. Moreover, existing evaluation methodologies for conventional software do not suffice in the case of IS, due to the unpredictable performance properties that are not known at design time and may emerge at the execution of IS. Finally, there is a remarkable and possibly unresolvable lack of consensus on definition of relevant terms and scope of applicability of IS. It is, indeed, very difficult to define evaluation methods when there is no agreement on even the fundamental definitions, on what an intelligent system is, what constitutes an emergent behavior or what the scope of an IS should be.

Currently, researchers and developers that desire to evaluate their systems, often devise their own ad-hoc, domain-specific evaluation methods. It is often the case that these methods are biased (most of the times with no deliberation) so that they produce the optimal results for a very strict set of environmental parameters and assumptions. Moreover, their ad-hoc nature prevents third parties to repeat the experimental setting and verify the findings.

In this paper, we present a complete, generic, domain independent methodology for evaluating IS performance, as well as a supporting software tool that automates most of the evaluation process. The proposed methodology exploits the software agent paradigm as a representative modeling concept and implementation vehicle for intelligent systems. Indeed, agents may be regarded as entities that exhibit autonomous behavior in unknown and dynamic environments [11], being capable of encapsulating any existing intelligent technology, spanning from genetic algorithms [9], data mining [14] and machine learning [12], to reinforcement learning [6] and complex decision making techniques [5]. Agents rarely operate in isolation; they most often form groups or societies, either cooperating towards a common goal [13] or competing against each other on limited resources [17]. Thus, in multi-agent systems (MAS) [3] evaluators may focus on different levels of system granularity, ranging from single agent computational units and agent autonomy to multi-agent interaction, or even on complex global cooperation/competition aspects of MAS societies. The problem of performance evaluation is then reduced to three fundamental tasks, namely a) selection of appropriate metrics, b) monitoring agent activities for appropriate measurements, and c) aggregation of the conducted measurements. Within the context of this work, we address the above tasks, by providing a concise set of methodological steps and guidelines, as well as the corresponding agent structure that autonomously performs most of the monitoring workload.

The remainder of this paper is structured as follows: Section II reviews the current state-of-the-art in IS evaluation; Section III briefly presents the scope and basic concepts of our evaluation methodology; in Section IV, the automated evaluation tool, the *Evaluator Agent*, is presented; Section V applies the proposed methodology to *Symbiosis*, a MAS simulation environment for groups of autonomous entities and illustrates the results; Section VI concludes the paper and proposes some thoughts on future directions.

II. Related Work

Evaluation has been tightly coupled with artificial intelligence, since the early days of AI [15]. On their effort to define the capabilities and limits of machines, AI pioneers defined hypothetical evaluation benchmarks in order to compare potential computer behavior against the human intellect (e.g. [10]). The initial enthusiasm soon faded (during the infamous *AI winter*), giving room for traditional software evaluation, which focused on performance and quality assessment of conventional software products, as well as related productivity metrics. It is only recently, that IS have drawn once again the attention of computer scientists and engineers, this time with more realistic goals in specific engineering problems. However, as already mentioned, current evaluation efforts do not go beyond ad-hoc solutions.

There are two general approaches to the IS evaluation problem: the bottom-up and the top-down. The former, as elaborately represented by [19], advocates the definition of formal constructs and languages that will enable the definition of the appropriate terms and scope of IS. Evaluation will thereafter be gradually built upon these formal foundations. The latter approach observes that existing or newly implemented systems urge for evaluation methodologies and it is therefore preferable to instantly evaluate them at any cost. According to this approach, experiences from different ad-hoc evaluation attempts will be generalized into a concise domain-independent methodology, which in turn will be established at the time that IS reach a sufficient maturity level.

III. The Generalized Evaluation Methodology

Our generic evaluation methodology positions itself in compliance to the above-mentioned top-down approach. Motivated by the urging need to evaluate agent systems that are deployed with currently existing techniques, we provide a complete framework that will be readily available for developers. As a generic methodology, it could not take into account intrinsic implementation details, such as specific algorithms; instead it focuses on *observable* performance, which derives from system events and messages exchanged between the system modules and components. Moreover, this methodology addresses performance issues on different levels of granularity. It is the developer's choice to identify and focus on specific performance aspects at any level of detail, ranging from independent *computational units* of agents, to *autonomous agents*, *groups of agents*, or the entire *MAS*

With respect to each of the three basic tasks of the performance evaluation process (i.e. metrics, measurement and aggregation), our methodology provides either a theoretical representation tool with an accompanying set of guidelines or automated tools that assist evaluators throughout the process (see [2] for a detailed presentation of the methodology). More specifically:

1) *Selection of metrics*. Metrics are standards that define measurable attributes of entities, their units and their scope. Before any other decision, the evaluator must choose which attributes of the system he/she is interested in. For this purpose, we provide the Metrics Representation Graph (MRG), a hierarchical representation of metrics for a specific domain. Each node of MRG corresponds to a single metric. Leaf nodes represent directly measurable metrics (*simple metrics*), i.e. metrics that can be assigned with a specific measurement value. Measurable metrics, in turn, compose higher level metrics (*composite metrics*) that cannot be assigned with a specific value, but rather represent a higher level concept that is easier understood by the evaluator, using linguistic terms. For example, the simple metrics of *numberOfMessages* and *numberOfAgents* may be composed to produce the *scalability* composite metric. Traversing the hierarchy upward, one moves to higher level composite metrics, until one reaches the root, which is the total *systemEfficiency*. The evaluator is required to traverse the MRG and select only the metrics that are of interest to him/her. The general structure of MRG is provided in Figure 1.
2) *Measurement*. Measurement is the process of ascertaining the attributes, dimensions, extend, quantity, degree of capacity of some object of observation and representing these in the qualitative or quantitative terms of a data language [8]. Having selected the appropriate metrics, measurement is the next fundamental methodological step that systematically assigns specific values to these metrics. Typical measurement methods consists of experimental design and data collection. A measurement method is the answer to the question. Since measurement methods are difficult to summarize and categorize, we provide an automated tool for producing an Evaluator Agent that autonomously monitors the execution of a

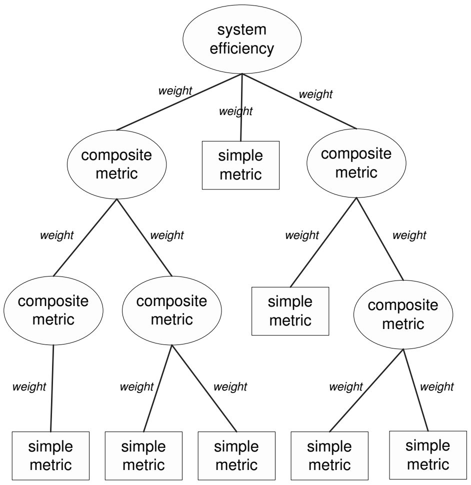


Fig. 1. General structure of the Metrics Representation Graph

system and records data related to the selected metrics. The Evaluator is presented in detail in Section IV.

3) *Fuzzy Aggregation*. Once the measurement procedure has been defined, the resulting metric-measurement pairs have to be aggregated to a single characterization for the investigated system. Aggregation, or composition, is the process of summarizing multiple measurements into a single measurement is such a manner that the output measurement will be characteristic of the system performance. Aggregation groups and combines the collected measurements, possibly by the use of weights of importance, in order to conclude to atomic characterization for the evaluated system. For example, an evaluated system may perform exceptionally well in terms of response time metrics (timeliness), but these responses may be far from correct (accuracy). Fuzzy sets [18] have been incorporated, since they provide efficient means of dealing with measurement of different scales and types, as well as of concluding to performance characterization, which is closer to the human language. Before aggregating the results, the evaluator needs to define appropriate fuzzy variables and membership function for the quantification of the measurement. He/she also has to define weights to each of the edges of the MRG, so that appropriate fuzzy aggregation operators may be later applied. These weights signify the importance of each sibling metric to the composition of the parent metric. At this moment, fuzzy quantification and weights assignment is done manually by the evaluator or a domain expert.

The complete set of steps that an evaluator is required to follow is summarized in Table I.

TABLE I
SUMMARIZATION OF METHODOLOGICAL STEPS

1.	Traverse MRG and select metrics
2.	Provide domain specific metrics (optionally)
3.	Determine metrics parameters
4.	Specify measurement method and parameters
5.	Execute experiments
6.	Define weights in the graph
7.	Define fuzzy variables and convert measurements accordingly
8.	Select and apply aggregation operators on the collected measurements

IV. EVALUATOR AGENT FRAMEWORK

In this section, we describe the general architecture and components of the *Evaluator Agent Framework*, a development framework for producing an agent structure that automates most parts of the proposed evaluation methodology. This framework is targeted to developers who need to evaluate the performance of their systems, either existing or under development. The basic requirements of this framework are the minimum modifications of the tested system at hand, as well as the minimum effort in writing evaluation specific code.

A. Purpose and Benefits

The purpose of the proposed framework is to guide the researcher/developer through the methodological steps of Table I by providing:

- visualization and manipulation of the MRG and the corresponding parameters of simple or composite metrics
- interactive specification of the fuzzy variables, membership functions and weights that correspond to the MRG
- automatic generation of the skeleton code of the *Evaluator Agent* that will undertake the task of monitoring the system for evaluation and collect all necessary information with respect to the selected metrics
- visualization of the performance evaluation results

At this point, a few of the abovementioned tasks still remain at the developers hands, as he/she is required to fill in the skeleton code of the Evaluator Agent with actual domain-specific parameters. Additionally, the developer is currently required to manually load and manipulate the MRG for the application domain at hand. At a latter phase, this will also be automated, as discussed in Section VI.

Applying the *Evaluator Agent Framework* to actual evaluation processes includes the realization of our primary motivation: the generalization of the evaluation process. By following a standardized step-wise approach for any given systems, two evaluators are expected to reach to the same conclusion. The framework will also reduce the time burden for evaluators, to tune parameters and manually monitor themselves the results. Additionally, readily available Application Programming Interfaces (APIs) for aggregating and visually presenting evaluation results may be used within any evaluation context. Fnally, this framework is envisioned to become a forum for collecting and utilizing knowledge for metrics from different domains. Developers within a domain-specific community will ideally be able to share MRGs and other experiences. Newcomers in any field will be provided with an MRG, accompanying weights and fuzzy sets from domain experts and will proceed to evaluation of their systems.

B. General Architecture

The general architecture of the proposed framework is depicted in Figure 2. The generic framework comprises five distinct components that are sequentially linked with user actions. The evaluation process initiates with the domain-specific MRG import action by the user. The MRG is presented through a interactive Graphical User Interface - GUI (Component A). After editing and manipulating the MRG, the user is provided with the skeleton of the *Evaluator Agent* (Component B). The user subsequently writes some domain specific code to produce the run-time Evaluator Agent and connects it to the system via the Evaluator Agent Middleware (Component C). The produced agent is then appended to the runtime of a MAS and starts monitoring and collecting measurement information (Component D). Upon user request or upon an end system event, the Evaluator Agent processes the logged information and produces graphs and other evaluation results (Component E), according to user preferences. Each of the components of the framework is further described in the next paragraph.

C. Components

In this subsection, the components of Figure 2 are further analyzed.

1) Component A - MRG GUI: This component is responsible for presenting the user with a visual representation of the MRG. The initiation of this process is done either by loading an existing MRG (which, ideally, is readily available from the domain communities) or by creating a new one. Node and edge manipulation tasks -such as edit and delete- are provided. For each simple or composite metric, the user may define a set of characteristic properties that are provided in drop-down menus and options. For example, for simple metrics, a user may define metric scales, units of measurement and metric types (e.g. range, boolean, nominal, ordinal etc). For composite metrics, the connection of a specific node to a set of other nodes (simple or composite metrics) with explicit edges declares that the parent node is composed of the children nodes. Through another MRG GUI option, users may also define weights for each edge, so that the participation importance of children metrics to the parent metric is determined. The set of weights will be later utilized in the fuzzy aggregation process.

After defining the structure and parameters of the MRG, the MRG GUI also provides a wizard for the determination of fuzzy variables that are necessary for the fuzzy quantification process. Appropriate fuzzy variables and corresponding membership functions are defined by the user and are correlated to each simple metric of the MRG. For example, a user may define the fuzzy variable *fastResponse* and correlate it to the simple metric *Response Time*. He/she must also provide a fuzzy membership function that maps actual measurements to the [0,1] range, as illustrated in Figure 3. It is evident that fuzzy variables and membership functions are heavily dependent on the application domain and are, currently, subjective specifications that are carried out manually by the user. The MRG GUI completes its execution by producing, upon user demand, the skeleton code of the Evaluator Agent.

The MRG GUI has been implemented as a plug-in for the Protégé Ontology Editor and Knowledge Acquisition System [4]. MRGs are represented in XML-RDF format and are loaded into the main Protégé platform. Readily available ontology visualization functions have been incorporated for the presentation of MRGs, and have been further enriched with metric-specific functionality for metric parameter, weight and fuzzy variable manipulation.

2) Component B - Skeleton Code Generator: Based on the fully defined MRG, the *Skeleton Code Generator* component is initiated in order to automatically produce the outline of an abstract, general Evaluator Agent, using the Java language. The resulting skeleton code consists of both complete and abstract classes. Complete classes implement the necessary infrasrtucture for processing, communication and logging tasks, whereas abstract classes are declared only to guide users through the domain specific addition to the Evaluator Agent.. Overall, the Skeleton Code Generator produces complete and abstract classes for:

- MRG specific manipulation functions
- metrics representation and processing functions
- tasks for collection of run-time data that correspond to the simple metrics of the selected MRG
- logging collected data into XML format
- processing XML log files
- aggregating measurement data
- communication primitives and message handling

The Skeleton Code Generator component essentially translates all the concepts that are represented in the MRG into specific or abstract code. If, for example, the user has selected metrics for the *agent level of granularity*, then the resulting code will be adjusted so that it efficiently addresses relevant single agent performance issues, such as accuracy, autonomy or timeliness. In a similar manner, if the user has selected the *MAS level of granularity*, the code will reflect performance issues, such as (possibly in addition to some of the above) scalability and modularity.

3) Component C - Evaluator Agent Middleware: The *Evaluator Agent Middleware* component serves as a connection API between the newly produced Evaluator Agent and the system under evaluation, as illustrated in Figure 4.

Since our evaluation methodology is based on the assumption that only observable behavior contains information on performance, the Evaluation Agent Middleware specifies the functions that manage observable events and observable messages, at different levels of granularity. The API provides functions for declaring, initiating, labeling and recording information on observable events and messages. Based on this API, the user is responsible to fill in the necessary code to the Skeleton Evaluator Agent, in order to produce a running Evaluator Agent. On the other hand, if it is not already implemented,

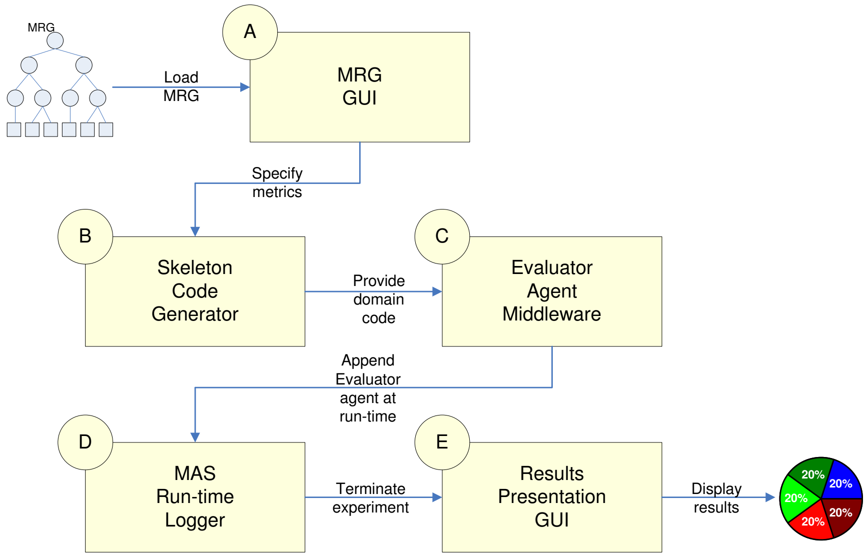


Fig. 2. Architecture of the Evaluator Agent Framework

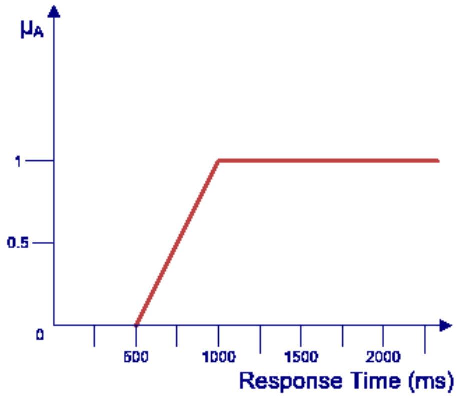


Fig. 3. Example fuzzy membership function for Response Time metric

he/she may be required to provide some additional code to the original system in order to adhere to the Evaluator Agent Middleware. After the provision of the necessary code, the Evaluator Agent is ready to be appended to the run-time of the system.

The Evaluation Agent Middleware could be implemented in any existing programming language or platform. However, for testing purposes, we have implemented this component using the Java Agent Development Environment (JADE) [1]. Jade provides a comprehensive API for agent construction, behavior specification, communication management, as well as a few very useful tools, including the Sniffer API for tracking message content and events at runtime.

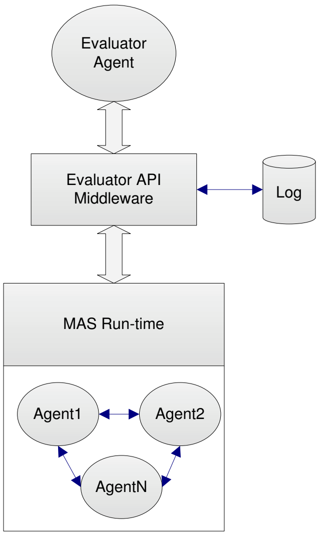


Fig. 4. Evaluator Agent Middleware

4) Component D - MAS runtime logger: This component undertakes the actual task of conducting the experimental measurement of the system's performance. The newly constructed Evaluator Agent participates as an observer agent in the MAS. On each declared event or sniffed message, the

Evaluator Agent calls the appropriate class or method in order to record performance-related information in XML format. This information may range from event or message timing to domain-specific parameter assessment, as for example the bid value in an electronic auction. The logging of this information is initiated at the designated *staring-event* and continues throughout execution until the designated *ending-event*. Both events, as well as iteration parameters, are defined by the user through the Evaluator Agent Middleware API.

5) Component E - Results Presentation GUI: For aggregation and presentation purposes, the *Results Presentation GUI* has been developed. This GUI loads one or more XML log files with all the performance information that has been recorded at runtime, as well as with the corresponding XML representation of the MRG. The user is then requested to select aggregation and presentation methods from a library of statistics, drawing, and fuzzy aggregation functions. Thus, for simple metrics a number of graphs and figures can be exported, while for composite metrics, fuzzy characterizations of parts or of the system as a whole are provided.

V. Test Case

In order to demonstrate the applicablity and validate the efficiency of the Evaluator Agent Framework, we have selected *Symbiosis* as a MAS for evaluation. In this section, a brief description is provided, and then we organize a selected set of metrics into a new instance of the MRG, apply the fuzzy quantification process and execute the experiments, using the generated Evaluator Agent.

A. Description of Symbiosis

Symbiosis [16] is mutli-agent simulation framework that follows the *animat* approach, as proposed by [7]. Animats represent autonomous, adaptive, learning entities that live and evolve in complex environments, in competition or collaboration with other animats. *Symbiosis* constitutes a virtual ecosystem, where two competing species of animats co-exist and share the environment's limited resources. Additionally, one of the two groups assumes the role of a group of *preys*, whereas the other is a group of *predators*. In addition to consumption of natural resources, predators may also consume preys. The goal of *Symbiosis* is to provide a simulation environment for testing and validating a number of emergent learning and adaptation techniques and the consequent effect of behavioral strategies.

The environment of *Symbiosis* is a $x \times y$ grid, where each cell can either be empty or occupied by:

- a natural resource, namely food, obstacle, trap
- a predator agent, or
- a prey agent

While natural resources are static, preys and predators are free to move in any neighbouring cell, aiming to maximizing their energy, either by visiting energy enhancing cells (food cells for preys and prey cells for predators) or by avoiding energy reducing cells (predator cells for preys and obstacles and traps for both species). Each agent is born with an initial amount of energy, certain vision and communication capabilities, a decision-making mechanism and reproduction abilities. The decision-making mechanisms employs genetic algorithms for the classification and evaluation of a set of action rules, based either on previous experiences or communicated from a neighbouring entity of the same species. Finally, in order to reproduce conditions that occur in real-world environments, uncertainty hs been introduced in *Symbiosis*, in the form of a parameterised vision error probability.

B. MRG Instance

The experimental measurement of the *Symbiosis* performance is based on a set of simple, measurable metrics that have been proposed and analyzed in [16]. These metrics are:

- *energy (en)*: the energy balance of an agent
- *age (ag)*: the number of *epochs* an agent lives
- *resource consumption rate (rcr)*: the ratio of energy enhancing cells that an agent has visited, to the number of total moves
- *trap collision rate (tcr)*: the ratio of trap cells that an agent has collided upon, to the number of total moves
- *unknown situation rate (usr)*: the ratio of the total unknown situation (no suitable rule applied), to the number of total moves
- *reproduction rate (rr)*: the ratio of the total offspring of an agent, to the number of total moves
- *effectiveness (e)*: the energy uptake rate minus the energy loss rate, to the energy availability rate
- *net effectiveness (e_{NET})*: the effectiveness of preys, without taking into account losses caused by their interaction with predators

After specifying the simple metrics, we need to aggregate them into composite metrics that are more comprehensible by the human evaluator. The resulting MRG is depicted in Figure 5. At depth *2*, we have concluded on the following composite metrics: *adaptability*, *scalability*, *durability* and *robustness*. All these metrics are further composed to produce both *preyEfficiency* and *predatorEfficiency* at depth *1*. Finally, the root composite metric, *MASEfficiency* is naturally composed by the two aforementioned metrics, with the addition of the *stability* simple metric, which corresponds to the deviation of total energy imbalance between the two species.

It must be noted that the weights presented in Figure 5 have been determined by a domain expert and are therefore subjective.

C. Fuzzy quantification

The next step is to determine a fuzzy variable and the corresponding membership value for each of the simple metrics defined in the previous paragraph. For simplicity and conciseness, we assign a **_high* fuzzy variable for each simple metric, where * is the name of the metric. For example, the fuzzy variable for *rcr* is named *rcr_high* and implies a high degree of resource consumption rate. Based on the knowledge provided by the domain expert, we have defined the membership function for *rcr_high*, which is depicted in

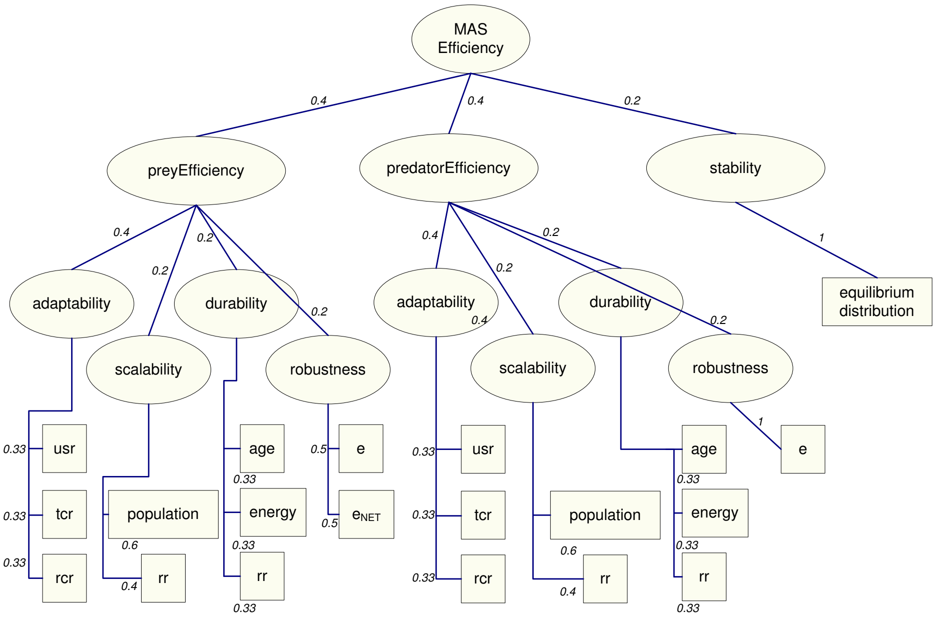


Fig. 5. An instance of the MRG for Symbiosis

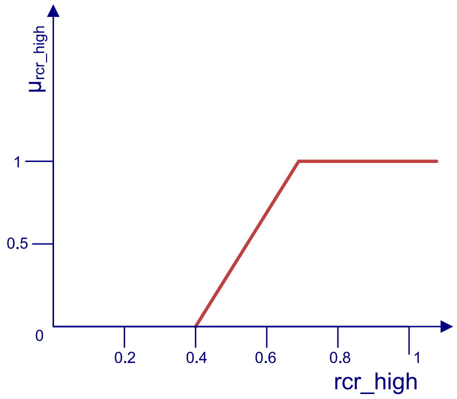


Fig. 6. Fuzzy membership function for rcr_high

Figure 6. We follow a similar approach for the rest of the simple metrics.

D. Experiments

Having completed the above steps, we continue with the automatic generation of the Skeleton Code and the actual implementation of the Evaluator Agent. Since the content of the exchanged messages between agents is not of importance to performance, the resulting agent is restricted to observe and record designated events, including food (or prey) consumption, trap collision, reproduction and unknown situations. The only burden assigned to the developer was to modify the original code in order to trigger new events in the above situations.

Two series of experiments were conducted in order to asses performance issues related to the behavior of the animats with respect to the classification mechanism and the environmental variety, respectively. In the first series of experiments, for certain environmental parameters, the impact of the classification mechanism to the system efficiency was examined. For varying values of the employed genetic algorithm invocation step (in the range of [50,500]), each of the selected simple metrics was measured. After the fuzzification and aggregation, it was determined that the highest system efficiency is a result of the genetic alogirhm step that equals to 100.

The above optimal value was then provided to the second series of experiments, which focus on prey's efficiency. A taxonomy of environments was created, as described in more detail in [16]. For each of these environments, the preys used the classification mechanism with the optimal value for the genetic algorithm step, whereas the predetators either used the same mechanism or did not use any learning mechanism at all. It was easily confirmed that Experiment B7 of the original paper provided the best results for *preyEfficiency* in total, as well as for *adaptability*, *durability* and *robustness* metrics.

Overall, the testing of the proposed evaluation methodology and the Evaluator Agent proved to be useful for carrying out preformance evaluation tasks for an already developed system. As expected, the results of this evaluation process adhere to the experimental findings of the original paper, a fact that was a principal goal of our system. Moreover, the performance of the system was analyzed in many more composite metrics, that were examined and compared in isolation of the rest of the system. This way, the evaluator may easily identify defective parts or modules of his/her system that affect the performance of the entire system. The only shortcoming of the entire experimentation process was the fact that the presence of a domain expert (in our case, the developer of the system) was necessary, both for the definition the MRG, as well as the provision of domain specific code for the Evaluator Agent.

VI. Conclusions

Driven by the urging need to provide general methodologies and tools for IS performance evaluation, we presented a novel methodology and accompanying tools for evaluating the unpredictable, emerging behavior of agents and MAS in dynamic environments. The proposed methodology provides concise methodological steps, which the evaluators may follow to guarantee a standardized and repeatable evaluation procedure. Focused on the observalble aspects of agent behavior, such as messages and events, the methodology provides representation tools for organizing, categorizing and aggregating performance metrics. Fuzzy sets have been incorporated to represent higher-level composite metrics that are more meaningful to the human evaluator.

The *Evaluator Agent Framework* was also described and demonstrated. The goal of this framework is to automate most of the steps of the above methodology, by providing GUIs for metrics and results manipulation, as well as a code generating component for automatic monitoring of observable behaviors at runtime. The produced *Evaluator Agent* monitors messages and events, while recording all performance related information for posterior processing. The Evaluator Agent Framework was successfully tested on *Symbiosis*, a MAS simulation framework for adaptive autonomous agents.

The most important future direction that emerges from this work is the automation of the MRG definition process. Currently, domain experts are being mobilized to specify the simple and composite metrics, the corresponding fuzzy variables, the fuzzy membership values and finally the weights in the MRG. Some of these tasks may be automated by exploiting information on previous evaluation efforts and historical data. It is also feasible to use this information as training datasets in order to train predefined, domain-specific MRGs. It is our vision that this effort will initiate the sharing of domain knowledge, metrics and practices towards more standardized evaluation procedures.

Acknowledgment

This paper is part of the 03ED735 research project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology and 75% from E.U.-European Social Funding).

References

[1] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi-agent systems with jade. In *Eleventh International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000)*, Boston, MA, USA, 2000.

[2] C. Dimou, A. Symeonidis, and P. Mitkas. Evaluating knowledge intensive multi-agent systems. In *Proceedings of the Autonomous Information Systems - Agents and Data Mining Conference*, St. Petersburg, Russia, 2007.

[3] W. G. *Multiagent systems. A modern approach to distributed artificial intelligence*. The MIT Press, 1999.

[4] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubzy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of protg: An environment for knowledge-based systems development.

[5] N. Jennings. An agent-based approach for building complex software systems. *Commun. ACM*, 44(4):35–41, 2001.

[6] L. P. Kaelbling, M. L. Littman, and A. P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[7] F. Krebs and H. Bossel. Emergent value orientation in self-organization of an animat. *Ecological Modelling*, 96(1):143–164, 1997.

[8] K. Krippendorff. *A Dictionary of Cybernetics*. The American Society of Cybernetics, Norfolk, VA, USA, 1986.

[9] F. Menczer, W. Street, and M. Degeratu. Evolving heterogeneous neural agents by local selection. In V. Honavar, M. Patel, and K. Balakrishnan, editors, *Advances in the Evolutionary Synthesis of Neural Systems*. MIT Press, Cambridge, MA, 2000.

[10] A. Newell and B. Buchanan. Artificial intelligence. *Encyclopedia of Science and Technology*, 2(1):146–150, 1997.

[11] H. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11:1–40, 1996.

[12] T. R. Payne and P. Edwards. Learning mechanisms for information filtering agents. In J. L. Nealon and N. S. Taylor, editors, *Proceedings of the UK Intelligent Agents Workshop*, pages 163–183, Oxford, 1997. SGES Publications.

[13] V. Renganarayanan, A. Nalla, and A. Helal. Internet agents for effective collaboration, 2001.

[14] A. Symeonidis. *Agent Intelligence through Data Mining*. Springer Science and Business Media, 2005.

[15] A. Turing. Computing machinery and intelligence. *Mind*, 59(1):443–460, 1950.

[16] F. Tzima, A. Symeonidis, and P. Mitkas. Symbiosis: Using predator-prey games as a test bed for studying competitive co-evolution. In *Proceedings of the Knowledge Intensive Multi-Agent Systems (KIMAS-07)*, Boston, MA, USA, 2007.

[17] M. Yokoo, E. Durfee, T. Ishida, and K. Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *International Conference on Distributed Computing Systems*, pages 614–621, 1992.

[18] L. Zadeh. Fuzzy sets. *Information and Control*, 8(1):338–353, 1965.

[19] L. A. Zadeh. In quest of performance metrics for intelligent systemsa challenge that cannot be met with existing methods. In *Proc. of the Third International Workshop on Performance Metrics for Intelligent Systems (PERMIS)*, 13-15 August 2002.

Information Management for High Performance Autonomous Intelligent Systems

Scott Spetka
SUNY Institute of Technology
and ITT Corp.
Utica and Rome, NY, USA
scott@cs.sunyit.edu

Scot Tucker
ITT Corp.
775 Daedalian Drive
Rome, NY, USA
Scot.Tucker@itt.com

George Ramseyer
Richard Linderman
Air Force Research Laboratory
Rome, NY, USA
George.Ramseyer@rl.af.mil

Abstract— The publish/subscribe model for information management is particularly well suited for use in intelligent autonomous systems, ranging from robots to tactical communication systems. Information management systems that support pub/sub inherently provide a high degree of autonomy for users and communicating systems. The pub/sub paradigm can allow autonomous intelligent systems to communicate without requiring connection to a centralized brokering system. Each system is responsible for part of the overall brokering function, which imposes a cost for local system resources and proportionally diminishes the intelligence that can be expressed by each node. This raises the question of whether there exist controls that each intelligent autonomous system can use to avoid over-committing resources for publication brokering, such that node intelligence is uncompromised. Issues which affect autonomy in a pub/sub system that is currently under development are addressed.

Keywords: *Quality of Service (QoS), High Performance Computing (HPC), Autonomy, Broker, Pub/Sub, Intelligent*

I. INTRODUCTION

The main advantage of publish/subscribe (pub/sub) information management systems for autonomous intelligent systems is the decoupling of senders and receivers [1]. Instead of listening to particular publishers, subscribers can specify publications they want to receive by content, based on meta-data associated with publications. Similarly, publishers submit publications without regard to exactly which subscribers will receive them or whether they are currently listening for new publications. A *broker* performs the key function of matching publications with subscribers. Brokering depends on subscription information from end users (subscribers) and knowledge of structure for performing matching functions.

Optimal brokering for pub/sub information management systems that support quality of service (QoS) constraints requires simultaneously optimizing parameters that measure a range of criteria, including: bandwidth, latency, jitter and error rates. The problem is similar to the problem of optimal routing in a multicast system, except that routing is content-dependent for pub/sub systems. Because of the Nondeterministic Polynomial-time (NP) hard nature of the problem, intelligent and heuristic approaches to routing for multi-constrained QoS multicast systems have been proposed [2][3].

The central issue for intelligent autonomous systems participating in a pub/sub brokering system while preserving a maximum degree of autonomy is the requirement that decisions be made based on a global state that can only be known through cooperation among participating brokers. But this places a requirement on the brokers to share their information and also to collect and maintain the information regarding remote systems that is needed locally. Requirements for storage, bandwidth and processing resources to support execution of intelligent algorithms and exchange of state information are generally proportional to a loss of autonomy due to participation in the system.

An intelligent autonomous pub/sub system is being developed at the Air Force Research Laboratory Information Directorate (AFRL/IF) [4]. Issues that affect autonomy and intelligence are surfacing in the system, and are being explored.

Implementing a distributed brokering service that scales well for increasing numbers of publications requires dynamically increasing resource usage as the number of publications being brokered increases. To meet QoS requirements for robustness, variable degrees of redundancy can be implemented. In addition, intelligent approaches to brokering must be considered, due to the complexity of the brokering problem in large systems and QoS constraints. Scalability for high performance information management systems provides the ability to add resources to handle increasing brokering loads on the system. Fairness issues must be considered and, when possible, measured, due to varying demands for resources to support cooperating brokers for pub/sub systems.

In the next section intelligent autonomous systems are introduced in the context of pub/sub information management systems. Then brokering architecture issues are discussed. The succeeding two sections present autonomy issues for this pub/sub architecture and for other architectures. The interplay of intelligent brokering and autonomy is discussed for each approach. Related pub/sub applications that could also be implemented in a high performance computing (HPC) environment are described. Ideas for future research are presented, and consideration of whether scalable HPC pub/sub systems can support a high degree of autonomy for participating systems is presented in the conclusion.

II. INTELLIGENT AUTONOMOUS SYSTEMS

Intelligent autonomous pub/sub systems rely on brokering functions to match publications with subscribers (Figure 1). Some of the factors that can affect the performance of brokering include: buffer space, queued messages, message input rates, bandwidth among brokers and bandwidth between brokers and system end users (publishers and subscribers). Intelligent algorithms that manage brokering functions predict and plan for future processing requirements.

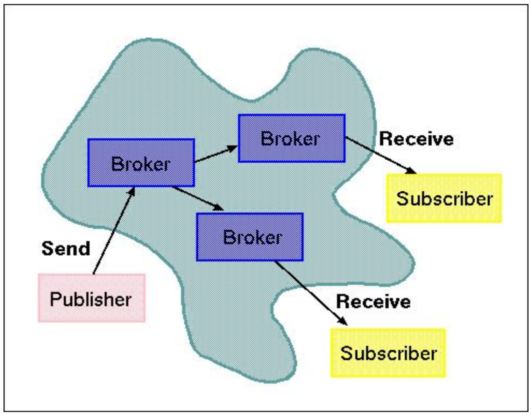


Fig. 1. Pub/Sub Information Management System

Intelligent systems are increasingly characterized by higher-level communication with understanding of content. Distributed system components receive inputs without specifying where those inputs should come from. Publications are sent without regard for the exact destinations. Some of the problems that system users face in formulating processing requests that are brokered by the system are similar to problems encountered when formulating requests to submit to an Internet search engine. For example, it may take several queries at a hardware store's Web site to find a water heater.

In a pub/sub-based system user inputs that specify a search for a local minimum or maximum on a surface could be published as a service request. There may be several subscribing services that could handle the request, depending on the degree of precision specified in the publication when the request is published. Results from each run can help the user to narrow a request, possibly by refining the required precision or by varying the search region.

In a pub/sub system, several subscriber HPC's that provide the requested service may receive the request and process it. In this case, several different responses may be received by the client that publishes the request, depending on the algorithm used for processing. It may be easy to choose the best result from the set of responses, eliminating the need for additional requests. After publishing a request for service, the user would normally wait for all processing sites to reply, to see if a good result has been returned, before sending in any further requests to refine the processing. However, lower latency can be achieved if an acceptable result is found, even if it is not optimal, so that processing can continue.

Our pub/sub architecture already implements a similar concept for reliable low-latency subscriptions. Subscribers always receive three subscriptions, through independent brokers. In this case, we know that all three will be identical, so we return the first publication received and ignore those that arrive later.

III. BROKERING ARCHITECTURE ISSUES

Our brokering architecture is designed to support the Joint Battlespace Infosphere (JBI) reference architecture [5]. The JBI specifies a common application programming interface (CAPI) for the interaction of end users, publishers and subscribers, with the system. The brokering function uses XML metadata, at least conceptually, to route publications from publishers to subscribers. As system load, measured by publications passing through the system, increases, demands on the brokering services increase. A parallel design provides scalability, which allows increasing the number of brokering nodes supporting the system.

An efficient pub/sub system that can operate across HPC systems is desirable, to allow load balancing and support processing for jobs that require more processors than may be available on any one HPC systems. Computations can also be distributed across hybrid HPC platforms when part of the computation may be performed more efficiently on particular architectures. For example, some parts of HPC codes perform better on shared memory systems, like the IBM P5, while other parts of the computation can take advantage of message passing on Linux clusters.

Resources that are contributed by a system to support distributed brokering activities on behalf of remote systems have the greatest impact on autonomy. Autonomous systems may be supporting brokering services even when there are no local publishers or subscribers. System performance will be degraded due to the support for other communicating systems that share the common pub/sub infrastructure.

Intelligent brokering systems generally require increased distributed state information at finer granularity, leading to increased storage, bandwidth and processing costs for each broker. In parallel broker designs, increased load can cause additional brokers to be dynamically added to the system. The HPC broker, implemented on a cluster computer, provides a capability for offloading processing, thereby enhancing autonomy for brokers and improving QoS processing.

IV. AUTONOMY IN HPC BROKER IMPLEMENTATIONS

Autonomy for brokers can be measured in terms of local storage, bandwidth and processing costs demanded of participating systems and also the degree to which individual systems can control their own resources. In an intelligent brokering system, cooperating brokers can offload work to other brokers, thereby improving overall system performance. However, forcing work on a broker may impact its ability to meet agreed upon QoS requirements. Of course, if the group of brokers as a whole agreed to a request for QoS, it would less affect the reputation of the underperforming broker. But, cooperative negotiations limit autonomy, by moving the decision to support a level of QoS for a publisher/subscriber from an individual broker to a committee.

The main advantage of our HPC brokering system is scalability. Within each HPC in our pub/sub environment, processing nodes may be dedicated to either brokering or other HPC applications. When additional brokering nodes are needed, due to increasing demands, in order to meet QoS requirements, they can be added at the HPC where the additional load will be supported. The decision to assign the load to a particular HPC, and whether the assigned processing load must be accepted, certainly impacts the autonomy of the system.

If HPCs make local decisions to voluntarily add brokering resources to the local broker pool, other HPCs could maintain smaller pools of broker nodes, giving them an unfair advantage. However, adding additional intelligence into decisions to increase the number of broker nodes increases overhead and can ultimately lead to committee decisions to allocate additional brokers at a given HPC, again resulting in the erosion of autonomy for the HPC which must contribute resources.

Defining and measuring autonomy for brokers in an intelligent Pub/Sub system is the key to providing QoS controls and assurance. In our HPC pub/sub implementation, increased communication requirements are gracefully supported by gradually reducing available processing resources to maintain an appropriate level of communications support for applications where processing is distributed across HPC systems. Figure 2 shows four HPC centers sharing resources to provide an execution environment for three parallel programs. One of the programs is performing digital signal processing, another is performing cryptanalysis and another is executing the Modtran atmospheric analysis program. All three applications depend on the pub/sub system, which is shown spanning all four HPC's, for their communications needs. Each of the HPC centers is making some processors available for use by the pub/sub system in supporting system-wide communications.

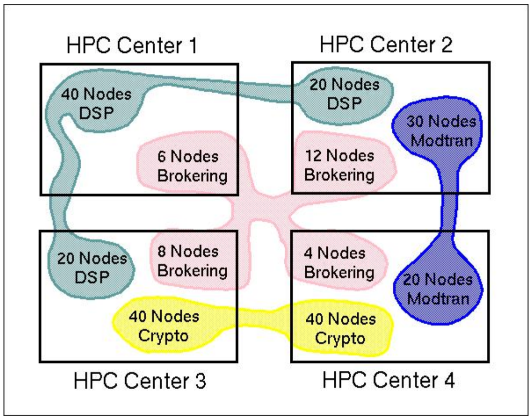


Fig. 2. Distributed Broker Architecture for HPC

V. AUTONOMY IN OTHER BROKER IMPLEMENTATIONS

Peer-to-peer networks can be used to implement pub/sub, but they naturally infringe upon the autonomy of workstations participating in distribution of messages. Increased activity for brokering on behalf of publication streams that pass through a peer system which is neither their origination nor their destination impose a load that may not be particularly welcome. The more intelligent the brokering scheme, the more processing and storage overhead are imposed on the cooperating peer. Similar concerns for autonomy have been studied for mobile peer-to-peer networks [6].

Serving as a broker for an open peer-to-peer system could also have implications for autonomy such as the loss of ability to filter messages based on content. Administrators may be responsible for transporting messages without ever approving of the users sending them or of the types of messages that they are sending. The Freenet [7] is an example of a dissemination system that is not exactly a pub/sub system. Participating Freenet sites must relinquish some of their control, especially over content. In the Freenet, "Users contribute to the network by giving bandwidth and a portion of their hard drive (called the 'data store') for storing files." Part of the mechanism which ensures the privacy of Freenet users is based on encrypting messages that are routed through the Freenet..

Some distributed broker architectures implement agent-based approaches. These approaches usually assume that agents can be decoupled from the entities that they represent. However, as in the peer-to-peer case, increases in processing, storage and communication, associated with increasingly intelligent algorithms, reduce the autonomy of participating systems that support brokering functions. Enhancing autonomy for perceptive middleware and intelligent agents is considered by Dimakic [8].

VI. RELATED WORK

There is a lot of work on QoS in pub/sub systems, but most of it pays little attention to autonomy issues. The SIENA publish/subscribe event notification service [9] is dynamically reconfigurable to adapt to the processing requirements of brokers using feedback from the on-line evaluation of performance models. SIENA routers can be dynamically added when they are needed, and routing functions can be redistributed. The idea is similar to our approach to scaling, explained above.

The Object Management Group (OMG) [10] Distributed Data Service for Real-Time Systems (DDS) standard [11] is an open international middleware standard directly addressing publish-subscribe communications for real-time and embedded systems. The DDS standard has been partially implemented with The Ace Orb (TAO) by several companies, including; Object Computing, Inc. [12], Real-Rime Innovations, Inc. [13], Prism Technologies, Inc. [14]. While autonomy is not a primary consideration for DDS, it places content filtering functions close to the end uses and brokers based on "topics". Users subscribe and publish to topics. Brokering topics minimizes the need for intelligent brokering, but increases communication costs for publications in topics which are filtered when they arrive at the subscriber.

VII. FUTURE RESEARCH

Distributed architectures afford the opportunity to assign brokering for incoming subscriptions fairly among participating brokers. In systems where acceleration techniques are used to enhance brokering services, it may be both fair and efficient to concentrate new subscriptions for implementation at a single broker, but assign batches of new subscriptions to brokers in a round robin manner.

This approach would be effective in systems where field-programmable gate arrays (FPGAs) are used to support brokering. Since it is expensive to synthesize and load a new FPGA design, the cost should be shared evenly among all brokers. It should have a minimal overall effect on publication throughput rates during update cycles, when enough new subscriptions have been received to warrant the cost of rebuilding the FPGA.

Over a longer time frame, our intelligent autonomous pub/sub-based system will need to implement a new paradigm for distributed computing that goes beyond SOAP [15] and Grid [16] protocols currently implemented for distributed computing. All routing in our system will intelligently find dynamically changing destinations for services that may help to find a solution to a problem, similar to the way that humans solve problems today.

VII. CONCLUSION

Our HPC cluster broker architecture shows that autonomy and scalability share similar characteristics, making scalable HPC architectures appear to be a good approach to implement autonomous pub/sub information management systems. The more brokers we have, the less they have to cooperate. In general, when functions are bound to particular locations, it limits autonomy by making it difficult to decide locally that a service should migrate to another system, to recover local resources. Scalability assures that additional processing resources can be used effectively and that applications are designed with component granularity that supports component migration.

We have shown that approaches to autonomy are feasible for pub/sub information management systems. More intelligence requires more knowledge of what is happening at remote brokers, loads on specific queues, etc. Scaling the brokering support provides the needed resources to support increasing intelligence in systems. Although this architecture is proven for general-purpose information management systems, we believe that it is well suited to support information management functions in other areas of autonomous intelligent distributed systems as well.

REFERENCES

[1] Combs, V., Linderman, M., "A Jini-Based Publish and Subscribe Capability", Proceedings of SPIE -- Volume 4863, Java/Jini Technologies and High-Performance Pervasive Computing, June 2002, pp. 59-69.

[2] Yuan, Xin; Liu, Xingming; "Heuristic Algorithms for Multi-constrained Quality of Service Routing" INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 2, 22-26 April 2001 Page(s):844 - 853 vol.2

[3] Wang, Junwei; Wang,Xingwei; Huang, Min, "Hybrid Intelligent QoS Multicast Routing Algorithm in NGI", Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)

[4] Madhavan , R., Messina,I, E., “Performance Metrics for Intelligent Systems (PerMIS) 2006 Workshop: Summary and Review”, Applied Imagery Pattern Recognition (AIPR) Workshop: Theory and Application of Model-based Image Analysis, Cosmos Club, Washington DC, October 11-13, 2006.

[5] Linderman, M., Combs, V.T., Hillman, R.G., Muccio, M.T., McKeel, R.W., "Joint Battlespace Inforphere (JBI): Information Management in a Netcentric Environment, AFRL-IF-RS-TR-2006-178, May 2006.
[6] Jari, Veijalainen, "Autonomy, Heterogeneity and Trust in Mobile P2P Environments", International Conference on Multimedia and Ubiquitous Engineering, 2007. MUE '07. April 2007 Page(s):41 - 47
[7] The Freenet Project - http://freenetproject.org/
[8] Dimakis, N.; Soldatos, J.; Polymenakos, L.; Schenk, M.; Pfirrmann, U.; Burkle, A.; "Perceptive Middleware and Intelligent Agents Enhancing Service Autonomy in Smart Spaces" IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2006. IAT '06. Dec. 2006 Page(s):276 - 283
[9] Caporuscio, M., Di Marco, A., Inverardi, P., "Run-time Performance Management of the Siena publish/subscribe Middleware", Proceedings of the 5th international workshop on Software and performance WOSP '05, July 2005
[10] Object Management Group - omg.org
[11] Data-Distribution Service for Real-Time Systems (DDS) - http://portals.omg.org/dds
[121] Object Computing, Inc., http://www.ociweb.com/products/dds
[13]Real-Rime Innovations, Inc., http://www.rti.com/products/data_distribution/dds_leader.html
[14] Prism Technologies, Inc. http://www.prismtechnologiesinc.com
[15] http://www.w3.org/TR/soap12-part0/
[16] http://www.ogf.org/

Efficient Monte Carlo Computation of Fisher Information Matrix using Prior Information

Sonjoy Das
University of Southern California
Los Angeles, California, USA
Sonjoy.Das@usc.edu

James C. Spall
The Johns Hopkins University
Laurel, Maryland, USA
james.spall@jhuapl.edu

Roger Ghanem
University of Southern California
Los Angeles, California, USA
ghanem@usc.edu

***Abstract*—The Fisher information matrix (FIM) is a critical quantity in several aspects of mathematical modeling, including input selection, model selection, and confidence region calculation. For example, the determinant of the FIM is the main performance metric for choosing input values in a scientific experiment with the aims of achieving the most accurate resulting parameter estimates in a mathematical model. However, analytical determination of the FIM in a general setting, especially in nonlinear models, may be difficult or almost impossible due to intractable modeling requirements and/or intractable high-dimensional integration.**

To circumvent these difficulties, a Monte Carlo (MC) simulation-based technique, resampling algorithm, based on the values of log-likelihood function or its exact stochastic gradient computed by using a set of pseudo data vectors, is usually recommended. This paper proposes an extension of the current algorithm in order to enhance the statistical characteristics of the estimator of the FIM. This modified algorithm is particularly useful in those cases where the FIM has a structure with some elements being analytically known from prior information and the others being unknown. The estimator of the FIM, obtained by using the proposed algorithm, simultaneously preserves the analytically known elements and reduces the variances of the estimators of the unknown elements by capitalizing on the information contained in the known elements.

Keywords: *Fisher information matrix, Monte Carlo simulation.*

I. Introduction and Motivating Factors

The precision matrix, a measure of accuracy of the estimates (based on a set of input values) of model parameters (to be denoted by $\theta_1, \cdots, \theta_p$) of a scientific model, plays a key role in the field of optimal design [1, Chapter 17] in which the input values to the model are selected such that $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_p]^T$ is estimated with maximum possible accuracy. Here, the superscript, T, is transpose operator. The Fisher information matrix (FIM) is often chosen as the precision matrix in the field of experimental design involving nonlinear models and, the determinant of the FIM is the most popularly used performance measure in this context. In particular, optimal design is determined as a set of inputs to the model by maximizing the determinant of the FIM over all possible inputs. Subsequently, the model parameter, $\boldsymbol{\theta}$, is updated based on the resulting set of optimal inputs.

However, the analytical determination of the FIM may be a formidable undertaking in a general setting, specially in nonlinear models, due to intractable modeling requirements and/or intractable high-dimensional integration. To avoid this analytical problem, a computational technique based on Monte Carlo (MC) simulation technique, called resampling approach [1, Section 13.3.5], [2], may be employed to estimate the FIM.

There may be also instances in practice when some elements of the FIM are analytically known from prior information while the other elements are unknown (and need to be estimated). In a recent work [3], a FIM of size 22 × 22 was observed to have the structure as shown in Fig. 1.

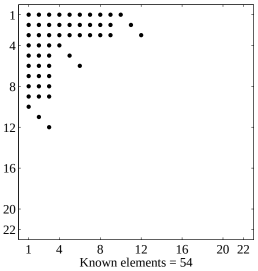


Fig. 1. Fisher information matrix with known elements as marked; void part consists of unknown elements.

In such cases, the above resampling approach, however, still yields the *full* FIM without taking any advantage of the information contained in the analytically known elements. The prior information related to the known elements of FIM is not incorporated while employing this algorithm for estimation of the unknown elements. The resampling based estimates of the known elements are also "wasted" because these estimates are simply replaced by the analytically known elements. The issue yet to be examined is whether there is a way of focusing the averaging process (required in the resampling algorithm) — on the elements of interest (unknown elements that need to be estimated) — that is more effective than simply extracting the estimates of those elements from the full FIM estimated by employing the existing resampling algorithm.

The current work presents a modified and improved (in the sense of variance reduction) version of the resampling approach for estimating the unknown elements of the FIM by "borrowing" the information contained in the analytically known elements.

II. Fisher Information Matrix: Definition and Notation

Consider a set of n random data vector (to be treated as column vector) $\{\boldsymbol{\mathcal{Z}}_1, \cdots, \boldsymbol{\mathcal{Z}}_n\}$ and stack them in $\mathbf{Z}_n$, *i.e.*,

$\mathbf{Z}_n = [\boldsymbol{\mathcal{Z}}_1^T, \cdots, \boldsymbol{\mathcal{Z}}_n^T]^T$. Let the multivariate joint probability density or mass (or hybrid density/mass) function (pdf) of $\mathbf{Z}_n$ be denoted by $p_{\mathbf{Z}_n}(\cdot|\boldsymbol{\theta})$ that is parameterized by $\boldsymbol{\theta}$. The likelihood function of $\boldsymbol{\theta}$ is then given by $\ell(\boldsymbol{\theta}|\mathbf{Z}_n) = p_{\mathbf{Z}_n}(\mathbf{Z}_n|\boldsymbol{\theta})$ and the associated log-likelihood function, L, by $L(\boldsymbol{\theta}|\mathbf{Z}_n) \equiv \ln \ell(\boldsymbol{\theta}|\mathbf{Z}_n)$.

Let us define the $p\times 1$ gradient vector, $\mathbf{g}$, of L by $\mathbf{g}(\boldsymbol{\theta}|\mathbf{Z}_n) = \partial L(\boldsymbol{\theta}|\mathbf{Z}_n)/\partial\boldsymbol{\theta}$ and the $p\times p$ Hessian matrix, $\mathbf{H}$, by $\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z}_n) = \partial^2 L(\boldsymbol{\theta}|\mathbf{Z}_n)/\partial\boldsymbol{\theta}\,\partial\boldsymbol{\theta}^T$. Then, the $p \times p$ FIM, $\mathbf{F}_n(\boldsymbol{\theta})$, is defined [1, Section 13.3.2] as follows,

$$\mathbf{F}_n(\boldsymbol{\theta}) \equiv E\left[\mathbf{g}(\boldsymbol{\theta}|\mathbf{Z}_n)\cdot\mathbf{g}^T(\boldsymbol{\theta}|\mathbf{Z}_n)\big|\,\boldsymbol{\theta}\right] = -E\left[\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z}_n)|\boldsymbol{\theta}\right], \tag{1}$$

provided that the derivatives and expectation (the expectation operator, E, is with respect to the probability measure of $\mathbf{Z}_n$) exist. The equality '$=$' in (1) is followed [1, p. 352 – 353] by assuming that L is twice differentiable with respect to $\boldsymbol{\theta}$ and the regularity conditions [4, Section 3.4.2] hold for the likelihood function, ℓ. The Hessian-based form above is more amenable to the practical computation for FIM than the gradient-based form that is used for defining the FIM.

III. Current Resampling Algoritm — No use of Prior Information

The current resampling approach is based on producing a set of large (say N) number of Hessian estimates from either the values of the log-likelihood function or (if available) its exact stochastic gradient both of which, in turn, are computed from a set of pseudo data vector, $\{\mathbf{Z}_{\text{pseudo}}(1), \cdots, \mathbf{Z}_{\text{pseudo}}(N)\}$, with each $\mathbf{Z}_{\text{pseudo}}(i)$, $i = 1, \cdots, N$, being digitally simulated from $p_{\mathbf{Z}_n}(\cdot|\boldsymbol{\theta})$ and statistically independent of each other. The set of pseudo data vector acts as a proxy for the observed data set in the resampling algorithm. The average of the negative of these Hessian estimates is reported as an estimate of the $\mathbf{F}_n(\boldsymbol{\theta})$.

For i-th pseudo data, let the k-th estimate of the Hessian matrix, $\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z}_{\text{pseudo}}(i))$, in the resampling algorithm, be denoted by $\hat{\mathbf{H}}_k^{(i)}$. Then, $\hat{\mathbf{H}}_k^{(i)}$, as per resampling scheme, is computed as [2],

$$\hat{\mathbf{H}}_k^{(i)} = \frac{1}{2}\left\{\frac{\delta\mathbf{G}_k^{(i)}}{2\,c}\left[\Delta_{k1}^{-1}, \cdots, \Delta_{kp}^{-1}\right] + \left(\frac{\delta\mathbf{G}_k^{(i)}}{2\,c}\left[\Delta_{k1}^{-1}, \cdots, \Delta_{kp}^{-1}\right]\right)^T\right\}, \tag{2}$$

in which $c > 0$ is a small number, $\delta\mathbf{G}_k^{(i)} \equiv \mathbf{G}(\boldsymbol{\theta} + c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i)) - \mathbf{G}(\boldsymbol{\theta} - c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i))$ and the perturbation vector $\boldsymbol{\Delta}_k = [\Delta_{k1}, \cdots, \Delta_{kp}]^T$ is a user-generated random vector statistically independent of $\mathbf{Z}_{\text{pseudo}}(i)$. The random variables, $\Delta_{k1}, \cdots, \Delta_{kp}$, are mean-zero and statistically independent and, also the inverse moments, $E[|1/\Delta_{km}|]$, $m = 1, \cdots, p$, are finite.

The symmetrizing operation (the multiplier $1/2$ and the indicated sum) as shown in (2) is useful in optimization problems to compute a symmetric estimate of the Hessian matrix with finite samples [2]. This also maintains a symmetric estimate of $\mathbf{F}_n(\boldsymbol{\theta})$, which itself is a symmetric matrix.

Depending on the setting, $\mathbf{G}(\cdot|\mathbf{Z}_{\text{pseudo}}(i))$, as required in $\delta\mathbf{G}_k^{(i)}$, represents the k-th direct measurement or approximation of the gradient vector, $\mathbf{g}(\cdot|\mathbf{Z}_{\text{pseudo}}(i))$. If the direct measurement or computation of $\mathbf{g}$ is feasible, $\mathbf{G}(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i))$ represent the *direct* k-th measurements of $\mathbf{g}(\cdot|\mathbf{Z}_{\text{pseudo}}(i))$ at $\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k$. Otherwise, $\mathbf{G}(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i))$ represents the k-th *approximation* of $\mathbf{g}(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i))$ based on the values of $L(\cdot|\mathbf{Z}_{\text{pseudo}}(i))$.

If the direct measurements or computations of $\mathbf{g}$ are not feasible, $\mathbf{G}$ in (2) can be computed by using the classical finite-difference (FD) technique [1, Section 6.3] or the simultaneous perturbation (SP) gradient approximation technique [5], [1, Section 7.2] from the values of $L(\cdot|\mathbf{Z}_{\text{pseudo}}(i))$. For the computation of gradient approximation based on the values of L, there are advantages to using *one-sided* [1, p. 199] SP gradient approximation (relative to the standard two-sided SP gradient approximation) in order to reduce the total number of function measurements or evaluations for L. The SP technique for gradient approximation is quite useful when p is large and usually superior to FD technique when the objective is to estimate $\mathbf{F}_n(\boldsymbol{\theta})$ by employing the resampling algorithm. The formula for the one-sided gradient approximation using SP technique is given by,

$$\mathbf{G}^{(1)}(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i)) = (1/\tilde{c})\Big[L(\boldsymbol{\theta} + \tilde{c}\tilde{\boldsymbol{\Delta}}_k \pm c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i)) - L(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k|\mathbf{Z}_{\text{pseudo}}(i))\Big]\begin{bmatrix}\tilde{\Delta}_{k1}^{-1}\\ \vdots \\ \tilde{\Delta}_{kp}^{-1}\end{bmatrix}, \tag{3}$$

in which superscript, (1), in $\mathbf{G}^{(1)}$ indicates that it is one-sided gradient approximation ($\mathbf{G} = \mathbf{G}^{(1)}$), $\tilde{c} > 0$ is a small number and $\tilde{\boldsymbol{\Delta}}_k = [\tilde{\Delta}_{k1}, \cdots, \tilde{\Delta}_{kp}]^T$ is generated in the same statistical manner as $\boldsymbol{\Delta}_k$, but otherwise statistically independent of $\boldsymbol{\Delta}_k$ and $\mathbf{Z}_{\text{pseudo}}(i)$. It is usually recommended that $\tilde{c} > c$.

At this stage, let us also formally state that the perturbation vectors, $\boldsymbol{\Delta}_k$ and $\tilde{\boldsymbol{\Delta}}_k$, satisfy the following condition,[1, Chapter 7].

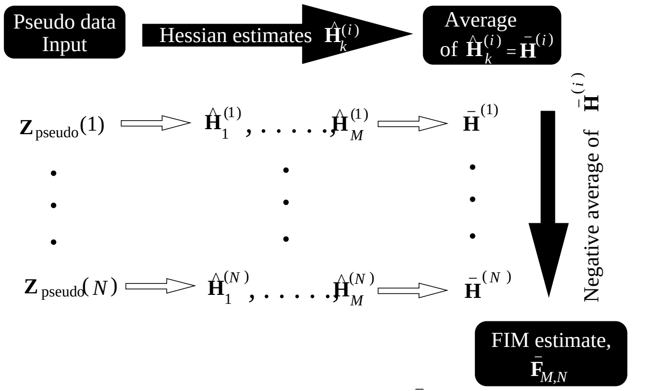
C.1: **(Statistical properties of the perturbation vector)** The random variables, Δ_{km} (and $\tilde{\Delta}_{km}$), $k = 1, \cdots, N$, $m = 1, \cdots, p$, are statistically independent and almost surely (a.s.) uniformly bounded for all k, m, and, are also mean-zero and symmetrically distributed satisfying $E[|1/\Delta_{km}|] < \infty$ (and $E[|1/\tilde{\Delta}_{km}|] < \infty$).

Let us also assume that the moments of Δ_{km} and $1/\Delta_{km}$ (and, of $\tilde{\Delta}_{km}$ and $1/\tilde{\Delta}_{km}$) up to fifth order exist (this condition will be used later in Section IV-C). Since Δ_{km} (and $\tilde{\Delta}_{km}$) is symmetrically distributed, $1/\Delta_{km}$ (and $1/\tilde{\Delta}_{km}$) is also symmetrically distributed implying that,

I: **(Statistical properties implied by C.1)** All the odd moments of Δ_{km} and $1/\Delta_{km}$ (and of $\tilde{\Delta}_{km}$ and $1/\tilde{\Delta}_{km}$) up to fifth order are zeros, $E[(\Delta_{km})^q] = 0$ and $E[(1/\Delta_{km})^q] = 0$ ($E[(\tilde{\Delta}_{km})^q] = 0$ and $E[(1/\tilde{\Delta}_{km})^q] = 0$) , $q = 1, 3, 5$.

The random vectors, $\boldsymbol{\Delta}_k$ (and $\tilde{\boldsymbol{\Delta}}_k$), are also independent across k. The random variables, $\Delta_{k1}, \cdots, \Delta_{kp}$ (and $\tilde{\Delta}_{k1}, \cdots, \tilde{\Delta}_{kp}$), can also be chosen identically distributed. In fact, independent and identically distributed (i.i.d.) (across both k and m) mean-zero random variable satisfying C.1 is a perfectly valid choice for Δ_{km} (and $\tilde{\Delta}_{km}$). In particular, Bernoulli ± 1 random variable for Δ_{km} (and $\tilde{\Delta}_{km}$) is a valid — but not the necessary — choice among other probability distributions satisfying C.1.

The current resampling algorithm is schematically shown in Fig. 2. As shown in this figure, it is preferred to generate several Hessian estimates by generating more than one (say, $M > 1$) perturbation vectors, $\boldsymbol{\Delta}_1, \cdots, \boldsymbol{\Delta}_M$, for each pseudo data vector, $\mathbf{Z}_{\text{pseudo}}(i)$, if the pseudo data vectors are expensive to simulate relative to the Hessian estimate. However, it should be noted that $M = 1$ has certain optimality properties [2] and it is assumed throughout this work that for each pseudo data vector, *only one* perturbation vector is generated and, thus, *only one* Hessian estimate is computed. The current work can, however, be readily extended to the case when $M > 1$. Therefore, from now on, the index of the pseudo data vector will be changed from i to k. Consequently, the pseudo data vector will be denoted by $\mathbf{Z}_{\text{pseudo}}(k)$, $k = 1, \cdots, N$, the difference in gradient and the Hessian estimate in (2), respectively, will simply be denoted by $\delta\mathbf{G}_k$ and $\hat{\mathbf{H}}_k$, $k = 1, \cdots, N$, and the notation of the one-sided gradient approximation in (3) will take the form of $\mathbf{G}^{(1)}(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}_k | \mathbf{Z}_{\text{pseudo}}(k))$. Finally, the following simplification of notation for the estimate of $\mathbf{F}_n(\boldsymbol{\theta})$ will also be used from now on,



Pseudo data Input
Hessian estimates $\hat{\mathbf{H}}_k^{(i)}$
Average of $\hat{\mathbf{H}}_k^{(i)} = \bar{\mathbf{H}}^{(i)}$
$\mathbf{Z}_{\text{pseudo}}(1)$
$\hat{\mathbf{H}}_1^{(1)}, \ldots\ldots \hat{\mathbf{H}}_M^{(1)}$
$\bar{\mathbf{H}}^{(1)}$
$\mathbf{Z}_{\text{pseudo}}(N)$
$\hat{\mathbf{H}}_1^{(N)}, \ldots\ldots \hat{\mathbf{H}}_M^{(N)}$
$\bar{\mathbf{H}}^{(N)}$
Negative average of $\bar{\mathbf{H}}^{(i)}$
FIM estimate, $\bar{\mathbf{F}}_{M,N}$

Fig. 2. Schematic model for forming estimate, $\bar{\mathbf{F}}_{M,N}$, of $\mathbf{F}_n(\boldsymbol{\theta})$; adapted from [2].

$$\hat{\mathbf{F}}_n \equiv \bar{\mathbf{F}}_{1,N}. \tag{4}$$

In the next section, the main idea of the current work with a brief highlight of the relevant theoretical basis is presented. The proposed scheme, that is similar in some sense to the one for Jacobian/Hessian estimates presented earlier [6], modifies and improves the current resampling algorithm by simultaneously preserving the known elements of the FIM and yielding better (in the sense of variance reduction) estimators of the unknown elements.

IV. Improved Resampling Algorithm — using Prior Information

Let the k-th estimate of Hessian matrix, $\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z}_{\text{pseudo}}(k))$, per proposed resampling algorithm be denoted by $\tilde{\mathbf{H}}_k$. In this section, the estimator, $\tilde{\mathbf{H}}_k$, is shown separately for two different cases: Case 1 :– when only the measurements of the log-likelihood function, L, are available and, Case 2 :– when the measurements of the exact gradient vector, $\mathbf{g}$, are available. To contrast the two cases, the superscript, (L), is used in $\tilde{\mathbf{H}}_k^{(L)}$ and $\hat{\mathbf{H}}_k^{(L)}$ to represent the dependence of $\tilde{\mathbf{H}}_k$ and $\hat{\mathbf{H}}_k$ on measurements of L for Case 1 and, the superscript, $(\mathbf{g})$, in $\tilde{\mathbf{H}}_k^{(\mathbf{g})}$ and $\hat{\mathbf{H}}_k^{(\mathbf{g})}$ for Case 2.

A. Additional Notation

Denote the (i,j)-th element of $\mathbf{F}_n(\boldsymbol{\theta})$ by $F_{ij}(\boldsymbol{\theta})$ (non-bold character and suppressing the subscript, n, in the symbolic notation representing the element of $\mathbf{F}_n(\boldsymbol{\theta})$) for simplification of notation. Let $\mathbb{I}_i$, $i = 1, \cdots, p$, be the set of column indices of the known elements of the i-th row of $\mathbf{F}_n(\boldsymbol{\theta})$ and $\mathbb{I}_i^c$ be the complement of $\mathbb{I}_i$. Consider a $p \times p$ matrix, $\mathbf{F}_n^{(\text{given})}$, whose (i,j)-th element, $F_{ij}^{(\text{given})}$, is defined as follows,

$$F_{ij}^{(\text{given})} = \begin{cases} F_{ij}(\boldsymbol{\theta}), & \text{if } j \in \mathbb{I}_i \\ 0, & \text{if } j \in \mathbb{I}_i^c \end{cases}, \quad i = 1, \cdots, p. \tag{5}$$

Consider another $p \times p$ matrix, $\boldsymbol{\mathcal{D}}_k$, defined by,

$$\boldsymbol{\mathcal{D}}_k = \boldsymbol{\Delta}_k \, [\Delta_{k1}^{-1}, \cdots, \Delta_{kp}^{-1}]. \tag{6}$$

together with a corresponding matrix, $\tilde{\boldsymbol{\mathcal{D}}}_k$, obtained by replacing all Δ_{ki} in $\boldsymbol{\mathcal{D}}_k$ with the corresponding $\tilde{\Delta}_{ki}$ (note that $\boldsymbol{\mathcal{D}}_k$ is symmetric when the perturbations are i.i.d. Bernoulli distributed).

B. The Step-by-Step Description of the Proposed Resampling Algorithm

The new estimate, $\tilde{\mathbf{H}}_k$, is extracted from $\tilde{\mathbf{H}}_{k0}$ that is defined below separately for Case 1 and Case 2.

Case 1: only the measurements of the log-likelihood function, L, are available,

$$\tilde{\mathbf{H}}_{k0}^{(L)} = \hat{\mathbf{H}}_k^{(L)} - \frac{1}{2}\left[\tilde{\boldsymbol{\mathcal{D}}}_k^T(-\mathbf{F}_n^{(\text{given})})\boldsymbol{\mathcal{D}}_k + (\tilde{\boldsymbol{\mathcal{D}}}_k^T(-\mathbf{F}_n^{(\text{given})})\boldsymbol{\mathcal{D}}_k)^T\right]. \tag{7}$$

Case 2: measurements of the exact gradient vector, $\mathbf{g}$, are available,

$$\tilde{\mathbf{H}}_{k0}^{(\mathbf{g})} = \hat{\mathbf{H}}_k^{(\mathbf{g})} - \frac{1}{2}\left[(-\mathbf{F}_n^{(\text{given})})\boldsymbol{\mathcal{D}}_k + ((-\mathbf{F}_n^{(\text{given})})\boldsymbol{\mathcal{D}}_k)^T\right]. \tag{8}$$

The estimates, $\tilde{\mathbf{H}}_k^{(L)}$ and $\tilde{\mathbf{H}}_k^{(\mathbf{g})}$, are readily obtained from, respectively, $\tilde{\mathbf{H}}_{k0}^{(L)}$ in (7) and $\tilde{\mathbf{H}}_{k0}^{(\mathbf{g})}$ in (8) by replacing the (i,j)-th element of $\tilde{\mathbf{H}}_{k0}^{(L)}$ and $\tilde{\mathbf{H}}_{k0}^{(\mathbf{g})}$ with known values of $-F_{ij}(\boldsymbol{\theta})$, $j \in \mathbb{I}_i$, $i = 1, \cdots, p$. The new estimate, $\tilde{\mathbf{F}}_n$, of $\mathbf{F}_n(\boldsymbol{\theta})$ is then computed by averaging the Hessian estimates $\tilde{\mathbf{H}}_k$, and taking negative value of the resulting average. For convenience and, also since the main objective is to estimate the FIM, the matrix, $\tilde{\mathbf{F}}_{n0}$, can be first obtained by computing the (negative) average of the matrices $\tilde{\mathbf{H}}_{k0}$, and subsequently, replacing the (i, j)-th element of $\tilde{\mathbf{F}}_{n0}$ with the analytically known elements, $F_{ij}(\boldsymbol{\theta}), j \in \mathbb{I}_i, i = 1, \cdots, p$, of $\mathbf{F}_n(\boldsymbol{\theta})$ would yield the new estimate, $\tilde{\mathbf{F}}_n$.

The matrices, $\hat{\mathbf{H}}_k^{(L)}$ in (7) or $\hat{\mathbf{H}}_k^{(\mathbf{g})}$ in (8), need to be computed by employing the existing resampling algorithm that is based on (2) and Fig. 2. Note that $\mathbf{F}_n^{(\text{given})}$ as shown in the right-hand-sides of (7) and (8) is known by (5). It must be noted as well that the random perturbation vectors, $\boldsymbol{\Delta}_k$ in $\boldsymbol{\mathcal{D}}_k$ and $\tilde{\boldsymbol{\Delta}}_k$ in $\tilde{\boldsymbol{\mathcal{D}}}_k$, as required in (7) and (8) must be the *same* simulated values of $\boldsymbol{\Delta}_k$ and $\tilde{\boldsymbol{\Delta}}_k$ used in the existing resampling algorithm while computing the k-th estimate, $\hat{\mathbf{H}}_k^{(L)}$ or $\hat{\mathbf{H}}_k^{(\mathbf{g})}$.

Next a summary of the salient steps, required to produce the estimate, $\tilde{\mathbf{F}}_n$ (*i.e.*, $\tilde{\mathbf{F}}_n^{(L)}$ or $\tilde{\mathbf{F}}_n^{(\mathbf{g})}$ with the appropriate superscript) of $\mathbf{F}_n(\boldsymbol{\theta})$ per modified resampling algorithm as proposed here, is presented below. Figure 3 is a schematic of the following steps.

Step 0. **Initialization:** Construct $\mathbf{F}_n^{(\text{given})}$ as defined by (5) based on the analytically known elements of the FIM. Determine $\boldsymbol{\theta}$, the sample size (n) and the number (N) of pseudo data vectors that will be generated. Determine whether log-likelihood, $L(\cdot)$, or gradient vector, $\mathbf{g}(\cdot)$, will be used to compute the Hessian estimates, $\hat{\mathbf{H}}_k$. Pick a small number, c, (perhaps $c = 0.0001$) to be used for Hessian estimation (see (2)) and, if required, another small number, $\tilde{c}$ (perhaps $\tilde{c} = 0.00011$), for gradient approximation (see (3)). Set $k = 1$.

Step 1. At the k-th step perform the following tasks,

- **a.** **Generation of pseudo data:** Based on $\boldsymbol{\theta}$, generate the k-th pseudo data vector, $\mathbf{Z}_{\text{pseudo}}(k)$, by using MC simulation technique.
- **b.** **Computation of $\hat{\mathbf{H}}_k$:** Generate $\boldsymbol{\Delta}_k$ (and also $\tilde{\boldsymbol{\Delta}}_k$, if required, for gradient approximation) by satisfying C.1. Using $\mathbf{Z}_{\text{pseudo}}(k)$, $\boldsymbol{\Delta}_k$ or/and $\tilde{\boldsymbol{\Delta}}_k$, evaluate $\hat{\mathbf{H}}_k$ (*i.e.*, $\hat{\mathbf{H}}_k^{(L)}$ or $\hat{\mathbf{H}}_k^{(\mathbf{g})}$) by using (2).
- **c.** **Computation of $\boldsymbol{\mathcal{D}}_k$ and $\tilde{\boldsymbol{\mathcal{D}}}_k$:** Use $\boldsymbol{\Delta}_k$ or/and $\tilde{\boldsymbol{\Delta}}_k$, as generated in the above step, to construct $\boldsymbol{\mathcal{D}}_k$ or/and $\tilde{\boldsymbol{\mathcal{D}}}_k$ as defined in section IV-A.
- **d.** **Computation of $\tilde{\mathbf{H}}_{k0}$:** Modify $\hat{\mathbf{H}}_k$ as produced in **Step 1b** by employing (7) or (8) as appropriate in order to generate $\tilde{\mathbf{H}}_{k0}$ (*i.e.*, $\tilde{\mathbf{H}}_{k0}^{(L)}$ or $\tilde{\mathbf{H}}_{k0}^{(\mathbf{g})}$).

Step 2. **Average of $\tilde{\mathbf{H}}_{k0}$:** Repeat **Step 1** until N estimates, $\tilde{\mathbf{H}}_{k0}$, are produced. Compute the (negative) mean of these N estimates. (The standard recursive representation of sample mean can be used here to avoid the storage of N matrices, $\tilde{\mathbf{H}}_{k0}$). The resulting (negative) mean is $\tilde{\mathbf{F}}_{n0}$.

Step 3. **Evaluation of $\tilde{\mathbf{F}}_n$:** The new estimate, $\tilde{\mathbf{F}}_n$, of $\mathbf{F}_n(\boldsymbol{\theta})$ per modified resampling algorithm is simply obtained by replacing the (i, j)-th element of $\tilde{\mathbf{F}}_{n0}$ with the analytically known elements, $F_{ij}(\boldsymbol{\theta})$, $j \in \mathbb{I}_i$, $i = 1, \cdots, p$, of $\mathbf{F}_n(\boldsymbol{\theta})$. To avoid the possibility of having a non-positive semi-definite estimate, it may be desirable to take the symmetric square root of the square of the estimate (the `sqrtm` function in MATLAB may be useful here).

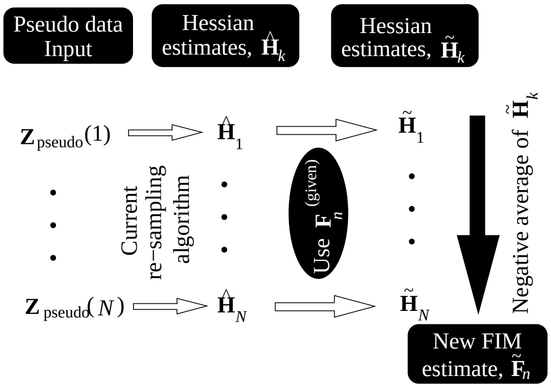

Pseudo data Input
Hessian estimates, $\hat{\mathbf{H}}_k$
Hessian estimates, $\tilde{\mathbf{H}}_k$
$\mathbf{Z}_{\text{pseudo}}(1)$
$\hat{\mathbf{H}}_1$
$\tilde{\mathbf{H}}_1$
Current re−sampling algorithm
Use $\mathbf{F}_n^{(\text{given})}$
Negative average of $\tilde{\mathbf{H}}_k$
$\mathbf{Z}_{\text{pseudo}}(N)$
$\hat{\mathbf{H}}_N$
$\tilde{\mathbf{H}}_N$
New FIM estimate, $\tilde{\mathbf{F}}_n$

Fig. 3. Schematic of algorithm for computing the new FIM estimate, $\tilde{\mathbf{F}}_n$.

The new estimator, $\tilde{\mathbf{F}}_n$, is better than $\hat{\mathbf{F}}_n$ in the sense that it would preserve exactly the analytically known elements of $\mathbf{F}_n(\boldsymbol{\theta})$ as well as reduce the variances of the estimators of the unknown elements of $\mathbf{F}_n(\boldsymbol{\theta})$.

C. Theoretical Basis for the Modified Resampling Algorithm

For notational simplification, the subscript 'pseudo' in $\mathbf{Z}_{\text{pseudo}}(k)$ and the dependence of $\mathbf{Z}(k)$ on k would be suppressed (note that $\mathbf{Z}_{\text{pseudo}}(k)$ is identically distributed across k). Since, $\boldsymbol{\Delta}_k$ is usually assumed to be statistically independent across k and an identical condition for $\tilde{\boldsymbol{\Delta}}_k$ is also assumed, their dependence on k would also be suppressed in the forthcoming discussion. Let also the (i, j)-th element of $\hat{\mathbf{H}}_k$ and $\tilde{\mathbf{H}}_k$ be, respectively, denoted by $\hat{H}_{ij}$ and $\tilde{H}_{ij}$ with the appropriate superscript. The two cases as described earlier by (7) and (8) are considered next in the following two separate subsections.

1) Case 1 - only the measurements of L are available: The main objective here is to compare variance of $\hat{H}_{ij}^{(L)}$ and variance of $\tilde{H}_{ij}^{(L)}$ to show the superiority of $\tilde{\mathbf{H}}_k^{(L)}$, which leads to the superiority of $\tilde{\mathbf{F}}_n^{(L)}$.

It is assumed here that the gradient estimate is based on one-sided gradient approximation using SP technique given by (3). Based on a Taylor expansion, the i-th component of $\mathbf{G}^{(1)}(\boldsymbol{\theta}|\mathbf{Z})$, $i = 1, \cdots, p$, that is an approximation of the i-th component, $g_i(\boldsymbol{\theta}|\mathbf{Z}) \equiv \partial L(\boldsymbol{\theta}|\mathbf{Z})/\partial\theta_i$, of $\mathbf{g}(\boldsymbol{\theta}|\mathbf{Z})$ based on the values of $L(\cdot|\mathbf{Z})$, can be readily shown to given by,

$$\begin{aligned} G_i^{(1)}(\boldsymbol{\theta}|\mathbf{Z}) &= \frac{L(\boldsymbol{\theta} + \tilde{c}\tilde{\boldsymbol{\Delta}}|\mathbf{Z}) - L(\boldsymbol{\theta}|\mathbf{Z})}{\tilde{c}\,\tilde{\Delta}_i} && (9) \\ &= \sum_l g_l(\boldsymbol{\theta}) \frac{\tilde{\Delta}_l}{\tilde{\Delta}_i} + \frac{1}{2}\tilde{c} \sum_{l,m} H_{lm}(\boldsymbol{\theta}) \frac{\tilde{\Delta}_m \tilde{\Delta}_l}{\tilde{\Delta}_i} \\ &\quad + \frac{1}{6}\tilde{c}^2 \sum_{l,m,s} \frac{\partial H_{lm}(\overline{\boldsymbol{\theta}})}{\partial \theta_s} \frac{\tilde{\Delta}_s \tilde{\Delta}_m \tilde{\Delta}_l}{\tilde{\Delta}_i}, && (10) \end{aligned}$$

in which $H_{lm}(\boldsymbol{\theta}|\mathbf{Z}) \equiv \partial^2 L(\boldsymbol{\theta}|\mathbf{Z})/\partial\theta_l\,\partial\theta_m$ is the (l, m)-th element of $\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z})$, $\overline{\boldsymbol{\theta}} = \lambda(\boldsymbol{\theta} + \tilde{c}\tilde{\boldsymbol{\Delta}}) + (1 - \lambda)\boldsymbol{\theta} = \boldsymbol{\theta} + \tilde{c}\lambda\tilde{\boldsymbol{\Delta}}$

(with $\lambda \in [0,\, 1]$ being some real number) denotes a point on the line segment between $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + \tilde{c}\tilde{\boldsymbol{\Delta}}$ and, in the expression after the last equality, the condition on $\mathbf{Z}$ is suppressed for notational clarity and, also the summations are expressed in abbreviated format where the indices span their respective and appropriate ranges.

Given $G_i(\cdot|\mathbf{Z}) \equiv G_i^{(1)}(\cdot|\mathbf{Z})$ by (10), the (i,j)-th element of $\hat{\mathbf{H}}_k^{(L)}$ can be readily obtained from,

$$\hat{H}_{ij}^{(L)} = \frac{1}{2}\left[\hat{J}_{ij}^{(L)} + \hat{J}_{ji}^{(L)}\right], \qquad (11)$$

in which $\hat{J}_{ij}^{(L)}$ (J is to indicate Jacobian for which the symmetrizing operation should not be used) and its expression based on Taylor expansions of the associated terms, $G_i(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}|\mathbf{Z})$, are shown below (A third order Taylor expansion is applied on the first group of summand of $G_i(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}|\mathbf{Z}) \equiv G_i^{(1)}(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}|\mathbf{Z})$, that is obtained by replacing $\boldsymbol{\theta}$ with $\boldsymbol{\theta} \pm c\boldsymbol{\Delta}$ in (10), and first order Taylor expansions are applied on the second and third group of summand.),

$$\begin{aligned}\hat{J}_{ij}^{(L)} &\equiv \frac{G_i(\boldsymbol{\theta} + c\boldsymbol{\Delta}|\mathbf{Z}) - G_i(\boldsymbol{\theta} - c\boldsymbol{\Delta}|\mathbf{Z})}{2\,c\,\Delta_j} \\ &= \sum_{l,m} H_{lm}(\boldsymbol{\theta}|\mathbf{Z})\frac{\Delta_m}{\Delta_j}\frac{\tilde{\Delta}_l}{\tilde{\Delta}_i} + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(c^2) \\ &\quad + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}) + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}^2). \qquad (12)\end{aligned}$$

The subscripts in the 'big-O' terms, $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\cdot)$, in the above equation explicitly indicate that they depend on $\tilde{\boldsymbol{\Delta}}$, $\boldsymbol{\Delta}$ and $\mathbf{Z}$. In these *random* 'big-O' terms, the point of evaluation, $\boldsymbol{\theta}$, is suppressed for notational clarity. By the use of C.1 and further assumptions on the continuity and uniformly (in k) boundedness conditions on all the derivatives (up to fourth order) of L, it can be shown that $|O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(c^2)/c^2| < \infty$ almost surely (a.s.) (a.s. with respect to the joint probability measure of $\tilde{\boldsymbol{\Delta}}$, $\boldsymbol{\Delta}$ and $\mathbf{Z}$) as $c \longrightarrow 0$ and, both $|O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c})/\tilde{c}| < \infty$ a.s. and $|O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}^2)/\tilde{c}^2| < \infty$ a.s. as $\tilde{c} \longrightarrow 0$. The effects of $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}^2)$ are not included in $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c})$. The reason for showing $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c})$ separately in (12) is that this term vanishes upon expectation because it involves either $E[\tilde{\Delta}_r]$ or $E[1/\tilde{\Delta}_r]$, $r = 1, \cdots, p$, both of which are zero by implication I and rest of the terms appeared in $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c})$ do not depend on $\tilde{\boldsymbol{\Delta}}$. The other terms, $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(c^2)$ and $O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}^2)$, do not vanish upon expectation. Subsequently, by noting the fact that $\tilde{\boldsymbol{\Delta}}$, $\boldsymbol{\Delta}$ and $\mathbf{Z}$ are statistically independent of each other and by using the condition C.1, it can be readily shown that,

$$E[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}] = E\left[H_{ij}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}\right] + O(c^2) + O(\tilde{c}^2). \qquad (13)$$

Note that the 'big-O' terms, $O(c^2)$ and $O(\tilde{c}^2)$, satisfying $|O(c^2)/c^2| < \infty$ as $c \longrightarrow 0$ and $|O(\tilde{c}^2)/\tilde{c}^2| < \infty$ as $\tilde{c} \longrightarrow 0$, are *deterministic* unlike the random 'big-O' terms in (12). In the context of FIM, $E\left[H_{ij}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}\right] = -F_{ij}(\boldsymbol{\theta})$ by (Hessian-based) definition using which (along with the symmetry of the FIM, $\mathbf{F}_n(\boldsymbol{\theta})$) $E[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$ follows straight from (11)-(13) as below,

$$E[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}] = -F_{ij}(\boldsymbol{\theta}) + O(c^2) + O(\tilde{c}^2). \qquad (14)$$

The variance of $\hat{H}_{ij}^{(L)}$ is to be computed next. It is given by,

$$\text{var}[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}] = \frac{1}{4}\left(\text{var}[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}] + \text{var}[\hat{J}_{ji}^{(L)}|\boldsymbol{\theta}] + 2\text{cov}[\hat{J}_{ij}^{(L)}, \hat{J}_{ji}^{(L)}|\boldsymbol{\theta}]\right). \qquad (15)$$

The expression of a typical variance term, $\text{var}[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}]$, in (15) would now be determined followed by the deduction of the expression of covariance term, $\text{cov}[\hat{J}_{ij}^{(L)}, \hat{J}_{ji}^{(L)}|\boldsymbol{\theta}]$.

By the use of (12), it can be shown after some simplification that,

$$\begin{aligned}\text{var}[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}] &= \sum_{l,m} a_{lm}(i,j)\text{var}\left[H_{lm}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}\right] \\ &\quad + \sum_{\substack{l,m \\ lm \neq ij}} a_{lm}(i,j)\left(E\left[H_{lm}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}\right]\right)^2 \\ &\quad + O(c^2) + O(\tilde{c}^2) + O(c^2\tilde{c}^2), \qquad (16)\end{aligned}$$

in which $a_{lm}(i,j) = E[\Delta_m^2/\Delta_j^2]E[\tilde{\Delta}_l^2/\tilde{\Delta}_i^2]$. The expression in (16) is essentially derived by using the mutual independence of $\tilde{\boldsymbol{\Delta}}$, $\boldsymbol{\Delta}$ and $\mathbf{Z}$, the condition C.1 and the implication I. In addition, it is also assumed that all the combinations of covariance terms involving $H_{lm}(\boldsymbol{\theta}|\mathbf{Z})$, $H_{lm,s}(\boldsymbol{\theta}|\mathbf{Z})$ and $H_{lm,rs}(\boldsymbol{\theta}|\mathbf{Z})$, $l, m, s, r = 1, \cdots, p$, exist *around* $\boldsymbol{\theta}$ that indicates the point of evaluation of these functions.

Next, the expression of $\text{cov}[\hat{J}_{ij}^{(L)}, \hat{J}_{ji}^{(L)}|\boldsymbol{\theta}]$, $j \neq i$, can be deduced by using identical arguments to the ones that are used in deriving the expression of $\text{var}[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}]$ above. Further simplification based on the use of mutual statistical independence among $\tilde{\boldsymbol{\Delta}}$, $\boldsymbol{\Delta}$ and $\mathbf{Z}$ and, also on the Hessian-based definition and symmetry of FIM as well as on condition C.1 and implication I yields the following, $j \neq i$,

$$\begin{aligned}&\text{cov}[\hat{J}_{ij}^{(L)}, \hat{J}_{ji}^{(L)}|\boldsymbol{\theta}] \\ &= 2\left\{\text{var}\left[(H_{ij}(\boldsymbol{\theta}|\mathbf{Z}))|\,\boldsymbol{\theta}\right] + \left(E\left[(H_{ij}(\boldsymbol{\theta}|\mathbf{Z}))|\,\boldsymbol{\theta}\right]\right)^2\right\} \\ &+2E[H_{ii}(\boldsymbol{\theta}|\mathbf{Z})H_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] - (F_{ij}(\boldsymbol{\theta}))^2 + O(c^2) + O(\tilde{c}^2) + O(c^2\tilde{c}^2). \quad (17)\end{aligned}$$

Now, the variance of $\hat{H}_{ij}^{(L)}$, $\text{var}[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$, for $j \neq i$, can be readily obtained from (15) by using (16) and (17). Note that $\text{var}[\hat{H}_{ii}^{(L)}|\boldsymbol{\theta}]$ is same as $\text{var}[\hat{J}_{ii}^{(L)}|\boldsymbol{\theta}]$ that can be directly obtained from (16) by replacing j with i. The contributions of the variance and covariance terms (as appeared in (15)) to $\text{var}[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$ are compared next with the contributions of the respective variance and covariance terms to $\text{var}[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}]$.

Consider the (i,j)-th element of $\tilde{\mathbf{H}}_k$ associated with (7) that is given by,

$$\tilde{H}_{ij}^{(L)} = \frac{1}{2}\left(\tilde{J}_{ij}^{(L)} + \tilde{J}_{ji}^{(L)}\right), \ \forall j \in \mathbb{I}_i^c, \qquad (18)$$

and

$$\tilde{H}_{ij}^{(L)} = -F_{ij}(\boldsymbol{\theta}), \ \forall j \in \mathbb{I}_i. \qquad (19)$$

In (18), $\tilde{J}_{ij}^{(L)}$ is defined as,

$$\tilde{J}_{ij}^{(L)} = \hat{J}_{ij}^{(L)} - \sum_{l}\sum_{m \in \mathbb{I}_l}(-F_{lm}(\boldsymbol{\theta}))\frac{\Delta_m}{\Delta_j}\frac{\tilde{\Delta}_l}{\tilde{\Delta}_i}, \quad \forall j \in \mathbb{I}_i^c. \quad (20)$$

Note that $E[\sum_l \sum_{m\in\mathbb{I}_l}(-F_{lm}(\boldsymbol{\theta}))(\Delta_m/\Delta_j)(\tilde{\Delta}_l/\tilde{\Delta}_i)|\boldsymbol{\theta}] = 0$ in (20), $\forall j \in \mathbb{I}_i^c$, implying that $E[\tilde{J}_{ij}^{(L)}] = E[\hat{J}_{ij}^{(L)}]$, $\forall j \in \mathbb{I}_i^c$. By using this fact along with identical arguments, that are used in deducing (14), immediately results in, $\forall i = 1, \cdots, p$,

$$E[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}] = \begin{cases} -F_{ij}(\boldsymbol{\theta}) + O(c^2) + O(\tilde{c}^2), \forall j \in \mathbb{I}_i^c, \\ -F_{ij}(\boldsymbol{\theta}), \qquad\qquad\qquad\qquad \forall j \in \mathbb{I}_i. \end{cases} \quad (21)$$

Noticeably, the expressions of the 'big-O' terms both in (14) and in the first equation of (21) are precisely same implying that $E[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}] = E[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$, $\forall j \in \mathbb{I}_i^c$.

While var$[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}] = 0$, $\forall j \in \mathbb{I}_i$, by (19) clearly implying that var$[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}] <$ var$[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$, $\forall j \in \mathbb{I}_i$, the deduction of expression of var$[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}]$, $\forall j \in \mathbb{I}_i^c$, is the task that will be considered now. In fact, this is the main result associated with the variance reduction from prior information available in terms of the known elements of $\mathbf{F}_n(\boldsymbol{\theta})$.

The first step in determining this variance is to note that the expression of $\hat{J}_{ij}^{(L)}$ in (12) can be decomposed into two parts as shown below,

$$\begin{aligned}\hat{J}_{ij}^{(L)} &= \sum_l \left[\sum_{m\in\mathbb{I}_l} H_{lm}(\boldsymbol{\theta}|\mathbf{Z}) \frac{\Delta_m}{\Delta_j} \frac{\tilde{\Delta}_l}{\tilde{\Delta}_i} + \sum_{m\in\mathbb{I}_l^c} H_{lm}(\boldsymbol{\theta}|\mathbf{Z}) \frac{\Delta_m}{\Delta_j} \frac{\tilde{\Delta}_l}{\tilde{\Delta}_i} \right] \\ &\quad + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(c^2) + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}) + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}^2). \quad (22)\end{aligned}$$

The elements, $H_{lm}(\boldsymbol{\theta}|\mathbf{Z})$, of $\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z})$ in the right-hand-side of (22) are not known. However, since by (Hessian-based) definition $E[H_{lm}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] = -F_{lm}(\boldsymbol{\theta})$, approximation of the unknown elements of $\mathbf{H}(\boldsymbol{\theta}|\mathbf{Z})$ in the right-hand-side of (22), particularly those elements that correspond to the elements of the FIM that are known *a priori*, by the negative of those elements of $\mathbf{F}_n(\boldsymbol{\theta})$ is the primary idea based on which the modified resampling algorithm is developed. This approximation introduces an error term, $e_{lm}(\boldsymbol{\theta}|\mathbf{Z})$, that can be defined by, $\forall m \in \mathbb{I}_l$, $l = 1, \cdots, p$,

$$H_{lm}(\boldsymbol{\theta}|\mathbf{Z}) = -F_{lm}(\boldsymbol{\theta}) + e_{lm}(\boldsymbol{\theta}|\mathbf{Z}), \quad (23)$$

and this error term satisfies the following two conditions that directly follow from (23), $\forall m \in \mathbb{I}_l$, $l = 1, \cdots, p$,

$$E[e_{lm}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] = 0, \quad (24)$$

$$\text{var}[e_{lm}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] = \text{var}[H_{lm}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}]. \quad (25)$$

Also, introduce X_{lm}, $l = 1, \cdots, p$, as defined below,

$$X_{lm}(\boldsymbol{\theta}|\mathbf{Z}) = \begin{cases} e_{lm}(\boldsymbol{\theta}|\mathbf{Z}), & \text{if } m \in \mathbb{I}_l, \\ H_{lm}(\boldsymbol{\theta}|\mathbf{Z}), & \text{if } m \in \mathbb{I}_l^c. \end{cases} \quad (26)$$

Now, substitution of (23) in (22) results in a known part in the right-hand-side of (22) involving the analytically known elements of FIM. This known part is transferred to the left-hand-side of (22) and, consequently, acts as a feedback to the current resampling algorithm yielding, in the process, an expression of $\tilde{J}_{ij}^{(L)}$. By making use of (26), it can be shown that $\tilde{J}_{ij}^{(L)}$ can be compactly written as,

$$\begin{aligned}\tilde{J}_{ij}^{(L)} &= \sum_{l,m} X_{lm}(\boldsymbol{\theta}|\mathbf{Z}) \frac{\Delta_m}{\Delta_j} \frac{\tilde{\Delta}_l}{\tilde{\Delta}_i} + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(c^2) + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}) \\ &\quad + O_{\tilde{\boldsymbol{\Delta}},\boldsymbol{\Delta},\mathbf{Z}}(\tilde{c}^2), \quad \forall j \in \mathbb{I}_i^c. \quad (27)\end{aligned}$$

The variance of $\tilde{J}_{ij}^{(L)}$, $\forall j \in \mathbb{I}_i^c$, can be computed by considering the right-hand-side of (27) in an identical way as described earlier for $\hat{J}_{ij}^{(L)}$. The expression for var$[\tilde{J}_{ij}^{(L)}|\boldsymbol{\theta}]$, $\forall j \in \mathbb{I}_i^c$, follows readily from (16) by replacing H_{lm} with X_{lm} because of the similarity between (12) and (27). Use of (26) first and subsequent uses of (24)-(25) on the resulting expression finally yields an expression of var$[\tilde{J}_{ij}^{(L)}|\boldsymbol{\theta}]$. Subtracting the finally yielded expression of var$[\tilde{J}_{ij}^{(L)}|\boldsymbol{\theta}]$ from (16), it can be shown by having recourse to the Hessian-based definition of FIM, $E[H_{lm}(\boldsymbol{\theta}|\mathbf{Z})\,|\boldsymbol{\theta}] = -F_{lm}(\boldsymbol{\theta})$, that, $\forall j \in \mathbb{I}_i^c$, $i = 1, \cdots, p$,

$$\begin{aligned}\text{var}[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}] - \text{var}[\tilde{J}_{ij}^{(L)}|\boldsymbol{\theta}] &= \sum_l \sum_{m\in\mathbb{I}_l} a_{lm}(i,j)\,(F_{lm}(\boldsymbol{\theta}))^2 \\ &\quad + O(c^2) + O(\tilde{c}^2) + O(c^2\tilde{c}^2) \; > \; 0. \quad (28)\end{aligned}$$

The inequality above follows from the fact that $a_{lm}(i,j) = (E[\Delta_m^2/\Delta_j^2]\ E[\tilde{\Delta}_l^2/\tilde{\Delta}_i^2]) > 0$, $l, m = 1, \cdots, p$, for any given (i,j) and assuming that at least one of the known elements, $F_{lm}(\boldsymbol{\theta})$, in (28) is not equal to zero. It must be remarked that the bias terms, $O(c^2)$, $O(\tilde{c}^2)$ and $O(c^2\tilde{c}^2)$, can be made negligibly small by selecting c and $\tilde{c}$ small enough that are primarily controlled by users. Note that if $\Delta_1, \cdots, \Delta_p$ and $\tilde{\Delta}_1, \cdots, \tilde{\Delta}_p$ are both assumed to be Bernoulli ± 1 i.i.d. random variables, then $a_{lm}(i,j)$ turns out to be unity.

At this point it should be already clear that var$[\tilde{H}_{ii}^{(L)}|\boldsymbol{\theta}] <$ var$[\hat{H}_{ii}^{(L)}|\boldsymbol{\theta}]$, if $j = i \in \mathbb{I}_i^c$, by (28).

Next step is to compare cov$[\tilde{J}_{ij}^{(L)}, \tilde{J}_{ji}^{(L)}|\boldsymbol{\theta}]$ with cov$[\hat{J}_{ij}^{(L)}, \hat{J}_{ji}^{(L)}|\boldsymbol{\theta}]$, $j \neq i$, $\forall j \in \mathbb{I}_i^c$, in order to conclude that var$[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}] <$ var$[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$. As var$[\tilde{J}_{ij}^{(L)}|\boldsymbol{\theta}]$ is deduced from var$[\hat{J}_{ij}^{(L)}|\boldsymbol{\theta}]$ by the similarity of the expressions between (12) and (27) along with other arguments, following identical line of arguments, the expression of cov$[\tilde{J}_{ij}^{(L)}, \tilde{J}_{ji}^{(L)}|\boldsymbol{\theta}]$ can be deduced and, finally, by keeping in mind that $j \in \mathbb{I}_i^c$ and by using (26), it can be shown that,

$$\begin{aligned}&\text{cov}[\hat{J}_{ij}^{(L)}, \hat{J}_{ji}^{(L)}|\boldsymbol{\theta}] - \text{cov}[\tilde{J}_{ij}^{(L)}, \tilde{J}_{ji}^{(L)}|\boldsymbol{\theta}] \\ &= 2(E\,[\,H_{ii}(\boldsymbol{\theta}|\mathbf{Z})H_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}] - E\,[\,X_{ii}(\boldsymbol{\theta}|\mathbf{Z})X_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}]) \\ &\quad + O(c^2) + O(\tilde{c}^2) + O(c^2\tilde{c}^2), \; j \neq i, \forall j \in \mathbb{I}_i^c. \quad (29)\end{aligned}$$

Note that X_{ii} and X_{jj} must take the form from one of following four possibilities: (1) e_{ii} and e_{jj}, (2) e_{ii} and H_{jj}, (3) H_{ii} and e_{jj}, (4) H_{ii} and H_{jj}. While $E[H_{ii}(\boldsymbol{\theta}|\mathbf{Z})H_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] - E[X_{ii}(\boldsymbol{\theta}|\mathbf{Z})\ X_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}]$ is 0 for the fourth possibility, it can be shown by using (23) and the Hessian-based definition of FIM that for the other possibilities, this difference is given by $F_{ii}(\boldsymbol{\theta})F_{jj}(\boldsymbol{\theta})$ that is greater than 0 since $\mathbf{F}_n(\boldsymbol{\theta})$ is positive definite matrix.

Therefore, using (28) and the fact that $E[H_{ii}(\boldsymbol{\theta}|\mathbf{Z})H_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] - E[X_{ii}(\boldsymbol{\theta}|\mathbf{Z})\ X_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] \geq 0$, that appeared in (29), and also noting that the bias terms, $O(c^2)$, $O(\tilde{c}^2)$ and $O(c^2\tilde{c}^2)$, can be made negligibly small by selecting c and $\tilde{c}$ small enough, the following can be concluded immediately from (15) and an identical expression of $\mathrm{var}[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}]$,

$$\mathrm{var}[\tilde{H}_{ij}^{(L)}|\boldsymbol{\theta}] < \mathrm{var}[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}], \quad i,j = 1,\cdots,p.$$

In deducing the expressions of mean and variance, several assumptions related to the existences of derivatives of L with respect to $\boldsymbol{\theta}$ and also the existences of expectations of these derivatives are required as 'hinted' earlier sporadically. For a complete list of assumptions and rigorous derivation of these expressions, readers are referred to [7].

Since Case 2 is simpler than Case 1 that has already been considered in full possible detail within the limited space, the next section simply presents the final and important points for Case 2 that highlight the fact that the variance of $\tilde{H}_{ij}^{(\mathbf{g})}$ is less than the variance of $\hat{H}_{ij}^{(\mathbf{g})}$.

2) Case 2 *- measurements of* $\mathbf{g}$ *are available:* In this section, it is assumed that the measurements of exact gradient vector, $\mathbf{g}$, are available. The (i,j)-th element, $\hat{H}_{ij}^{(\mathbf{g})}$, (the dependence on k is suppressed) of $\hat{\mathbf{H}}_k^{(\mathbf{g})}$ is then given by,

$$\hat{H}_{ij}^{(\mathbf{g})} = \frac{1}{2}\left[\hat{J}_{ij}^{(\mathbf{g})} + \hat{J}_{ji}^{(\mathbf{g})}\right], \tag{30}$$

in which $\hat{J}_{ij}^{(\mathbf{g})}$ and its expression based on third-order Taylor expansion of the associated terms, $g_i(\boldsymbol{\theta} \pm c\boldsymbol{\Delta}|\mathbf{Z})$, are shown below,

$$\hat{J}_{ij}^{(\mathbf{g})} \equiv \frac{g_i(\boldsymbol{\theta}+c\boldsymbol{\Delta}|\mathbf{Z}) - g_i(\boldsymbol{\theta}-c\boldsymbol{\Delta}|\mathbf{Z})}{2\,c\,\Delta_j} = \sum_l H_{il}(\boldsymbol{\theta}|\mathbf{Z})\frac{\Delta_l}{\Delta_j} + O_{\boldsymbol{\Delta},\mathbf{Z}}(c^2). \tag{31}$$

The expectation of $\hat{H}_{ij}^{(\mathbf{g})}$ follows straight from (31) by carefully using the identical arguments as described in previous subsection for $E[\hat{H}_{ij}^{(L)}|\boldsymbol{\theta}]$,

$$E[\hat{H}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}] = -F_{ij}(\boldsymbol{\theta}) + O(c^2). \tag{32}$$

On the other hand, the (i,j)-th element of $\tilde{\mathbf{H}}_k$ associated with (8) is given by,

$$\tilde{H}_{ij}^{(\mathbf{g})} = \begin{cases} \frac{1}{2}(\tilde{J}_{ij}^{(\mathbf{g})} + \tilde{J}_{ji}^{(\mathbf{g})}), & \forall j \in \mathbb{I}_i^c, \\ -F_{ij}(\boldsymbol{\theta}), & \forall j \in \mathbb{I}_i, \end{cases}$$

with $\tilde{J}_{ij}^{(\mathbf{g})}$ being given by,

$$\tilde{J}_{ij}^{(\mathbf{g})} = \hat{J}_{ij}^{(\mathbf{g})} - \sum_{l \in \mathbb{I}_i}(-F_{il}(\boldsymbol{\theta}))\frac{\Delta_l}{\Delta_j}, \quad \forall j \in \mathbb{I}_i^c. \tag{33}$$

As shown for $\tilde{H}_{ij}^{(L)}$ earlier in Case 1, it can shown in an identical way that $\forall i = 1,\cdots,p$,

$$E[\tilde{H}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}] = \begin{cases} -F_{ij}(\boldsymbol{\theta}) + O(c^2), & \forall j \in \mathbb{I}_i^c, \\ -F_{ij}(\boldsymbol{\theta}), & \forall j \in \mathbb{I}_i. \end{cases} \tag{34}$$

Again, the expressions of the 'big-O' terms both in (32) and in the first equation of (34) are precisely same implying that $E[\tilde{H}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}] = E[\hat{H}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}]$, $\forall j \in \mathbb{I}_i^c$.

Next, the difference between $\mathrm{var}[\hat{H}_{ii}^{(\mathbf{g})}|\boldsymbol{\theta}]$ and $\mathrm{var}[\tilde{H}_{ii}^{(\mathbf{g})}|\boldsymbol{\theta}]$ can be deduced by carefully following the identical arguments as used for Case 1 in the previous subsection and this difference can be shown to be given by $\forall j \in \mathbb{I}_i^c$,

$$\mathrm{var}[\hat{J}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}] - \mathrm{var}[\tilde{J}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}] = \sum_{l \in \mathbb{I}_i} b_l(j)\,(F_{il}(\boldsymbol{\theta}))^2 + O(c^2) > 0. \tag{35}$$

Here, $b_l(j) = E[\Delta_l^2/\Delta_j^2] > 0$, $l = 1,\cdots,p$, and it turns out to be unity if $\Delta_1,\cdots,\Delta_p$ is assumed to be Bernoulli ± 1 i.i.d. random variables and, as always, the bias term, $O(c^2)$, can be made negligibly small by selecting the user-controlled variable c small enough.

The difference between the contributing covariance terms can also be shown to be given by,

$$\mathrm{cov}[\hat{J}_{ij}^{(\mathbf{g})},\hat{J}_{ji}^{(\mathbf{g})}|\boldsymbol{\theta}] - \mathrm{cov}[\tilde{J}_{ij}^{(\mathbf{g})},\tilde{J}_{ji}^{(\mathbf{g})}|\boldsymbol{\theta}] = E[H_{ii}(\boldsymbol{\theta}|\mathbf{Z})H_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\boldsymbol{\theta}] - E\left[X_{ii}(\boldsymbol{\theta}|\mathbf{Z})X_{jj}(\boldsymbol{\theta}|\mathbf{Z})|\,\boldsymbol{\theta}\right] + O(c^2),\ j \neq i, \forall j \in \mathbb{I}_i^c. \tag{36}$$

Therefore, it can be immediately concluded by using identical arguments as used for Case 1 that

$$\mathrm{var}[\tilde{H}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}] < \mathrm{var}[\hat{H}_{ij}^{(\mathbf{g})}|\boldsymbol{\theta}], \quad i,j = 1,\cdots,p.$$

Finally, since $\hat{\mathbf{H}}_k$, $k = 1,\cdots,N$, are statistically independent of each other and $\tilde{\mathbf{H}}_k$, $k = 1,\cdots,N$, are also statistically independent of each other, it can be concluded straightway that (i,j)-th element of $\hat{\mathbf{F}}_n = -(1/N)\sum_{k=1}^N \hat{\mathbf{H}}_k$ and $\tilde{\mathbf{F}}_n = -(1/N)\sum_{k=1}^N \tilde{\mathbf{H}}_k$ (with appropriate superscript, (L) or $(\mathbf{g})$) satisfy the following relation, $i,j = 1,\cdots,p$,

$$\mathrm{var}[\tilde{F}_{ij}|\boldsymbol{\theta}] = \frac{\mathrm{var}[\tilde{H}_{ij}|\boldsymbol{\theta}]}{N} < \mathrm{var}[\hat{F}_{ij}|\boldsymbol{\theta}] = \frac{\mathrm{var}[\hat{H}_{ij}|\boldsymbol{\theta}]}{N}. \tag{37}$$

Therefore, we conclude this section by stating that the better estimator, $\tilde{\mathbf{F}}_n$, (*vis-à-vis* the current estimator, $\hat{\mathbf{F}}_n$) as determined by using the modified resampling algorithm would preserve the exact analytically known elements of $\mathbf{F}_n(\boldsymbol{\theta})$ as well as reduce the variances of the estimators of the unknown elements of $\mathbf{F}_n(\boldsymbol{\theta})$.

Next section presents an example illustrating the effectiveness of the modified resampling algorithm.

V. Numerical Illustrations and Discussions

Consider independently distributed scalar-valued random data $\mathbf{z}_i$ with $\mathbf{z}_i \sim N(\mu, \sigma^2 + c_i\alpha)$, $i = 1,\cdots,n$, in which μ and $(\sigma^2 + c_i\alpha)$ are, respectively, mean and variance of $\mathbf{z}_i$ with c_i being some known nonnegative constants and $\alpha > 0$. Here, $\boldsymbol{\theta}$ is considered as $\boldsymbol{\theta} = [\mu, \sigma^2, \alpha]^T$. This is a simple extension of an example problem already considered in literature [1, Example 13.7]. The analytical FIM, $\mathbf{F}_n(\boldsymbol{\theta})$, can be readily determined for this case so that the MC resampling-based estimates of $\mathbf{F}_n(\boldsymbol{\theta})$ can be verified with the analytical FIM. It can be shown that the analytical FIM is given by,

$$\mathbf{F}_n(\boldsymbol{\theta}) = \begin{bmatrix} F_{11} & 0 & 0 \\ 0 & F_{22} & F_{33} \\ 0 & F_{33} & F_{33} \end{bmatrix},$$

in which $F_{11} = \sum_{i=1}^{n}(\sigma^2 + c_i\alpha)^{-1}$, $F_{22} = (1/2)\sum_{i=1}^{n}(\sigma^2 + c_i\alpha)^{-2}$ and $F_{33} = (1/2)\sum_{i=1}^{n} c_i\,(\sigma^2{+}c_i\alpha)^{-2}$. Here, the value of $\boldsymbol{\theta}$, that is used to generate the pseudo data vector (as a proxy for $\mathbf{Z}_n = [\mathbf{z}_1, \cdots, \mathbf{z}_n]^T$) and to evaluate $\mathbf{F}_n(\boldsymbol{\theta})$, is assumed to correspond to $\mu = 0$, $\sigma^2 = 1$ and $\alpha = 1$. The values of c_i across i are chosen between 0 and 1, which are generated by using MATLAB uniform random number generator, `rand`, with a given seed (`rand('state',0)`). Based on $n = 30$ yields a positive definite $\mathbf{F}_n(\boldsymbol{\theta})$ whose eigenvalues are given by 0.5696, 8.6925 and 20.7496.

To illustrate the effectiveness of the modified MC based resampling algorithm, it is assumed here that only the upper-left 2×2 block of the analytical FIM is known *a priori*. Using this known information, both the existing [2] and the modified resampling algorithm as proposed in this work are employed to estimate the FIM. For Hessian estimation per (2), c is considered as 0.0001 and, for gradient-approximation per (3), $\tilde{c}$ is considered as 0.00011. Bernoulli ±1 random variable components are considered to generate both the perturbation vectors, $\boldsymbol{\Delta}_k$ and $\tilde{\boldsymbol{\Delta}}_k$.

The results are summarized in Table I. The mean-squared error (MSE) of $\hat{\mathbf{F}}_n$ and $\tilde{\mathbf{F}}_n$ are first computed; for example, in the case of $\hat{\mathbf{F}}_n$, $\text{MSE}(\hat{\mathbf{F}}_n)$ is computed as $\text{MSE}(\hat{\mathbf{F}}_n) = \sum_{ij}(\hat{F}_{ij} - F_{ij}(\boldsymbol{\theta}))^2$. The relative MSE are computed, for example, in the case of $\hat{\mathbf{F}}_n$, as $\text{relMSE}(\hat{\mathbf{F}}_n) = 100 \times \text{MSE}(\hat{\mathbf{F}}_n)/\sum_{ij}(F_{ij}(\boldsymbol{\theta}))^2$. The effectiveness of the modified resampling algorithm can be clearly seen from the fourth column of the table that shows substantial MSE reduction. The relative MSE reduction in the table is computed as $100 \times (\text{MSE}(\hat{\mathbf{F}}_n) - \text{MSE}(\tilde{\mathbf{F}}_n))/\text{MSE}(\hat{\mathbf{F}}_n)$. In this column also shown within parentheses are variance reduction. The relative variance reduction are computed as $100 \times (A - B)/A$, in which $A = \sum_{ij} \text{var}[\hat{F}_{ij}|\boldsymbol{\theta}]$ and $B = \sum_{ij} \text{var}[\tilde{F}_{ij}|\boldsymbol{\theta}]$.

It would also be interesting to investigate the effect of the modified resampling algorithm on the MSE reduction in the estimators of the unknown elements of the FIM in contrast to a rather 'naive approach' in which the estimates of the unknown elements are simply extracted from $\hat{\mathbf{F}}_n$. To see the improvement in terms of MSE reduction of the estimators of the *unknown* elements of the FIM, the elements corresponding to the upper-left 2×2 block of $\hat{\mathbf{F}}_n$ obtained from the current resampling algorithm are simply replaced by the corresponding *known* analytical elements of the FIM, $\mathbf{F}_n(\boldsymbol{\theta})$. Therefore, the results (shown in Table II) only display the contributions of the MSE from the estimators of the unknown elements of FIM. The relative MSE reductions are reported as earlier by showing $100 \times (\text{MSE}(\hat{\mathbf{F}}_n) - \text{MSE}(\tilde{\mathbf{F}}_n))/\text{MSE}(\hat{\mathbf{F}}_n)$. This table clearly reflects the superiority of the modified resampling algorithm as presented in this work over the current resampling algorithm. In this table, similar results on variance are also reported within parentheses.

Table I-II essentially highlight the substantial improvement of the results (in the sense of MSE reduction as well as variance reduction) of the modified MC based resampling algorithm over the results of the current MC based resampling algorithm. Of course, this degree of improvement is controlled by the values of the known elements of the analytical FIM; see (28) and (29) for Case 1 and (35) and (36) for Case 2.

Cases	Error in FIM estimates		MSE (variance) reduction
	relMSE($\hat{\mathbf{F}}_n$) [MSE($\hat{\mathbf{F}}_n$)]	relMSE($\tilde{\mathbf{F}}_n$) [MSE($\tilde{\mathbf{F}}_n$)]	
Case 1	0.3815 % [1.9318]	0.0033 % [0.0169]	99.1239 % (97.7817 %)
Case 2	0.0533 % [0.2703]	0.0198 % [0.1005]	62.8420 % (97.5856 %)

TABLE I
MSE AND MSE REDUCTION OF FIM ESTIMATES ($N = 2000$).

Cases	MSE($\hat{\mathbf{F}}_n$)
	(and A):
	naive approach
		MSE($\tilde{\mathbf{F}}_n$)
		(and B)
			MSE
			(variance)
			reduction

Case 1	0.1288 (0.0159)	0.1021 (0.0006)	20.7235 % (95.9179 %)
Case 2	0.0885 (0.0030)	0.0878 (0.0002)	0.7930 % (94.4222 %)

TABLE II
MSE COMPARISON FOR $\hat{\mathbf{F}}_n$ AND $\tilde{\mathbf{F}}_n$ ONLY FOR THE UNKNOWN ELEMENTS OF $\mathbf{F}_n(\boldsymbol{\theta})$ ACCORDING TO $\mathbf{F}_n^{(\text{GIVEN})}$ ($N = 100000$) (SIMILAR RESULTS ON VARIANCE ARE REPORTED WITHIN PARENTHESES, $A = \sum_{i=1}^{p}\sum_{j\in\mathbb{I}_i^c} \text{VAR}[\hat{F}_{ij}|\boldsymbol{\theta}]$ AND $B = \sum_{i=1}^{p}\sum_{j\in\mathbb{I}_i^c} \text{VAR}[\tilde{F}_{ij}|\boldsymbol{\theta}]$).

VI. CONCLUSIONS

The present work re-visits the resampling algorithm and computes the variance of the estimator of an arbitrary element of the FIM. A modification in the existing resampling algorithm is proposed simultaneously preserving the known elements of the FIM and improving the statistical characteristics of the estimators of the unknown elements (in the sense of variance reduction) by utilizing the information available from the known elements. The numerical example showed significant improvement of the results (in the sense of MSE reduction as well as variance reduction) of the proposed resampling algorithm over that of the current resampling algorithm.

REFERENCES

[1] J. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control.* Wiley-Interscience, 2003.

[2] ——, "Monte carlo computation of the Fisher information matrix in nonstandard settings," *J. Comput. Graph. Statist.*, vol. 14, no. 4, pp. 889–909, 2005.

[3] S. Das, R. Ghanem, and J. C. Spall, "Asymptotic Sampling Distribution for Polynomial Chaos Representation of Data: A Maximum Entropy and Fisher information approach," in *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, Dec 13-15, 2006, CD rom.

[4] P. Bickel and K. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics, Vol I.* Prentice Hall, 2001.

[5] J. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automat. Control*, vol. 37, no. 3, pp. 332–341, 1992.

[6] J. C. Spall, "Feedback and weighting mechanisms for improving Jacobian (Hessian) estimates in the adaptive simultaneous perturbation algorithm," in *Proc. of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, June 14-16, 2006, pp. 3086–3091.

[7] S. Das, "Efficient calculation of Fisher information matrix: Monte Carlo approach using prior information," Master's thesis, Department of Applied Mathematics and Statistics, The Johns Hopkns University, Baltimore, Maryland, USA, May 2007, `http://dspace.library.jhu.edu/handle/1774.2/32459`.

Performance of $6D$ LuM and FFS SLAM — An Example for Comparison using Grid and Pose Based Evaluation Methods

Rolf Lakaemper, Andreas Nüchter, Nagesh Adluru, and Longin Jan Latecki

The focus of this paper is on the performance comparison of two simultaneous localization and mapping (SLAM) algorithms namely $6D$ Lu/Milios SLAM and Force Field Simulation (FFS). The two algorithms are applied to a $2D$ data set. Although the algorithms generate overall visually comparable results, they show strengths & weaknesses in different regions of the generated global maps. The question we address in this paper is, if different ways of evaluating the performance of SLAM algorithms project different strengths and how can the evaluations be useful in selecting an algorithm. We will compare the performance of the algorithms in different ways, using grid and pose based quality measures.

I. Introduction

The simultaneous localization and mapping (SLAM) problem is one of the basic problems in autonomous robot navigation. In the past many solutions of the simultaneous localization and mapping (SLAM) problem have been proposed [16]. However, it is difficult for engineers and developers to choose a suitable algorithm, due to a lack of true benchmarking experiments. In the well-known Radish (The Robotics Data Set Repository) repository [10] algorithms and results as bitmapped figures are available, but the algorithms have not been compared against each other. A valuable source for state of the art performance are competitions like RoboCup [7], Grand Challenge [4] or the European Land Robotics Trial [8]. However, the aim of such competitions is to evaluate whole systems under operational conditions, but are not well suited for the performance evaluation of vital components like perception. This paper presents two methodologies for comparing the results of state of the art SLAM algorithms, namely $6D$ LuM [3] and FFS [11].

LuM and FFS SLAM, treat the mapping problem as an optimization problem, that is a maximal likelihood map learning method. The algorithms seek to find a configuration ξ^*, i.e., scan poses that maximizes the likelihood of observations and could be written as

$$\xi^* = \underset{\xi}{\operatorname{argmax}} F(\xi),$$

where F is a function measuring the map quality or likelihood.

This paper is organized as follows: After an overview of related work, section III will give a brief description of the compared SLAM algorithms. Section IV presents our evaluation methodology, followed by the results. Section VI concludes.

II. Related Work

A. Robot Mapping

State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion models and perception models. Through integration of these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Besides the robot pose, positions of landmarks are also estimated. Closed loops, i.e., revisiting a previously visited area of the environment, play a special role here: Once detected, they enable the algorithms to bound the error by deforming the mapped area to yield a topologically consistent model. For e.g. [15] addresses the issues in loop-closing problems. Several strategies exist for solving SLAM. Thrun [16] surveys existing techniques, like, maximum likelihood estimation, Expectation Maximization, Extended Kalman Filter, Sparsely Extended Information Filter SLAM. FastSLAM [18] and its improved variants like [9] use Rao-Blackwellized particle filters.

SLAM in well-defined, planar indoor environments is considered solved. However, little effort has been spent in *comparing* the performance evaluation of the SLAM algorithms. Given vast literature and various successful approaches for SLAM, such comparative studies are needed to choose appropriate SLAM algorithms for specific applications.

B. Performance Evaluation

Most research in the SLAM community aims at creating consistent maps. Recently, on the theoretical side of SLAM, Bailey et al. proves that EKF-SLAM fails in large environments [1] and FastSLAM is inconsistent as a statistical filter: it always underestimates its own error in the medium to long-term [2] that is it becomes over-confident. Besides focusing on such consistency issues, little effort has been made in *comparative* studies of SLAM algorithms.

Comparing two or more SLAM algorithms needs quantitative performance metrics like robustness, rate of convergence, quality of the results. Though the metrics used for comparison

Rolf Lakaemper, Nagesh Adluru and Longin Jan Latecki are with the Department of Computer and Information Sciences, Temple University, Philadelphia, U.S.A. lakamper@temple.edu, nagesh@temple.edu, latecki@temple.edu

Andreas Nüchter is with the Knowledge Systems Research Group of the Institute of Computer Science , University of Osnabrück, Germany. nuechter@informatik.uni-osnabrueck.de

in this paper are not completely new, the use of them in this context has not been done before, to the best of our knowledge. In this paper we mainly focus on the rate of convergence and quality of results of the two algorithms. They are measured in two different ways: occupancy grid based and pose based, as described in section IV.

III. Description of Mapping Algorithms

A. FFS

FFS [11] treats map alignment as an optimization problem. Single scans, possibly gained from different robots, are kept separately but are superimposed after translation and rotation to build a global map. The task is to find the optimal rotation and translation of each scan to minimize a cost function defined on this map. FFS is a gradient descent algorithm, motivated by the dynamics of rigid bodies in a force field. In analogy to Physics, the data points are seen as masses, data points of a single scan are rigidly connected with massless rods. The superimposition of scans defines the location of masses, which induces a force field. In each iteration, FFS transforms (rotates/translates) all single scans simultaneously in direction of the gradient defined by the force field under the constraints of rigid movement; the global map converges towards a minimum of the overlying potential function, which is the cost function. FFS is motivated by physics, but is adapted to the application of map alignment. It differs in the definition of the potential function, and in the choice of the step width of the gradient descent. The potential is defined as

$$P = \frac{1}{2} \sum_{p_i \in \mathcal{P}} \sum_{p_j \in \mathcal{P} \setminus p_i} \int_{\infty}^{r} \frac{m_1 m_2 \cos(\angle(p_i, p_j))}{\sigma_t \sqrt{2\pi}} e^{\left(-\frac{z^2}{2\sigma_t^2}\right)} dz \quad (1)$$

with $r = \sqrt{(X-x)^2 + (Y-y)^2}$,$p_i = (X,Y)$,$p_j = (x,y) \in \mathcal{P}$, $\mathcal{P}$ is the set of all transformed data points.

The potential function measures the probability of visual correspondence between all pairs of data points based on distance, direction and visual importance of data points: in (1) m_1, m_2 denote the visual importance of two data points, $\angle(p_i, p_j)$ the difference of direction of two points. Defining the visual importance of points dynamically is a simple interface to incorporate low or mid level perceptual properties (e.g. shape properties) into the of the global map into the optimization process. In contrast to algorithms like ICP, FFS does not work on optimization of nearest neighbor correspondences only, but (theoretically) takes into account all pairs of correspondences. Different techniques built into FFS drastically reduce the computational complexity.

FFS is steered by two parameters, σ_t in eq. 1 and the step width Δ_t of the gradient descent. σ steers the influence of distance between points. Initially set to a big value to accumulate information from a large neighborhood, it linearly decreases over the iterations to focus on local properties. The step width Δ_t in the FFS gradient descent is defined by an exponentially decreasing cooling process, similar to techniques like simulated annealing. Initially set to a high value it allows for significant transformations to possibly escape local minima. Decreasing the step enables local adjustment in combination with a low σ_t.

To conclude, the basic properties of FFS are

1) Data point correspondences are not made by a hard decision, but an integral between pairs of points defines the cost function instead of hard 'nearest neighbor' correspondences
2) FFS is a gradient approach, it does not commit to an optimal solution in each iteration step
3) The iteration step towards an optimal solution is steered by a 'cooling process', that allows the system to escape local minima
4) FFS transforms all scans simultaneously thus searching in $3n$ space of configurations with n scans.
5) FFS easily incorporates structural similarity modeling human perception to emphasize/strengthen the correspondences

B. $6D$ LuM

To solve SLAM, a $6D$ graph optimization algorithm for global relaxation based on the method of Lu and Milios [12] is employed, namely Lu and Milios style SLAM (LUM). Details of the $6D$ optimization, i.e., how the matrices have to be filled, can be found in [3]:

Given a network with $n+1$ nodes $X_0, ..., X_n$ representing the poses $V_0, ..., V_n$, and the directed edges $D_{i,j}$, we aim to estimate all poses optimally to build a consistent map of the environment. For simplicity, we make the approximation that the measurement equation is linear, i.e.

$$D_{i,j} = X_i - X_j.$$

An error function is formed such that minimization results in improved pose estimations:

$$\mathbf{W} = \sum_{(i,j)} (D_{i,j} - \bar{D}_{i,j})^T C_{i,j}^{-1} (D_{i,j} - \bar{D}_{i,j}). \quad (2)$$

where $\bar{D}_{i,j} = D_{i,j} + \Delta D_{i,j}$ models random Gaussian noise added to the unknown exact pose $D_{i,j}$. The covariance matrices $C_{i,j}$ describing the pose relations in the network are computed based on the paired points of the ICP algorithm. The error function Eq. (2) has a quadratic form and is therefore solved in closed form by Cholesky decomposition in the order of $\mathcal{O}(n^3)$ for n poses ($n \ll N$). The algorithm optimizes Eq. (2) gradually by iterating the following five steps [3]:

I) Compute the point correspondences (n closest points) using a distance threshold (here: 20 cm) for any link (i,j) in the given graph.
II) Calculate the measurement vector $\bar{D}_{ij}$ and its covariance C_{ij}.
III) From all $\bar{D}_{ij}$ and C_{ij} form a linear system $\mathbf{GX} = \mathbf{B}$.
IV) Solve for $\mathbf{X}$
V) Update the poses and their covariances.

For this GraphSLAM algorithm the graph is computed as follows: Given initial pose estimates, we compute the the number of closest points with a distance threshold (20 cm). If there are more than 5 point pairs, a link to the the graph is added.

To summarize, the basic properties of $6D$ LuM are

1) Data point correspondences are made by a hard decision using 'nearest neighbor' point correspondences
2) $6D$ LuM computes the minimum in every iteration
3) $6D$ LuM transforms all scans simultaneously
4) This GraphSLAM approach has been extended successfully to process $3D$ scans with representation of robot poses using 6 degrees of freedom.

In this paper, we process 2D laser range scans with the $6D$ LuM algorithm, i.e., in the range data the height coordinate is set 0. In this case the the algorithms shows the behaviour of the original Lu and Milios [12] GraphSLAM method.

IV. Evaluation

Evaluation of SLAM algorithms applied to real world data often faces the problem that ground truth information is hard to collect. For example, in settings of Search and Rescue environments, data sets are scanned, which usually have no exact underlying blue print, due to the nature of the random spatial placement of (sparse) landmarks and features. Hence map inherent qualities, like entropy of the distribution of data points, must be used to infer measures of quality that reflect their ability to map the real world. In the experiments, we will compare the performance of $6D$ LuM and FFS SLAM using a grid based and a pose based approach. Especially the grid based approach will be compared to visual inspection, which in this setting could be seen as a subjective ground truth of the performance evaluation. The reason for the choice of $6D$ LuM and FFS SLAM are the following:

- Both, $6D$ LuM and FFS SLAM, are state of the art algorithms to simultaneously process multiple scans, which is needed in settings of multi-robot mapping, which is a problem that currently has stronger focus in robot mapping.
- By visual inspection, $6D$ LuM and FFS perform, intuitively spoken, alike, although differing in details. Evaluation of the algorithms should be able to report this behavior.

It should be noted that $6D$ LuM is applied here to a $2D$ dataset, to compare it to the currently available version of the FFS algorithm, which works on $2D$ scan data only. Hence the LuM performance is only evaluated on three dimensions.

A. Occupancy Grid Based

Occupancy grids are used to represent the environment by discretizing the space into grid cells that have probabilistic occupancy values accumulated by sensor readings. They were introduced by [13] and are very popular in SLAM community. Learning occupancy grids is an essential component of the SLAM process. Once built they can be used to evaluate the likelihood of the sensor readings and also be used for guiding the exploration task as they are useful for computing the information gain of actions.

The likelihood of sensor readings is computed usually using different sensor models like beam-model, likelihood-field model or map-correlation model [17]. The information gain of actions can be computed using change in entropy of the grid. We use these basic ideas to compare the outcome of the two SLAM algorithms.

We use the beam penetration model described in [6] to compute likelihood of the sensor readings. Entropy of the grid is computed as described in [14]. Once the final map is obtained we compute the log-likelihood of *all* sensor readings with trajectory given out by the algorithm as

$$\mathcal{L}(m, \mathbf{x}_{1:n}) = \sum_{i=1}^{n} \sum_{j=1}^{K} \log(p(z_{ij})|\mathbf{x}_i, m))$$

where m is the final occupancy grid, $\mathbf{x}_{1:n}$ is the final set of poses, K is the number of sensor readings at each pose and $p(z_{ij}|\mathbf{x}_i, m)$ is computed using beam-penetration model as in [6]. The log-likelihood ranges from $-\infty$ to 0 and the higher it is the better the algorithm's output.

The entropy of the map is computed based on the common independence assumption about the grid cells. Since each grid cell in the occupancy grid is a binary random variable the entropy of $H(m)$ is computed as follows as described in [14].

$$H(m) = -\sum_{c \in m} p(c) \log p(c) + (1 - p(c)) \log(1 - p(c))$$

Since the value of $H(m)$ is not independent of the grid resolution it's important either to use same resolution or to weight the entropy of each cell with its size when comparing output from two algorithms. The lower the entropy of the map the better the outcome is. It is important to note that the entropy of the map and the likelihood-scores are not completely uncorrelated.

B. Pose Based

The occupancy grid based evaluations are very useful in the sense that they do not need "ground truths" to compare the results. But their memory requirements are proportional to the dimensionality and size of the environment. The pose based evaluations have an advantage in terms of memory requirements but require "ground truth" data to compare to. Here we present the technique that can be used to measure the quality of the output of SLAM algorithm assuming ground truth trajectory *is* available. The ground truth data can be obtained by surveying the environment as done in [5].

The SLAM algorithm gives out a final set of poses $\mathbf{x}_{1:n}$. Let the set of ground truth poses be $\mathbf{x}^G_{1:n}$. Since each pose in $2D$ mapping has three components viz. x, y, θ we compute the average error in each of the components. It is important that both the output of SLAM algorithm and the ground truth poses are in the same global frame. This could be done by rotating and translating the set of poses such that the first corresponding pose in each set is $(0, 0, 0)$. Once the poses are in same global frame the average error in each component is computed as:

$$E(x) = \frac{1}{n}\sum_{i=1}^{n}|x_i - x_i^G|$$

$$E(y) = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^G|$$

$$E(\theta) = \frac{1}{n}\sum_{i=1}^{n}\cos^{-1}(\cos(\theta_i - \theta_i^G))$$

$E(\theta)$ is computed as shown above so that the difference between the orientations is always between 0 and π.

V. Experiments

A. The Data, Visual Inspection

Both algorithms will be evaluated based on their performance on the NIST disaster data set with the same initial set of poses, see fig. 1. The data set consists of 60 scans and is especially complicated to map, since the single scans have minimal overlap only, and no distinct landmarks are present in the single scans. For this data set, no reliable ground truth pose data exists. This configuration was gained by random distortion of a manually gained global map.

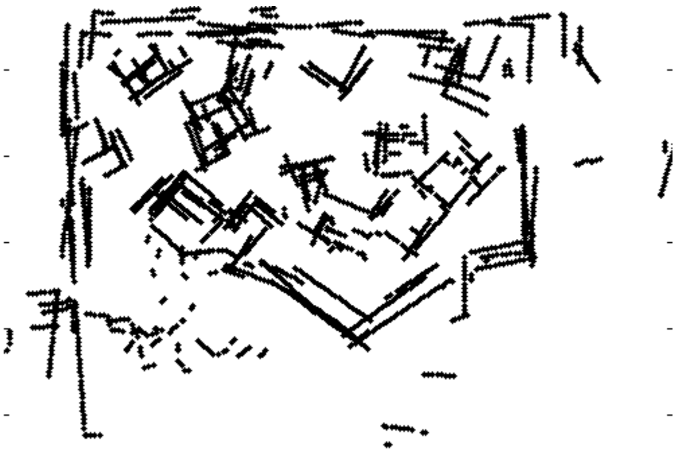


Fig. 1. Initial configuration of the NIST data set. The data consists of 60 scans. The scale is in centimeters.

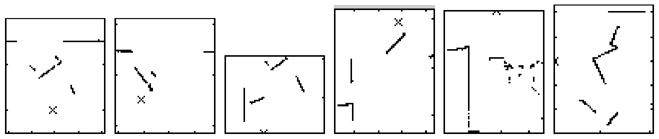


Fig. 2. 6 example scans of the NIST data set. In fig. 1, they can be located on the left side.

Six sample scans are shown in fig. 2. The final results of LuM and FFS respectively are shown in fig. 3. Visual inspection of 3 shows the following properties:

- The overall appearance of both approaches is equal.
- The mapping quality in different details is different: while FFS performs better in the left half, especially in the top left quarter, LuM shows a more visually consistent result in the right half, especially the top right corner.

To test if the evaluation does reflect these properties, we performed the following tests:

- First, entropy (and additional, likelihood-score) of the entire global maps (global evaluation) of both algorithms over all iterations are computed. This should reflect the behavior of both algorithms to converge towards optimal values, which should be in the same order of magnitude for both metrics.
- To check the evaluation of the different quality of mapping details in different areas, we split the result maps into four quarters and evaluated separately (regional evaluation).

In the LuM algorithm, 500 iterations were performed. FFS stopped automatically after 50 iterations, detecting a condition of changes in poses below a certain threshold. To compare all iteration steps, we extended the final result (iteration 50) to iterations $51 - 500$.

B. Grid Based Global Evaluation

The entropies and the likelihood-scores of the maps as the algorithms progress are shown in the fig. 4(a) and 4(b) respectively.

Please note the different scale on the iteration axis in the intervals $[1 - 50]and(50 - 500]$, in the first interval the iterations increase in units of 1, whereas in the second they increase in units of 10. This holds for all following figures.

You can see that the entropy decreases non-monotonically in case of FFS while in case of LuM it tends to be monotonically decreasing. This is based in the different nature of both algorithms: FFS is gradient based approach that has a built in ”cooling strategy” for the step width to possibly escape local minima. In the beginning, FFS takes bigger steps, yielding a non monotonic behavior in its target function, which is also visible in the entropy. LuM optimizes its pose in each iteration, leading to a more smooth behavior, bearing the risk of being caught in local minima. This is also reflected in the convergence behavior in terms of speed: since LuM commits to optimal solutions earlier, it converges faster in the beginning, slowing down afterwards. FFS is slower (or more positively: more careful) in the first steps, due to the choice of step width that causes a jittering behavior. After the step width is balanced, FFS reaches its optimum very quickly. Interestingly, in both cases the near optimum value is reached after about 50 iterations.

The entropy score of both algorithms is comparable, which fulfills the expectations based on the visual inspection.

Similar behavior is observed in the likelihood scores. Hence the grid based evaluation is able to reflect the properties of both algorithms in the case of global evaluation.

C. Regional Evaluation

The maps are split into four regions, being North-West, North-East, South-West, South-East. Only the results for entropies are shown here, the likelihood scores did not lead to

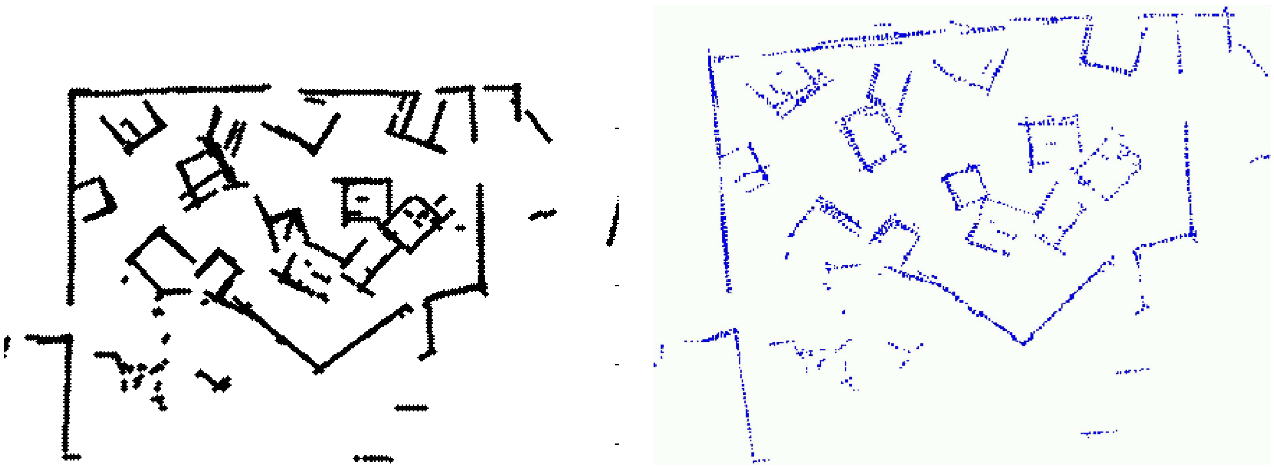
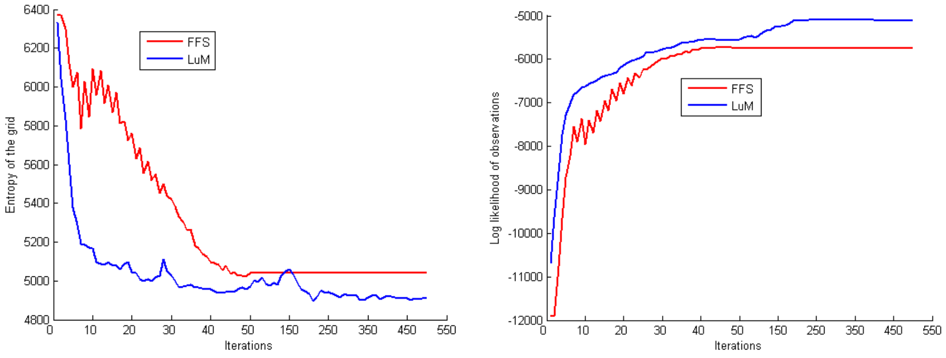


Fig. 3. Result of FFS (left) and LuM (right) on NIST data set, initialized as in 1. Evaluated by the overall visual impression, both algorithms perform comparably. Differences in details can be seen especially in the top left, where FFS performs better, and the top right, where LuM is more precise.



(a)

(b)

Fig. 4. (a): The entropy of the map $H(m)$ at various stages of FFS and LuM. (b): The likelihood-score $\mathcal{L}(m, \mathbf{x}_{1:n})$ at various stages of the algorithms. Please note the different scale on the iteration axis in the intervals $[1 - 50] and (50 - 500]$.

additional further information. We expect better results for FFS in the North-West region, whereas LuM should outperform FFS in the North-East region, results for the southern regions should not vastly differ from each other.

The results are presented in fig. 5. fig. 5(a) shows the behavior for the North-West region of the map while 5(b) shows for North-East, 5(c) for South-West and 5(d) for South-East.

In accord with visual inspection, FFS is evaluated to perform better on the North-West region (fig. 5(a)) while LuM performs better in other regions. However, looking at the difference in final values, we can see that they always differ in ranges between ~ 30 and ~ 80 units: (a) $\sim 430 - 480$, (b) $\sim 3200 - 3280$, (c) $\sim 278 - 309$, (d) $\sim 950 - 1000$. Hence, although the tendency in the north regions is correct, the comparison to the southern regions, which should yield a smaller distance in values, does not clearly verify the correct estimation.

D. Global Pose Based Estimation

Pose based estimation needs a ground truth reference pose, see section IV-B. Since a ground truth for the NIST data set is not available, we just use the final set of poses of each algorithm. This necessarily leads to a graph that converges to an error of zero. Hence it does not give any information about the actual mapping quality, but it shows the behavior of the algorithms in terms of rate of convergence. Fig. 6 shows the behavior of the algorithms using error-metrics presented in section IV-B.

With respect to path to convergence, the pose based evaluation also shows the same properties of LuM and FFS as the grid based: LuM is ”more monotonic”, while FFS has jittering behavior. Interestingly the pose based evaluation shows FFS converging faster, which is in contrast to the result using grid based evaluation. While reasons for this different result will be topic of future discussion, it again shows that the choice of evaluation method has an influence on the property description of the algorithms.

VI. Conclusions and Future Work

This paper has presented a performance evaluation of two simultaneous localization and mapping (SLAM) algorithms namely $6D$ Lu/Milios SLAM ($6D$ LUM) and Force Field Simulation (FFS). These two algorithms have been applied to a $2D$ data set, provides by NIST. The results have been compared using two different metrics, i.e., an occupancy grid

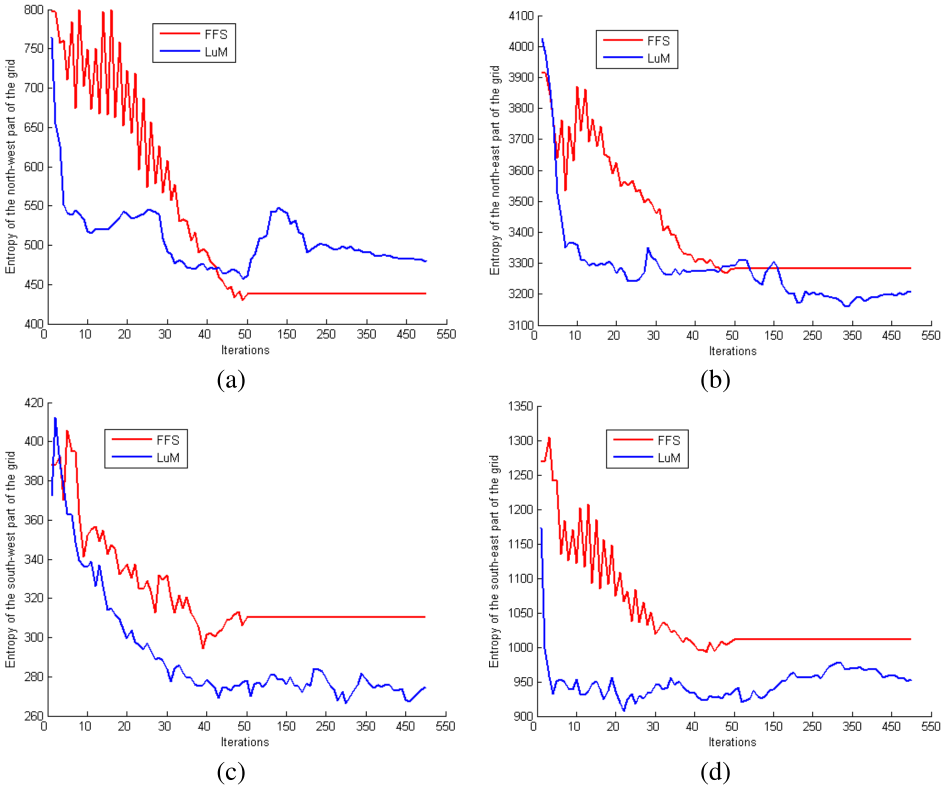


Fig. 5. (a): $H(m)$ for North-West (top-left quadrant) region of m. (b): $H(m)$ for North-East (top-right quadrant). (c): For South-West (bottom-left). (d): For South-East (bottom-right)

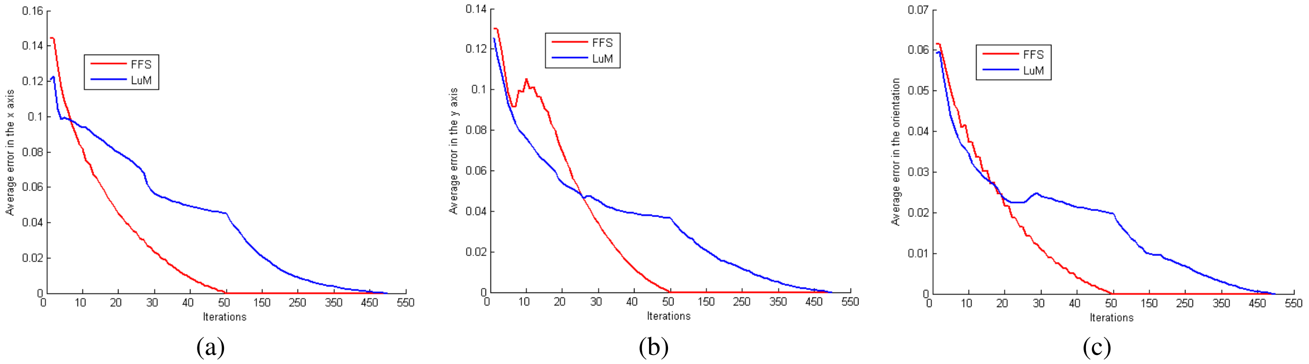


Fig. 6. (a): $E(x)$ for FFS and LuM. (b): $E(y)$. (c): $E(\theta)$ for FFS and LuM. The errors $E(x)$ and $E(y)$ are given in meters, $E(\theta)$ is given in radians.

based method and a pose based method. In addition these metrics have checked by visual inspection for plausibility. $6D$ LUM and FFS show similar performances on the data set considered in this paper.

Needless to say a lot of work remains to be done. The two algorithms have been on one data set. However, in robotic exploration task the environment is the greatest element of uncertainty. Mapping algorithms might fail in certain environments. In future work we plan to benchmark mapping algorithms using more suitable standardized tests and evaluate on automatically generated test cases. The grid and pose based evaluation methods will be used for these evaluations.

References

[1] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the EKF-SLAM Algorithm. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, Bejing, China, 2006.

[2] Tim Bailey, Juan Nieto, and Eduardo Nebot. Consistency of the FastSLAM Algorithm. In *IEEE International Conference on Robotics and Automation (ICRA '06))*, Orlando, Florida, U.S.A., 2006.

[3] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally Consistent 3D Mapping with Scan Matching. *Journal of Robotics and Autonomous Sytems*, 2007, (To appear).

[4] Defense Advanced Research Projects Agency (DARPA) Grand Challenge. `http://www.darpa.mil/grandchallenge/index.asp`, 2007.

[5] M. W. M. G. Dissanayake, P. M. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotic and Automation (TRA)*, 27(3):229–241, 2001.

[6] A. Eliazar and R. Parr. DP-SLAM 2.0. In *IEEE International Conference on Robotics and Automation (ICRA '04)*, 2004.

[7] The RoboCup Federation. http://www.robocup.org/, 2007.

[8] FGAN. http://www.elrob2006.org/, 2007.

[9] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics (TRO)*, 23:34–46, 2007.

[10] A. Howard and N. Roy. Radish: The Robotics Data Set Repository, Standard data sets for the robotics community. http://radish.sourceforge.net/, 2003 – 2006.

[11] R. Lakaemper, N. Adluru, L. J. Latecki, and R. Madhavan. Multi Robot Mapping using Force Field Simulation. *Journal of Field Robotics, Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems. (To appear)*, 2007.

[12] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333 – 349, October 1997.

[13] H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Mag.*, 9(2):61–74, 1988.

[14] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proceedings of Robotics: Science and Systems (RSS '05)*, pages 65–72, Cambridge, MA, USA, 2005.

[15] C. Stachniss, D. Hähnel, W. Burgard, and G. Grisetti. On actively closing loops in grid-based FastSLAM. *Advanced Robotics*, 19(10):1059–1080, 2005.

[16] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[17] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press Cambridge, 2005.

[18] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, U.S.A, April 2000.

Smart Assembly: Industry Needs and Challenges

John A. Slotwinski, Ph.D.
National Institute of Standards
and Technology
100 Bureau Drive, Mail Stop 8200
Gaithersburg, MD USA
john.slotwinski@nist.gov

Robert B. Tilove, Ph.D.
General Motors
30500 Mound Road
Warren, MI, USA
robert.tilove@gm.com

***Abstract*—** In recent years globalization has radically changed the nature of manufacturing (including manufacturing engineering), with an increasing emphasis on the management of complex, dynamic, interconnected supply chains. Manufacturing has become information and knowledge intensive, requiring the sharing of information accurately, inexpensively, and seamlessly throughout the extended enterprise and supply chain. Vertical integration is declining as a competitive advantage, and Original Equipment Manufacturers (OEMs) are instead focusing on managing core technologies and critical assets, emphasizing systems integration, assembly, service and marketing. To an increasing extent, part fabrication is globally outsourced, but there remains a business case for placing final assembly close to the customer.

"Smart Assembly" is about re-inventing assembly processes (engineering and operations) to succeed in this new environment. Smart Assembly may be a key aspect of future, thriving manufacturing enterprises.

General Motors (GM), in collaboration with the National Institute of Standards and Technology's (NIST) Manufacturing Engineering Laboratory (MEL), is developing a broad industry definition and vision for Smart Assembly, and is beginning to develop the technology and business process "roadmaps" that will provide a framework to focus and prioritize both current and future research and development (R&D) in this area.

In this paper, we will:

• Present a high-level business case for the importance of manufacturing in general, and Smart Assembly in particular.
• Present a working definition of Smart Assembly, and a vision of what Smart Assembly might look like in the future
• Describe efforts to increase awareness of Smart Assembly, through the creation of a smart assembly working group, which is hoping to refine the vision, scope, business case scenarios and roadmaps for what we hope will ultimately be a National Smart Assembly activity.
• Describe how Smart Assembly is being considered in the context of MEL's strategic planning.

***Keywords*:** *Assembly, Assembly Processes, Intelligent Manufacturing, Smart Assembly*

I. Introduction

The United States Council for Automotive Research (USCAR) is the umbrella organization of Chrysler, Ford and General Motors, which was formed in 1992 to further strengthen the technology base of the domestic auto industry through cooperative research.

On December 9, 2004, USCAR and the U.S. Department of Commerce's Technology Administration, through the National Institute of Standards and Technology (NIST), announced a new partnership to facilitate technological research and technology policy analysis focused on improving the manufacturing competitiveness of the U.S. automotive industry. Since then, the authors have been collaborating on a specific USCAR project related to the interactive modeling of assembly operations involving flexible parts such as hoses and cables.

In an effort to identify opportunities for expanding the collaborative research portfolio between General Motors (GM) Manufacturing Systems Research Laboratory (MSR) and NIST (perhaps, but not necessarily under the USCAR umbrella), the second author visited NIST's Manufacturing Engineering Laboratory (MEL) in 2005 to present a GM Research and Development (R&D) perspective on Virtual Manufacturing, and to gain a better understanding of MEL's mission, capabilities, and current projects. The primary observations/conclusions from this visit were:

• MEL technical capabilities in information technologies, metrology (including sensing & perception), controls, interoperability, and standards are world class.
• Current MEL programs of most relevance to MSR appeared to be Intelligent Control of Mobility Systems, Manufacturing Interoperability, and Smart Machining Systems.
• While these activities exhibit significant technical synergies with MSR interests in next generation automotive assembly systems and technologies, with a few notable exceptions (e.g. Virtual Manufacturing Environments, Next Generation Robots), the specific applications and context for the work at MEL were considerably different (military vehicles, product design & engineering, and machining).
• MEL researchers and management expressed a strong interest in better understanding industry needs in "smart assembly systems and technologies" in relation to their

mission and capabilities, and in further exploring opportunities for re-aligning and/or focusing their work to better address these needs.

The second author was invited to join the Manufacturing Engineering Laboratory at NIST for a six month appointment as a Visiting Scientist to (1) produce a review paper defining the state of the art and industry needs in "Smart Assembly", and (2) initiate the development and documentation of a conceptual framework, including information models and architecture, for "Smart Assembly", working in collaboration with NIST scientists in three MEL divisions (Manufacturing Metrology, Intelligent Systems, and Manufacturing Systems Integrartion). GM R&D approved a Domestic Temporary Assignment for this purpose from September 2006 – February 2007. This paper, and other documents referenced within, comprise the key deliverables and results of this special assignment.

The remainder of this paper is organized as follows:

- First, we present a view of manufacturing in today's "flat world" environment.
- Second, we review the vision and industry needs in "Smart Assembly". This material is drawn primarily from a workshop conducted at NIST in October 2006.
- Finally, we present an "Grand Challenge," a visionary example of what might be accomplished in practice should certain aspects of smart assembly be realized.

Taken together, these sections provide the starting point for the development and documentation of information models and architectures for "Smart Assembly."

II. MANUFACTURING IN THE "FLAT WORLD"

Manufacturing in today's "flat world" is network-centric, and employs dynamic, complex, interconnected supply chains. It is information and knowledge intensive, and requires the capability to share information accurately, inexpensively, and seamlessly. Original Equipment Manufacturers (OEMs) are focusing on core technologies and critical assets, and are transitioning their focus towards systems integration, assembly, service, and marketing, as vertical integration declines as a competitive strategy. Components are fabricated globally, and there is an emerging business case for locating final assembly close to customers. Smart Assembly is about reinventing assembly processes to succeed in this new environment.

Successful, correct assembly requires that many other things are first done successfully. Responsive, efficient assembly of high quality products with a high degree of product variation is the result of doing many things right, and these things are highly interdependent. These include: design for assembly, virtual simulation and validation, flawless launches, highly trained workers, knowledge asset management, supply chain management, real-time decision making, fast response to problems, maintenance, and line balancing.

III. VISION AND INDUSTRY NEEDS IN "SMART ASSEMBLY"

On October 3-4, 2006, approximately 60 researchers, software and equipment suppliers, and end users convened at NIST to discuss the next generation Smart Assembly (SA) capability. The team developed a vision for SA, determined basic needs and gaps, defined key enabling technologies, assessed interests in establishing an industry-led SA initiative, and defined next steps. [1]

A. Business Case and State of the Art in "Smart Assembly"

Manufacturing operations involve the preparation and processing of raw materials, the creation of components, and the assembly of components into subassemblies and finished products. The broader scope of manufacturing includes innovation, design, engineering, and management of life cycle performance.

Globalization is redefining the distribution of these manufacturing functions and operations. In recent years, many manufacturing sectors have gone offshore to produce, except for high-end, specialty products. Lower offshore labor costs make it difficult for U.S. manufacturers to produce cost competitive components in the United States. When one also considers that manufacturing costs of a product produced in China is 30 % to 50 % lower than the same product produced in the U.S., the near term threat to America's manufacturing base is very real.

The importance of manufacturing to America's economic well being remains very high. Manufacturing is the backbone of our economy and the cornerstone behind our national defense. It is the major source of our high economic leverage, well paying jobs, and R&D investment and innovation. Manufactured products account for over $900 billion worth of U.S. exports -- nearly two-thirds of all U.S. exports -- and manufacturing's value-added to the U.S. economy is approximately $2 trillion per year, contributing 12 % of the U.S. Gross Domestic Product (GDP). Every dollar invested in manufacturing spawns another $1.43 for the economy, and in the automotive industry, for example, every job results in 6.6 spin-off jobs in other industries (electronics, financial, materials, etc.) [2], [3].

To remain strong in the global marketplace, the U.S. must maintain its ability to produce excellent products cost effectively. This is not an easy challenge. Manufacturing is evolving in the "Flat World". Today's manufacturing is network centric and requires the effective management of dynamic, complex, and interconnected supply chains. Manufacturers have become information & knowledge intensive by necessity, demanding the ability to share information accurately, inexpensively, and seamlessly.

There is an emerging business case that considers service, logistics, shipping costs, regulatory and policy issues, and market intelligence for placing final assembly operations close to the customer. Focusing resources on improving technologies and processes associated with assembly of components into final products could conservatively achieve a $100 Billion/yr productivity increase for the U.S. [1].

Today, the vast majority of assembly operations are manual. In a typical automotive plant, at least 95% of assembly operations are manual, with automation being used only on simple tasks. Some manual tasks have been improved by providing operators with machine-assisted processes to help with ergonomics, productivity, and quality. Still, the trend is outsourcing of non-critical tasks to the most competitive supplier. The increasing reliance on information technology (IT) to optimize and operate the supply chain has become an integration nightmare for many companies, especially small and medium sized manufacturers who do not have the resources to develop customized integration solutions.

Assembly efficiency and capability is a competitive discriminator in every product manufacturing sector. Time is a key driver for successful assembly operations. Companies that can move an innovative new product from the drawing board to the loading dock before anyone else gain a huge advantage in profitability. Several case studies have demonstrated dramatic results of applying best practices and technologies to assembly operations. For example, Toyota's V-Comm Digital Mockup program[1] validates both product and manufacturing processes through digital assembly, reducing lead time for production by 33 %, design changes by 33 %, and development costs by 50 %. Boeing's advancements in assembly and supply chain integration on their 777 aircraft program have reduced product cycle development time by 91 % and reduced labor costs by 71 %.

B. Vision and Definition of Smart Assembly

Following the workshop, various yet similar definitions of Smart Assembly evolved. For the purpose of this paper, we adopt the following definition currently being developed by NIST's MEL:

> *Smart Assembly is the incorporation of learning, reconfigurability, human-machine collaboration, and model-based techniques into assembly systems to improve productivity, cost, flexibility, responsiveness and quality.*

SA goes well beyond traditional automation and mechanization to exploit the effective collaboration of man and machine in engineering and in operations. It integrates highly skilled, multi-disciplinary work teams with self-integrating and adaptive assembly processes. It unifies "virtual" and "real-time" information to achieve dramatic improvements in productivity, lead time, agility, and quality.

The vision for Smart Assembly is a system consisting of the optimal balance of people and automation interacting effectively, efficiently, and safely. People work in knowledgeable, empowered work teams that utilize best assembly practices and technologies. Virtual optimization and validation of assembly processes are used to ensure the best designs work the first time. Effective integration of automation and information technology into the human assembly process maximizes total system performance on a consistent basis. And finally, flawless execution of supply chain and product life cycle processes successfully synchronizes the entire assembly.

To summarize the key characteristics of Smart Assembly Systems:

- Empowered, knowledgeable people: A multi-disciplined, highly skilled workforce is empowered to make the best overall decisions.

- Collaboration: People and automation working in a safe, shared environment for all tasks.

- Reconfigurable: Modular "plug and play" system components are easily reconfigured and reprogrammed to accommodate new product, equipment, and software variations, and to implement corrective actions.

- Model and Data Driven: Modeling and simulation tools enable all designs, design changes, and corrective actions to be virtually evaluated, optimized, and validated before they are propagated to the plant floor. The "virtual" models and real-time plant floor systems are synchronized.

- Capable of Learning: Self integrating and adaptive assembly systems prevent repeated mistakes and avoid new ones.

C. Enabling Technologies for Smart Assembly

We can partition the significant enabling technologies for Smart Assembly into four inter-dependent thrust areas as illustrated below. The rows correspond to technology areas

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

(described below) for which R&D roadmaps can be developed. The columns correspond to industry-specific application (or “grand challenge”) scenarios involving a high degree of integration across the technology areas. These use-case scenarios outline the significant milestones, deliverables, and capability demonstrations that could be included in a future focused Smart Assembly R&D program (e.g. a national testbed).

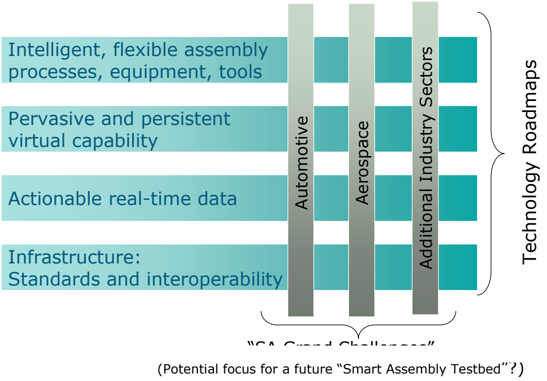

Intelligent, flexible assembly processes, equipment, tools
Pervasive and persistent virtual capability
Actionable real-time data
Infrastructure: Standards and interoperability
Automotive
Aerospace
Additional Industry Sectors
Technology Roadmaps
“SA Grand Challenges”
(Potential focus for a future “Smart Assembly Testbed”?)

Fig. 1 Smart Assembly Enabling Technologies

1) Intelligent Flexible Assembly Processes, Equipment, and Tools

Smart Assembly processes, equipment, and tools must be modular, low cost, and reusable. R&D in this area focuses on next generation robotics, sensors, controls, effectors, material handling, and assembly concepts.

Intelligent, safe cooperative robots will eliminate costly, hard, restrictive safety fences. Modular, multifunctional assembly system components will be readily reconfigurable – ultimately autonomously reconfigurable – allowing rapid changeover to initiate production of new products. These assembly systems will be self-integrating and self-configuring, negotiating their respective “roles and responsibilities” based on digital knowledge of product, process, and business requirements as defined by applicable product and process models. The assembly process will ensure sequenced material delivery to point-of-use to eliminate inventory waste and unnecessary material handling.

2) Accurate, easy-to-use, pervasive, persistent, virtual capability

We must be able to virtually launch a factory, optimize its operation and eliminate errors prior to production. R&D in this area focuses on next generation virtual modeling and simulation technologies to enable changes to the assembly system to be emulated in a cost effective manner before deployment, both to ensure they work, and to minimize disruption to on-going production operations.

A virtual capability will drive collaborative systems engineering. Product requirements and manufacturing capabilities and infrastructure will drive the creation of product models for Smart Assembly. The product models will support the definition and development of best assembly processes, with optimization and evaluation done in the “model space” of the virtual environment. The process models will be the foundation for intelligent closed-loop process control and will be robust enough to transfer directly to operations. Because the virtual capability is integrated into the manufacturing information infrastructure and business planning process, the models will be continuously updated so that the virtual plant floor accurately synchronizes with the real plant floor throughout the product life cycle.

3) Real time actionable information for man and machine

Real time, actionable information provides timely and accurate decision support for people and automation to keep operations, maintenance, and fault recovery activities optimized. R&D in this area focuses on wireless and web-enabled monitoring, prognostics, and intelligent maintenance.

The assembly system will be a sense-, analyze-, advise-and-respond environment. Sensors will monitor every parameter that is important to the operation, and control limits will be set for all parameters. The human-in–the-loop will be aided by excellence-in-information, instructions on what and how to perform, and monitor assurance of acceptable completion of tasks. The state of the assembly will be evaluated at all times, and any deviations will be made known. The assembly environment will function in a manner that is similar to the immune system of the human body, wherein anomalies that have no obvious symptoms are responded to in a very effective manner.

This mindset is giving birth to a new discipline called immune systems engineering. It is an environment wherein there is sufficient intelligence to monitor key parameters and determine, mandate, and ensure the best response. Self-diagnosing and self-healing will be attributes of the systems. The intelligent closed loop assembly environment will be achieved through advances in control and manufacturing diagnostics & prognosis technologies that embrace open architecture and modular functionality. The control & diagnostic function will be linked to the model-based environment to support the application of knowledge with data to enable automated generation of the necessary information to drive, control, monitor, and maintain assembly operations.

4) Real time actionable information for man and machine

The Smart Assembly environment must be interoperable at all levels (e.g. tool, cell, zone, line, plant, enterprise), with plug-and-play hardware and software (both virtual and physical) that communicate seamlessly across domains and different commercial toolsets. Both sending and receiving devices will speak the same language or have integral real-time translation incorporated into the communication systems. R&D on the development of harmonized standards for sufficient coverage of all assembly functions is the minimum requirement of the future strategy, and on interoperability, conformance, and performance testing relative to these standards.

IV. SMART ASSEMBLY GRAND CHALLENGES

The enabling technologies for Smart Assembly (rows in Figure 1) have been the subject of active R&D for many years and are not "new ideas." The visionary elements of "smart assembly" are not within the enabling technologies per se, but rather on the unique and substantial opportunities enabled by a "deep integration" across technologies. The current focus of the Smart Assembly activity is to develop and document "Smart Assembly Grand Challenges" that could provide the basis for a funded R&D initiative, and that involve significant integration across the rows of Figure 1.

Work on defining an appropriate set of Grand Challenge scenarios is underway, but to illustrate the idea, we shall outline one potential example. Although the example was developed from an automotive perspective, we suspect that it may apply (with perhaps minor modifications) in other industry sectors.

1) Example Challenge Scenario: Hybrid Emulation for Reconfigurable Automotive Body Shop

Jim is a Body Manufacturing Engineer for General Motors. Last year, GM launched a new product in at the plant in Arlington Texas. Jim was responsible for the design/configuration of all of the weld guns, clamps, and fixtures in the body shop, as well as the robot programs. In the past, this work would have been done by several groups/engineers. However, using a new generation of computer tools, Jim was able to optimally configure the body shop tooling from libraries of modular components, minimizing the number of tooling variations and maximizing flexibility, and he was able to develop and validate the robot programs completely in a virtual environment.

Today, from the web browser on his personal computer, Jim is able to monitor every aspect of the body shop in Arlington as it is running. He can navigate from the body shop, to any zone or line, to any workcell, or any robot or programmable logic controller (PLC). For example, if he is looking at a particular workcell, his screen displays a 3-D visualization of the robots doing the work. His screen looks remarkably like the one he was working with when he designed and validated the tooling and robot programs for the cell prior to production, only now, the robot positions and other information on his screen are being continuously updated by reading information from the plant floor network in real time, rather than by a simulation. He can pause the display at any time, and replay information from the past. Jim used this tool during the launch of the Arlington plant to "fine tune" the workcell to optimize performance and thruput of the line. He did this from his office in China where he was on special assignment at the time; he did not travel to Arlington during the launch of the plant.

Today, there is a problem at the plant. One of the robots in the workcell has gone down, and it will take four hours to repair. Jim receives an urgent message on his Blackberry, and immediately acknowledges the message, and launches the operations monitor to investigate from his office in Michigan. (Had Jim not been available, this message and the work described below could have been performed at any one of GM's Body Manufacturing Engineering centers globally. Jim is the first choice for the work because he is so familiar with the operation.)

Although the workcell is down, Jim can replay exactly what was happening in the workcell prior to the failure. From his screen, he selects all of the weld points that the failed robot was responsible for, and he selects several other robots, and requests each one to report whether or not they are capable of performing any of the welds that had been assigned to the robot requiring repair. He finds that it would be feasible to continue operations by temporarily reassigning weld points to other robots.

He selects the points to be reassigned and the robots to which they will be assigned. Each robot integrates the new points into their respective programs. Jim then simulates the operation of the line. Now, for the robots, weld controllers, and PLCs whose programs are being modified, the motions and information are being provided by a simulation tool; for the rest of the line, the motions and information are being provided by re-playing the real-time data collected before the line went down. In this way, Jim is able to verify that the new programs will work properly without interferences. He is able to make whatever modifications are necessary to insure proper operation. In fact, this is exactly the type of work Jim was doing when the plant was initially launched and he was fine tuning the operation.

When he is satisfied, he "releases" the modified programs to the robots, weld controllers, and PLCs, and informs the plant personnel that they may re-start the line with the new programs. It has taken Jim 15 minutes to complete this work. Later that day, when the robot has been repaired, plant personnel momentarily stop the line, restore the programs to their prior state, and resume normal operation. In the interim, the line is operating at 80 % of its normal thruput, and 20 minutes of production have been lost. In the past, 4 hours of production would have been lost while the failed robot is repaired.

V. CONCLUSIONS

Globalization has radically changed the nature of manufacturing, and Smart Assembly is about re-inventing assembly processes to succeed in this new environment. If successfully realized, Smart Assembly may be a key aspect of future, thriving manufacturing enterprises. Efforts are currently underway to increase awareness of Smart Assembly, through the creation of a Smart Assembly working group, which is hoping to refine the vision, score, business case scenarios, and roadmaps for what is hoped to ultimately be a national Smart Assembly activity.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of all of the participants in the NIST sponsored Smart Assembly Workshop conducted on October 3-4, 2007 (see http://smartassembly.wikispaces.com/) and specifically the following individuals who contributed substantially to the content of this report:

From the Manufacturing Engineering Laboratory at NIST: Dale Hall, Steve Ray, Al Wavering, Kevin Jurrens, Alkan Donmez, Fred Proctor, Rob Ivester, Ranchuri Sudarsan, and numerous other team members.

From the Manufacturing Systems Laboratory at GM Research: Roland Menassa, Stephan Biller, Jon Owen, Jeff Alden, Jeff Tew and members of their groups.

From ARC Advisory Group: Jim Caie,
From the Integrated Manufacturing Technology Initiative: Richard Neil,
From Oneida: Allan Martel,
From Rolls-Royce: Parker Sykes.

REFERENCES

[1] Draft workshop report is available at http://smartassembly.wikispaces.com/
[2] "Manufacturing in America", US Department of Commerce, 2004
[3] NAM, based on US Bureau of Labor Statistics, US Bureau of Economic Analysis, US Census Bureau

Science based Information Metrology for Engineering Informatics

Sudarsan Rachuri
George Washington University &
Design and Process Group
Manufacturing System Integration Division
National Institute of Standards and Technology, USA
sudarsan@cme.nist.ogv

Abstract— Engineering informatics is the discipline of creating, codifying (structure and behavior that is syntax and semantics), exchanging (interactions and sharing), processing (decision making), storing and retrieving (archive and access) the digital objects that characterize the cross-disciplinary domains of engineering discourse. It is absolutely critical that a sharing mechanism should preserve correctness (semantics), be efficient (for example, representation, storage and retrieval, interface), inexpensive (for example, resources, cost, time), and secure. In order to create such a sharing mechanism, we need a science-based approach for understanding significant relationships among the concepts and consistent standards, measurements, and specifications. To develop this science, it is essential to understand the interactions among the theory of languages, representation theory, and domain theory. Creating the science of information metrology will require a fundamental and formal approach to metrology, measurement methods and testing and validation similar to the physical sciences.

Keywords: *Engineering informatics, product lifecycle, standards, interoperability, metrics, semantics*

I. INTRODUCTION

A prerequisite for competitive advantage in manufacturing is a good and sustained investment in Engineering Informatics to describe a common product description that is shared among all stakeholders throughout the lifecycle of the product. Informatics is a conceptual synthesis of mathematics, computing science, and applications as implemented by information technology. Engineering informatics is the discipline of creating, codifying (structure and behavior that is syntax and semantics), exchanging (interactions and sharing), processing (decision making), storing and retrieving (archive and access) the digital objects that characterize the cross-disciplinary domains of engineering discourse. This is a relatively hard problem, as it requires combining a diverse set of emerging theories and technologies: namely, information science, information technology and product engineering, and many different cross-disciplinary domains.

The environment in which products are designed and produced is constantly changing, requiring timely identification and communication of failures, anomalies, changes in technology and other important influences. For such an adaptable organization to function, an information infrastructure that supports well-defined information exchange processes among the participants is critical. The IT industry that supplies engineering informatics support systems is currently vertically integrated. Vertically integrated support systems do not provide for opportunity of full diffusion of new innovations across the entire community of users. A study of engineering informatics support provided by a representative set of major software vendors shows that the availability of support tools is partial and incomplete. Some vendors cover several areas, while there are areas that are poorly covered or not covered at all by any vendor. Relying on a single vendor to cover all areas of support for engineering informatics would not provide the kind of innovation needed by the customers. There is a lack of interoperability across tools and that there are barriers to entry for software developers that could provide a plug and play approach to engineering informatics support. Currently only a few IT companies with vertically integrated tool sets are able to provide facilities that are even partially integrated.

The Product Lifecycle Management (PLM) concept holds the promise of seamlessly integrating all the information produced throughout all phases of a product's life cycle to everyone in an organization at every managerial and technical level, along with key suppliers and customers. PLM systems are tools that implement the PLM concept. As such, they need the capability to serve up the information referred to above, and they need to ensure the cohesion and traceability of product data.

A critical aspect of PLM systems is their product information modeling architecture [1]. Here, the traditional hierarchical approach to building software tools presents a serious potential pitfall: if PLM systems continue to access product information via Product Data Management (PDM) systems which, in turn, obtain geometric descriptions from Computer-Aided Design (CAD) systems, the information that becomes available will only be that which is supported by these latter systems.

II. PRODUCT REPRESENRATION AND INTEROPERABILITY

Interoperability is pervasive problem in today's information systems and the cost of the problem of managing interoperability is a major economic drain to most industries. The problem of supporting interoperability requires the development of standards through which different systems would communicate with each other. These standards vary from purely syntactical standards to standards for representing the semantics of the information being exchanged. However, for multiple systems to interoperate these systems will have to be tested for conformance, implementation and inter-operability among each other. These tests will have to encompass syntactic, content and semantic aspects of exchange between these systems.

A standardized exchange behavior within a specified set of conventions has a form (syntax), function (scope) and the ability to convey as unambiguously as possible an interpretation (semantics) when transferred from one participant to the other. The design of a standardized exchange in the context of information metrology is dictated by:

1. Language: the symbols, conventions and rules for encoding content with known expressiveness. Examples include First Order Logic [2], Knowledge Representation [2], OWL [3], UML [4], SysML [5], and EXPRESS [6].
2. Processible Expressiveness: the degree to which a language mechanism supports machine understanding or semantic interpretation. Expressiveness is closely connected to the scope of the content that can be expressed and to the precision associated with that content. Support of standardized exchange requires a set of complementary and interoperable standards.
3. Content: the information to be communicated. Content includes the model of information in the domain and the instances in the domain and explicates the relationship between the message and the behavior it intends to elicit from the recipient. Examples of content, include Standard for the Exchange of Product model data (STEP) [7], NIST Core Product Model (CPM) [8] and its extensions, the Open Assembly Model (OAM) [9], the Design-Analysis Integration model (DAIM) and the Product Family Evolution Model (PFEM).
4. Interface: User interface concerns efficiency of communication between the system and humans. Software interface concerns accurateness and completeness of communication between systems.

III. LONG TERM KNOWLEDGE RETENTION AND ARCHIVAL

These digital objects in engineering need to be preserved and shared in a collaborative and secure manner across the global enterprise and its extended value chain. The problem of digital preservation is very complex and open-ended (dynamic situations or scenarios that allow the individual users to determine the outcome). To understand the problem of digital archiving we need to define a taxonomy of usage scenarios as an initial guide to categorize different end-user access scenarios. The scenarios, which we call the "three Rs", are: (i) reference, (ii) reuse and (iii) rationale. The primary driver for the above categorization is the special retrieval needs for each of these scenarios. For example a collection intended primarily for reference may need to be organized differently than one intended for reuse, where not only the geometric aspects of the product are sought but also other information regarding manufacturing, part performance, assembly and other aspects. In a similar vein, rationale information may have to be packaged differently in that it may include requirements information along with other performance data on the part or the assembly. Given the range of uses and perspectives of the end-users will have large impact on the process of archiving and retrieval.

IV. SCIENCE-BASED INFORMATION METROLOGY

It is absolutely critical that a sharing mechanism should preserve correctness (semantics), be efficient (for example, representation, storage and retrieval, interface), inexpensive (for example, resources, cost, time), and secure (such as Role-Based Access Control). In order to create such a sharing mechanism, we need a science-based approach for understanding significant relationships among the concepts and consistent standards, measurements, and specifications. To develop this science, it is essential to understand the interactions among the theory of languages, representation theory, and domain theory. Creating the science of information metrology will require a fundamental and formal approach to metrology, measurement methods and testing and validation similar to the physical sciences. The effort involved will be cross-disciplinary in nature because 1) supply chain and engineering informatics are complex endeavors involving artifacts in several business areas 2) the industry does not have an established interoperability testing approach at the semantic level and 3) testing can consume a lot of time and there is no clear methodology to suggest what kinds of testing are essential.

Even though several disparate attempts were made in the past to understand this problem, the primary reasons that we can succeed now are: (1) The new sets of technical ideas that have emerged like the semantic web technologies, (2) better collaborative tools, and new models of software and standards development that have become dominant in the IT world, and (3) advanced mathematics, computer science, and logic-based systems. Besides, these developments the IT industry in large parts is moving away from products with only proprietary software components to a mixture of open source and proprietary components. The awareness and benefit of open source models are being embraced by large parts of the industry. The timing and attitude towards open standards and open source models have gained currency. The primary reason for the call for open standards is the current environment in the IT industry and the rise of the global network-based manufacturing. Both economic efficiency of global firms and design and manufacturing capabilities of these firms in the future will depend on the smooth functioning of the design and manufacturing information network especially for the small and medium enterprises (SMEs) to take advantage of the global and local markets.

V. CONCLUSIONS

The potential impacts of information metrology for engineering informatics include: (1) assistance to manufacturing industry end users and software vendors in ensuring conformance to information exchange standards; (2) creation of science of information metrology [10], (3) development of a fundamental and formal approach to information metrology, and (4) measurement, testing and validation methods similar to the physical sciences.

Disclaimer: No approval or endorsement of any commercial product by the National Institute of Standards and Technology or by the Syracuse University is intended or implied.

1. Sudarsan, R., Fenves, S. J., Sriram, R. D., and Wang, F., "A product information modeling framework for product lifecycle management," *Computer-Aided Design*, Vol. 37, No. 13, 2005, pp. 1399-1411.

2. Sowa, J. F., *Knowledge representation, Logical, Philosophical, and Computational Foundations*, Brooks/Cole1998.

3. Web Ontology Language (OWL). http://www.w3.org/2004/OWL/ . 2005.

4. OMG. UML 2.0 OCL Specification. http://www.omg.org/cgi-bin/doc?ptc/03-10-14 . 2004.

5. SysML - Open Source Specification Project. www.sysml.org . 2007.

6. Schenck, D., and Wilson, P. R., *Information*

modeling: the EXPRESS way, Oxford University Press, New York,1994.

7. Kemmerer, S., "STEP: The Grand Experience, (Editor)," NIST Special Publication 939, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA, 1999.

8. Fenves, S., Foufou, S., Bock, C., Bouillon, N., and Sriram, R. D., "CPM2: A Revised Core Product Model for Representing Design Information ," National Institute of Standards and Technology, NISTIR 7185, Gaithersburg, MD 20899, USA, 2004.

9. Sudarsan, R., Baysal, M. M., Roy, U., Foufou, S., Bock, C., Fenves, S. J., Subrahmanian, E., Lyons K.W, and Sriram, R. D., "Information models for product representation: core and assembly models," *International Journal of Product Development*, Vol. 2, No. 3, 2005, pp. 207-235.

10. Carnahan, L., Carver, G., Gray, M., Hogan, M., Hopp, T., Horlick, J., Lyon, G., and Messina, E., "Metrology for Information Technology (IT)," National Institute of Standards and Technology, Gaithersburg, MD 20899, USA, NISTIR 6025, 1997.

Evaluating Manufacturing Machine Control Language Standards: An Implementer’s View

Thomas R. Kramer
National Institute of Standards and Technology
MS8230
Gaithersburg, MD 20817, USA
kramer@cme.nist.gov

Abstract —The focus of this paper is: how can standards for manufacturing machine control languages be evaluated? What is required of a standard defining one of these languages so that implementations will interoperate? The paper provides a set of specific questions to ask about a control language standard. Reasons why the questions should be asked are given. Four machine control languages are used as examples:
EIA-274-D, BCL, DMIS, and STEP-NC.

Keywords: *control, language, machine, standard, AP 238, BCL, DMIS, EIA-274-D, ISO 10303, ISO 14649, STEP-NC*

I. INTRODUCTION

The focus of this paper is: how can standards for manufacturing machine control languages be evaluated? What is required of a standard defining one of these languages so that implementations will interoperate?

The paper deals only with languages intended to be used in control program files. There are also manufacturing machine control languages (such as the I++ DME Interface Specification and DMIS Part 2) intended to be used for transmitting individual commands one at a time. The issues for those languages are similar, but they are not addressed here.

In this paper, “manufacturing machines” means things such as machining centers, turning centers, and coordinate measuring machines that are run by computer numerical control systems.

The interfaces served by the standards are those between program generators and program execution systems. The programs that travel over these interfaces need to be capable of exercising the full functionality required by the task at hand (in general, all or almost all of the functionality of the receiving system). Hence, a manufacturing machine control language must provide a suite of program statements or commands that exercise that functionality.

The systems on the two sides of the interface must interoperate in the sense that a program passing across the interface must be executable by the receiving system and must do what the generator of the program intended it to do. Moreover, the intended meaning must be as described in the standard.

A standard for a language should describe how program statements should be executed in enough detail that if the standard is implemented in a number of sending systems and a number of receiving systems and all systems conform to the standard, any receiving system will do what is intended in any program generated by any sending system.

II. WHY AN IMPLEMENTER’S VIEW?

An implementer is a person who programs the software on one side or the other of an interface so that the software either generates programs or reads and executes programs. Because building an implementation requires understanding all the details of a language, even the most minute, implementers are the people in the best position to judge whether a standard is complete and unambiguous. Standards are usually written with the implementer as the primary audience. One World Wide Web Consortium web page, for example, says “Specifications are aimed at people writing software to implement them” [15].

The implementer’s view of a control language standard is different from the standard writer’s view in the same way that a file reader is different from a file generator. The standard writer has concepts in mind and puts down statements in the formal language used to define the standard, statements in natural language, and diagrams to represent those concepts. The implementer reads the formal language, natural language, and diagrams and forms concepts from them. Just as parsing from characters into structures is much more difficult than generating character strings from structures, so implementing a standard is much more difficult than writing it.

In implementing a standard, the implementer is continually asking, “Is there more than one way in which this can reasonably be interpreted?”, and “Do I have to make some assumption about the meaning in order to implement it?” If the answer to either question is yes, the implementer can be confident that some other implementer will make the other choice or assumption, so that the other implementation will not interoperate.

III. EXAMPLES OF STANDARDS FOR MANUFACTURING MACHINE CONTROL LANGUAGES

The following manufacturing machine control language standards will be used as examples. The author has direct experience with all of them by having built an implementation and/or studied the standard and submitted detailed comments to the committee responsible for the standard.

A. EIA-274-D

EIA-274-D, dated February 1979, [4] is a standard of the Electronics Industry Association (EIA) and is a low-level language designed for execution on the controller of a machining center or turning center. Another name for it is RS-274-D, since it is also an ANSI standard under that name. It is informally called a "G and M code" standard, since it consists primarily of codes starting with G or M. In this respect and in meaning, it is similar to ISO 6983.

B. BCL

BCL [13] is a low-level language designed for execution on the controller of a machining center or turning center. It is a language whose acronym outlived its original name. The name started in 1983 as "Binary CL" (EIA 494 A). The "CL" probably stood for Cutter Location. The standard did not say what CL stood for. By the February 1997 proposal for EIA 494 C, BCL had changed into "Basic Control Language". The language itself changed also over that period from a terse gobbledegook of letters and digits into human-readable abbreviated English command names accompanied by parameter values having primitive data types (keyword, string, number, etc.), all represented using ASCII (American Standard Code for Information Interchange) characters.

C. 3. The STEP-NC milling family: ISO 14649 and STEP AP238

ISO is the International Organization for Standardization. STEP is the STandard for the Exchange of Product model data. ISO 14649 and STEP AP 238 are high-level languages for various types of numerically controlled machines. These standards are still being developed. The most mature parts (subdivisions of a standard) are applicable to machining, specifically Parts 10 and 11 of ISO 14649 [8], [9]. Only those two parts are discussed in this paper.

AP 238 [7] is largely a recasting of ISO 14649 semantics into the terms of the STEP "integrated generic resources" so that machine control programs may be processed (to a certain extent) by any system that can handle the STEP integrated generic resources. STEP itself is a series of several dozen parts designed for product data representation and exchange. All the STEP parts are part of ISO 10303. AP 238 encompasses multiple parts of ISO 14649. Only the portion of AP 238 relevant to machining centers is covered here.

These languages are "high-level" in the sense that they are designed to communicate geometry, machining operation data, and machining strategies and to leave other decisions (the generation of toolpaths, in particular) to the controller. However, they also include facilities for sending low-level commands that give tool paths in detail.

D. DMIS (Part 1)

DMIS (Dimensional Measuring Interface Standard) is a mid-level language for programs for coordinate measuring machines (CMMs) and other dimensional measuring equipment [2]. Since it must do numerical data analysis while it executes, DMIS requires much more complex execution software than do the EIA-274-D and BCL languages, but it does not require the use of strategies that STEP-NC requires. DMIS has been updated through five versions, starting in 1986. DMIS 5.0 is the most recent standard.

DMIS defines both a programming language (the input format) and a format for data reporting (the output format). Only the programming language is covered in this paper. There is a DMIS Part 2, which is an object interface specification. It is not further discussed here.

IV. QUESTIONS FOR EVALUATING A MANUFACTURING MACHINE CONTROL LANGUAGE STANDARD

To evaluate a manufacturing machine control language standard, the following questions should be asked.

A. Is the standard complete for the intended use?

B. Is the standard clear and unambiguous?

C. Is the standard defined using a high-level information modeling language for which processors (readers and code generators) are readily available?

D. If the standard is defined using a high-level information modeling language, is there a well-defined file representation that works with the high-level language?

E. If the standard is defined directly as a file format (i.e. not by using high-level language), is the method used to define the file format clear and unambiguous?

F. Is special software required for reading and writing program files or for assembling the file data into meaningful structures? If so, is it widely available, and is it free or costly?

G. Has the standard been tested? How? What were the results?

H. Is there a continuing committee devoted to maintaining the standard? What is the committee's track record of dealing with proposed changes?

I. Are there intellectual property issues that may make using the standard impossible or expensive in the future?

J. Is there a critical mass of conforming implementations of the standard? Does it appear there will be a critical mass in the future?

K. Does the standard have conformance classes? Are they part of the standard?

L. Is it necessary to follow a set of usage rules additional to the standard in order to build an interoperating implementation?

M. Are there mechanisms in place (formal or "natural") to insure that implementations conform to the standard?

V. DISCUSSION OF THE QUESTIONS

How much weight to give to the various questions depends on one's point of view. Most of the people to whom the performance of a manufacturing machine control language standard is important are in one of two groups: (A) end users trying to decide whether to acquire and use a system that implements the standard, or (B) systems developers trying to decide whether to implement the standard (particularly those working for systems vendors or for users building their own systems). End users who want to buy a system rather than to build one, for example, may not care how hard it is to build an implementation. A person building an implementation, as another example, may not care whether there are other implementations of the same sort.

The discussion of examples in this section reflects the opinion of only the author.

A. Is the standard complete for the intended use?

From a user's point of view, in order to determine whether a standard is complete enough, the user should make a list of the required machine functions and then determine whether the standard supports those functions. This may be done by studying the standard, by getting information from users of the standard, by observing conforming implementations in action, or by some combination of those.

There is no global answer to this question because the meaning of "the intended use" is dependent on who is doing the intending. There are no fixed boundaries on the set of people who might be users of a standard.

All of the examples are complete for simple use on machines of the sort for which they were originally developed, but all of them could be extended for more advanced functionality or for control of similar but different machines.

As an example of additional functionality on a target machine, EIA-274-D, which was designed to be used on machining centers, specifies that codes G36 to G39 are "permanently unassigned" (meaning revisions of the standard should not use them) and available for "individual use". Kearney and Trecker built a machining center with a broken tool detector, and extended EIA-274-D by using G38 to operate the detector [11].

As an example of a machine that falls a little out of the ballpark from the original target machines, DMIS was designed for doing dimensional measurements on a coordinate measuring machine using a touch probe, so DMIS has a "measure a point" command. Dimensional measurement can be done using a theodolite, but a point cannot be located all at once with a theodolite. It is necessary to take at least two measurements of angles to the same point and then calculate its location. DMIS does not provide a command to do that.

B. Is the standard clear and unambiguous?

For clarity and unambiguousness, there are two largely separable areas requiring attention: syntax and semantics. Syntax covers what tokens (i.e words, numbers, and special symbols), statements, and sequences of statements can legally be written. Semantics covers what a token, statement, or sequence of statements means.

Modern formal languages exist (EBNF, for example) that make it possible to specify the syntax of a control language very precisely. It is also possible to be precise about syntax in natural language, but that is more difficult.

There are no formal languages that make it possible to specify semantics. Only natural language and diagrams are available for conveying the meaning of a standard, and it is very difficult to specify semantics by these methods.

The level of being clear and unambiguous of most of the machine control standards the author has seen is not very high.

Ambiguity in a standard may be intentional or unintentional, but in either case ambiguity defeats interoperability. EIA-274-D, for example, is filled with intentional ambiguity. Appendixes A.1, A.2, and A.3 provide that each implementer can specify a host of things (such as how numbers can be written and whether dimension values are absolute or incremental) that need to be agreed between a program generator and a program executor.

STEP NC contains many instances of unintentional ambiguity (for example, the location of most open profiles is undefined). NIST has submitted 153 suggestions for technical changes in Part 10 of ISO14649 and 70 for Part 11. Many of these suggestions aim to eliminate ambiguity.

C. Is the standard defined using a high-level information modeling language for which processors (readers and code generators) are readily available?

Standards that are defined using a high-level information modeling language have a large advantage over those that are not. Examples of high-level information modeling languages include:

- EXPRESS (not an acronym) — developed as part of STEP [5],
- XML Schema (Extensible Markup Language Schema) developed by the World Wide Web Consortium [14].

These languages all contain primitive data types (integers, strings, lists, etc.) and provide for defining the sorts of interlinked data structures that are needed in a machine control language.

When a control language standard is written using one of these high-level languages, a certain amount of automatic processing may be done. Software is available that will read the file defining the control language, check its syntax, and generate source code in a computer language. The source code defines computer language structures corresponding to the structures in the control language and contains functions for accessing (extracting and inserting) the data in those structures. If a machine control language standard is written using a high-level language, the standard will probably have been checked for good syntax using an automatic checker, and a potential user of the standard who has a checker can use it to check the standard.

If a machine control language standard is written using a standard lower-level formal language such as EBNF (Extended Backus Naur Form) [10], it can be checked for syntax automatically, but utilities for generating computer code are not readily available. Moreover, since the lower level formal languages do not define structures, there is not enough information in the control language definition file to produce code useful for building an implementation. Only structures that mirror the syntax can readily be built by an automatic system working from EBNF, and the syntax structure is not likely to be the structure an implementer would like to use.

If a machine control language is written using an ad hoc description method, neither automatic syntax checking nor automatic code generation is feasible.

Since the semantics of a machine control language standard cannot currently be described in a formal language, it is never possible to generate an implementation automatically that does anything more than read program files, rewrite program files, allow browsing, and generate statistics, even if a high-level language has been used to define the control language.

D. If the standard is defined using a high-level information modeling language, is there a well-defined file representation that works with the high-level language?

With a high-level information modeling language, it is feasible to define a generic file format that combines with any information model defined in the language so that a specific file format exists for the model. Thus, when a manufacturing machine control language is modeled using the high-level information modeling language, general-purpose software designed to be used with the high-level language will read or write a file containing an executable machine control program without any work on the part of the implementer other than making a single library function call in a program. This saves an enormous amount of work an implementer would otherwise have to do and makes it much less likely that the reader or writer will not conform to the standard. Of course, when writing, a model of the program must be built before a "write this model to a file" function can be called.

EXPRESS and XML Schema have well-defined generic file formats of the sort just described. EXPRESS works with the STEP Part 21 format [6], while XML Schema works with XML [1]. A high-level language can work with more than one file format. An EXPRESS model can be used with XML, for example.

High-level information modeling languages generally are also built to support implementations that use databases (or persistent objects) and application programming interfaces rather than files for exchanges across an interface, but this paper does not deal with that. It is not common for stored machine control programs to be implemented that way.

The STEP-NC standards are all modeled using EXPRESS. None of the other three examples uses a high-level information modeling language.

E. If the standard is defined directly as a file format (i.e. not by using high-level language), is the method used to define the file format clear and unambiguous?

EIA-274-D uses English to define the file format. The English descriptions are generally hard to follow. There are several unintentional ambiguities and many intentional ones.

Section 3 of BCL (as proposed for EIA 494 C) defines overall file structure in English. Section 4 defines in English what the fields of a BCL record (a single statement) may be and what characters constitute a valid field (such as a text field, parameter separator field, or numerical field). Most of the descriptions in Sections 3 and 4 are clear and unambiguous, but the definition of "numerical field" is ambiguous. Sections 8.0.1 and 8.0.2 of BCL define a higher level syntax notation for defining what sequences of fields constitute valid commands. The higher level notation is clear and unambiguous and is used consistently in the succeeding parts of the standard.

DMIS 5.0 defines its file format two ways. First, much of Section 5 describes syntax in English, and a syntax notation defined briefly at the beginning of Section 6 is used in the remainder of Section 6 (over 400 pages) to define what sequences of fields constitute valid commands. Second, Annex C gives a definition of DMIS syntax in EBNF (although the lowest level, such as what sequence of characters makes a real number, is omitted). The file format of DMIS is thus generally clear and unambiguous. The use of EBNF has enabled the automatic construction of DMIS input file syntax checkers [12].

F. Is special software required for reading and writing program files or for assembling the file data into meaningful structures? If so, is it widely available, and is it free or costly?

In all four examples, there are at least two levels of encoding. All of the examples use ASCII code at the lower level to interpret bits as characters. Managing this level takes no special software. Every common programming language reads and writes ASCII. At the upper level (where reading means to convert a stream of ASCII characters into meaningful structures and writing means to convert structures into a character stream) all the examples except STEP-NC require special software for reading and writing. Typically:

• data structures must be designed,

• a parser must be built to receive a character stream and build and populate a hierarchy of structures, and

• a writer must be built to traverse a hierarchy of structures and generate a character stream.

In STEP-NC, ISO 14649 EXPRESS models can be used directly with STEP Part 21, and no special software is needed for reading and writing files beyond that which can be generated automatically. AP 238, however introduces a third level of encoding. Special software is needed not for reading and writing but for dealing with this third level of encoding. Most of the data in AP 238 is encoded at a level between a character stream and structures meaningful to an application programmer. The middle level is built in terms of entities from the STEP integrated resources, and Part 21 files contain representations of these structures. The structures in this middle level are utterly unintelligible to programmers conversant with machine control. In order to use AP 238 it is currently necessary to have special software that either (1) converts the integrated resources structures into structures like those that may be created directly from ISO 14649 and provides access functions for the 14649-like structures or (2) provides access functions for the integrated resources structures with semantics similar to those that may be created directly from ISO 14649. Currently, only the second method has been implemented, and there is only one provider of this type of special software. Building software of this sort has a high and steep learning curve.

G. Has the standard been tested? How? What were the results?

A manufacturing standard is like a piece of complex software. As with software, mistakes may be made in syntax or logic, and the functionality may not be what is intended by the authors. There is no chance that complex software will be bug free as it comes from the programmer. It must be compiled, tested, and debugged before release. There may be bugs in syntax, bugs in operation (writing beyond the end of an array, for example), and bugs in what the program does. This is universally acknowledged. Commercial software houses always have testing procedures in place. As with software, there is no chance that a complex manufacturing machine control language standard will be bug free as it comes from the authors. It should be implemented, tested, and debugged before final release. This is rarely acknowledged. Most standards development organizations do not have standards testing methods in place that must be applied before a proposed standard may be approved. STEP has testing and conformance procedures, but they are too little and too late to insure that a standard is of high quality at the time of first release.

Computer languages have compilers that make executables that can be tested. Standards do not. The best that can currently be done automatically with a standard, if it is defined using a high-level language, is to build a system that can read program files, rewrite program files, allow browsing, and generate statistics.

EIA-274-D allows so many choices (i.e. is so ambiguous) that only testing one of the many billions of legal variants is feasible. The extent to which a variant has been tested is dependent on the creator or vendor of the variant.

BCL appears to have been well-tested by the large organizations that used it, in close collaboration with the vendor that provided the implementations. If it had been widely implemented (correctly), it could have provided a high degree of interoperability.

There has been no formal testing program for DMIS. Some vendors appear to have implemented DMIS in conformance with the standard and tested carefully. Other vendors have not.

STEP-NC has been tested to a modest extent, but it is far from being fully tested. There are enough ambiguities in the standard that the notion of “conforming implementation” is tenuous. Further implementation tests are under way.

H. Is there a continuing committee devoted to maintaining the standard? What is the committee’s track record of dealing with proposed changes?

Technology advances rapidly so that additional functionality is needed periodically in any standard dealing with machine control. If a standard is not updated when new technology appears, implementers will extend the language in non-standard ways in order to use the technology.

Even if new technology does not appear, machine control language standards are so complex that it takes years to eliminate all the ambiguities and bugs.

To accomplish updating a standard, it is necessary to have a group in place that understands the standard and can judge the merits of proposed changes. Determining what updates are needed works most smoothly if there is an established, well documented process for updating the standard. The process should include consideration of requests from anyone for changes.

Of the examples, DMIS has a very good method of handling updates, STEP-NC is just getting started on its first round of updates, and EIA-274-D and BCL appear to have no currently active group.

In the DMIS system, a web site is open for Standard Improvement Requests (SIRs) from anyone [3]. Each request is logged and proceeds through several status states until consideration is complete. The web site shows the disposition of all of the hundreds of SIRs proposed since 1996. Once a SIR is entered in the system, anyone can submit a statement for or against the suggested change. No formal system will force the group in control of a standard be open to suggestions for change. In the case of DMIS, the group (now the DMIS Standards Committee) has been open to change and appears to treat all suggestions fairly. Other groups for other standards are often said to be less open and fair.

I. Are there intellectual property issues that may make using the standard impossible or expensive in the future?

Intellectual property right problems related to standards are not unusual. Potential users of a standard should look out for existing and foreseeable problems.

It is possible to get intellectual property rights (patents and copyrights) related to standards. If rights are granted, the owner may try to prevent others from using a standard or try to charge a fee for using it. A ploy the owner might use is to allow inexpensive usage at first and later, once the user has a substantial investment in using the standard, increase the fees. With patents, the owners rights are likely to be unclear, so that users may become involved in expensive litigation.

In 1993, U. S. Patent 5198990 was issued in which a claim was allowed for executing DMIS directly on a control system. The point of having a control language is to execute it, the idea of doing so directly is completely obvious, and patenting the obvious is not supposed to occur. Thus, it is disheartening that the U.S. Patent Office allowed the claim. The effect of the patent is said to have been that no one implemented DMIS for a few years. Eventually, it is said, an agreement was reached that the patent rights would not be used, and DMIS came into common use.

In 2004, U.S. Patent 6795749 was issued for a method of using ISO 14649. It is possible that this may have a chilling effect on the implementation of ISO 14649.

J. Is there a critical mass of conforming implementations of the standard? Does it appear there will be a critical mass in the future?

There need to be enough systems on each side of the interface that useful work can be done.

With EIA-274-D the very notion of conformance fails because the standard is so ambiguous. There are dozens of dialects of the language. Most computer aided manufacturing (CAM) systems have many different post-processors so that they will produce files in most dialects. Thus, it is feasible to use EIA-274-D, but programs are not portable from one machine to another except in some cases when the same company built both controllers. To be fair, EIA-274-D apparently never intended to support interoperability. EIA-274-D is analogous to the concept of "romance language" in that a speaker of one romance language will have a much easier time learning another romance language than will a speaker of Chinese or Swahili.

BCL is, perhaps, the saddest case. BCL is the clearest and least ambiguous of the standard languages for milling machines. Its usefulness, including the portability of programs, was proven by implementations in a few large installations (Rock Island Arsenal, in particular) but there are currently no known commercially available implementations. It seems to have died out. The better mousetrap did not make it in the marketplace.

STEP-NC - Commercially available implementations do not yet exist. There may or may not be a critical mass of implementations in the future.

DMIS - There are said to be several commercially available conforming implementations, but there are also said to be several commercial implementations that purport to implement DMIS but do not conform. There are also commercially available packages that include "DMIS" in their names but are not DMIS and do not claim to be. It is a "buyer beware" situation.

K. Does the standard have conformance classes? Are they part of the standard?

A conformance class is a subset of the specifications of a standard that is approved in some way for some type of use. For example, DMIS has prismatic and thin walled conformance classes. A conformance class may be defined by specifying which commands must be implemented and for each command, which parameters must be implemented.

There are at least three reasons to have conformance classes. First, for large languages, implementing the entire language may be beyond the capability of a vendor or the vendor may decide that it is not economically justified. Second, there may be some class of jobs which requires only a subset of the capabilities of the language. Third, there may be some set of machines which share a subset of the capabilities for which the language has commands.

If conformance classes have been defined for a standard but not incorporated in the standard itself, the status of the classes is in doubt (for example, it may not be clear under what circumstances the definition of the classes might change).

EIA-274-D does not define conformance classes.

DMIS defines two main conformance classes (prismatic and thin walled — meaning sheet metal) plus seven addenda for special capabilities such as rotary table and contact scanning. Moreover, there are three levels for each class and addendum. The DMIS conformance classes are not yet part of the standard.

BCL divides its commands into 32 groups called function sets. This is done in the standard. The intended use of the function sets is not described in text, but Appendix F, which suggests what the contents of machining process plans should be, says that a machining process plan should include a list of required function sets.

STEP-NC defines conformance classes for milling in section 5 of ISO 14649-11 and in section 6 of AP 238. These, however, are not the same. ISO 14649-11 defines six conformance classes by first dividing its entities into eight "data sets" (same idea as BCL's function sets) and then saying which combination of data sets must be included in which conformance class. AP 238 section 6 defines four conformance classes by providing a checklist of entities with a column for each class.

L. Is it necessary to follow a set of usage rules additional to the standard in order to build an interoperating implementation?

There may be communities of users of a standard that agree to follow a set of usage rules. In such communities, it is usually expected that if both the standard and the usage rules are followed, implementations will interoperate, but if the usage rules are not followed, implementations will not interoperate even if they conform to the standard. In some cases there is a fee to join the group, and the usage rules are not publicly available. Potential users of a standard should look out for this situation.

Usage rules may be desirable in several circumstances, such as:

- The standard is large and conformance classes have not been defined, so the rules serve to define a de facto conformance class.
- The standard is ambiguous.

It is much more desirable, however, to fix the standard so as to formalize the conformance classes and fix the bugs.

Of the examples, the author is aware of usage rules only in the case of AP 238 testing, and these rules do not seem to be intended to continue in the long run.

M. Are there mechanisms in place (formal or "natural") to insure that implementations conform to the standard?

Where there are many vendors on each side of a data interface and many users on only one side of a data exchange (readers and writers of HTML, for example), there is a "natural" mechanism for insuring that implementations conform. Any product that does not conform will not be used. No company can produce its own non-conforming flavor of HTML and coerce customers into using it.

Machine control languages never have the benefit of natural pressure for conformance. The markets are too small. Also, both sides of a machine control language interface (i.e., the programmer and the machine controller) are usually in the same customer company, so that the writer is able to adjust to whatever the reader expects. The customer rarely is able to insist on conformance to a standard in this situation. Vendors want to be able to claim to use a standard, but also want users to be unable to use products from other vendors. Thus, vendors may claim to implement a standard without actually conforming to the standard.

Thus, for machine control languages, formal procedures for ensuring conformance are needed in order to get conformance.

For machine control languages, conformance tests strict enough to ensure interoperability if passed are extremely difficult and time-consuming to devise. Once devised, they are difficult and time-consuming to apply. The details of this are enough to fill another paper.

The author is aware of no conformance mechanisms for EIA-274-D, and close conformance to the standard appears to be rare or non-existent.

BCL did not seem to have formal conformance mechanisms, but conformance (in the late 1990's) appeared to be excellent for two reasons. First, there was one primary vendor for BCL controllers. Second, the users were mostly large organizations (including the Rock Island Arsenal) with many machining centers who made the same parts many times and wanted to be able to use the same program on different machines.

For many years, DMIS did not have conformance requirements in the standard or conformance tests or conformance testing services. All manner of non-conforming implementations that claimed to use the standard came into existence. Commercial systems that were only generally similar to DMIS were built. Programs called DMIS programs were rarely interoperable between vendors. It became clear that something needed to be done to help achieve interoperability. In 2001, conformance requirements were included in DMIS 4.0. However, no conformance classes, conformance tests, or conformance testing services were defined at that time. Since then, conformance classes have been defined as discussed earlier, and modest conformance tests have been provided [12]. There is still no conformance testing service.

VI. CONCLUSION

This paper has presented thirteen questions the potential user of a machine control language standard might want to ask in order to decide whether to use the standard. Reasons why the questions should be asked have been given. As examples, partial answers to the questions have been provided for four machine control language standards.

REFERENCES

[1] Bray, T., et al., (editors), "Extensible Markup Language (XML) 1.0 Fourth Edition", World Wide Web Consortium, http://www.w3.org/TR/2006/REC-xml-20060816, 2006.

[2] Consortium for Advanced Manufacturing - International, "Dimensional Measuring Interface Standard Part I, Revision 05.0", Consortium for Advanced Manufacturing - International, 2004.

[3] Dimensional Metrology Standards Consortium, http:www.dmisstandard.org/content/blogsection/6/55, 2007.

[4] Electronic Industries Association, "EIA Standard EIA-274-D Interchangeable Variable Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines", EIA, 1979.

[5] International Organization for Standardization, "ISO International Standard 10303-11, Industrial automation systems and integration — Product data representation and exchange — Part 11: Description method: The EXPRESS language reference manual", International Organization for Standardization, 2003.

[6] International Organization for Standardization, "ISO International Standard 10303-21, Industrial automation systems and integration — Product data representation and exchange — Part 21: Clear text encoding of the exchange structure", International Organization for Standardization, 2002.

[7] International Organization for Standardization, "ISO International Standard 10303-238, Industrial automation systems and integration — Product data representation and exchange — Part 238: Application protocol: Application interpreted model for computerized numerical controllers", International Organization for Standardization, 2007.

[8] International Organization for Standardization, "ISO International Standard 14649-10, Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 10: General process data, second edition", International Organization for Standardization, 2004.

[9] International Organization for Standardization, "International Standard ISO 14649-11, Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 11: Process data for milling, second edition", International Organization for Standardization, 2004.

[10] International Organization for Standardization, "International Standard ISO/IEC 14977, Information technology — Syntactic metalanguage — Extended BNF", International Organization for Standardization, 1996.

[11] Kearney & Trecker Corporation, "Part Programming and Operating Manual, KT/CNC Control Type C", Pub 687D, Kearney & Trecker, 1979.

[12] National Institute of Standards and Technology, "DMIS Test Suite", http://www.isd.mel.nist.gov/projects/metrology_interoperability/dmis_test_suite.htm, 2007.

[13] Numerical Control BCL Standards Association, "NCBSA Standard Proposal for EIA 494 C, Basic Control Language (BCL) An ASCII Data Exchange Specification for Computer Numerical Control Manufacturing", Numerical Control BCL Standards Association, 1996.

[14] Walmsley, P. (editor), "XML Schema Part 0: Primer Second Edition", World Wide Web Consortium, http://www.w3.org/TR/2004/REC-xmlschema-0-20041028, 2004.

[15] World Wide Web Consortium, http://www.w3.org/XML/Core/#IPR, 2007.

Interoperability Testing for Shop Floor Measurement

Fred Proctor
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899
frederick.proctor@nist.gov

Bill Rippey
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899

John Horst, Joe Falco and Tom Kramer
NIST
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899

Abstract— Manufactured parts are typically measured to ensure quality. Measurement involves equipment and software from many different vendors, and interoperability is a major problem faced by manufacturers. The I++ Dimensional Measuring Equipment (DME) specification was developed to solve interoperability problems and enable seamless flow of information to and from dimensional metrology equipment. This paper describes validation testing of the I++ DME specification. The testing was intended to improve the specification and also to speed up its adoption by vendors. Testing issues are described, and a software test suite is detailed. Interoperability testing with real equipment was done over several years, and lessons learned from the testing will be presented. The paper concludes with recommendations for improving this type of testing.

Keywords: *interoperability, measurement, software testing*

I. Introduction

Automated geometric inspection of parts is done using coordinate measuring machines (CMMs). Traditionally, CMM vendors have sold a tightly-coupled software-hardware system for programming and controlling the inspection process. The last 15 years have seen large manufacturers acquire CMMs from many different vendors and endure the overhead of supporting multiple software applications. Further, 3rd party software vendors have been offering high quality products that often cannot be used because they are incompatible with some CMMs.

Automakers are major users of measurement equipment, and suffer from the cost and time to work around these incompatibilities. They have responded by supporting a specification for dimensional measurement equipment interoperability, called the I++ Dimensional Measuring Equipment Interface specification (I++ DME). The goal of I++ DME is to allow automakers, and any other manufacturers, to select the best software and equipment for their purposes and budgets and ensure that they work together seamlessly out of the box.

Specifications, like any result of a human endeavor, are never perfect and need to be tested (validated) to make sure they fulfill their requirements. For I++ DME, this means answering the questions, “Does I++ DME handle all of today’s measurement activities, or are important types of measurements or equipment left out? Is the specification written clearly and unambiguously, or will implementers have to make assumptions?” Likewise, products that claim to support I++ DME are never perfect and need to be tested (verified) to make sure they comply with the specification. This means answering the questions, “Does the product send only valid I++ DME messages? Does it respond appropriately to both valid and invalid messages?”

NIST has written an I++ DME test suite designed to help the specification writers make a better specification and the product vendors make better products. The test suite includes a simulated client that acts as the software that runs measurement plans, and a simulated server that acts as the equipment that makes the measurements. Test scripts cover all measurement activities, from startup through measurement and shutdown, including error conditions. A logging feature allows for later analysis of test results.

The I++ DME has undergone testing in a series of demonstrations involving real software and equipment at several important international quality technology expositions, including the 2004 International Manufacturing Technology Show (IMTS), the 2005 Quality Expo, and the 2005 – 2007 Control Shows. These multivendor demonstrations have included combinatorial testing of several software packages with several measurement machines. Comments from the participants, and their continuing participation, show that this level of testing rigor is valuable and helps to ensure quality products that meet customer requirements.

II. The measurement process

Before parts can be measured, they must be designed and at least partially manufactured. Design is normally done using computer-aided design (CAD) workstations that generate electronic design files that define the product requirements for subsequent downstream manufacturing operations. From the point of view of measurement, the design files contain dimensions and tolerances, and other requirements such as surface finish. A standard for the output of CAD information is ISO 10303, “Standard for the Exchange of Product Model Data,” also known as STEP [1]. STEP Application Protocol

(AP) 203 deals with design data; the second edition includes geometric dimensioning and tolerancing.

Although not part of the measurement process, computer-aided manufacturing (CAM) and computer- numerical control (CNC) are steps that define how the part is to be manufactured. It is worth noting that manufacturers would like to inspect as much as possible on the equipment used to manufacture the parts, in order to save the time it takes to move parts between equipment. Supporting this flexibility is one goal of interoperability specifications like I++ DME.

Given a part design, measurement plans are then developed which guide how specialized equipment or human experts are to inspect the part. A standard for the output of measurement planning is the Dimensional Measuring Interface Standard (DMIS) [2]. DMIS plans define the measurement sensors to be used (typically touch probes), features to be measured (such as surfaces and holes), and reports to be made.

Measurement plans are executed by software that connects to measurement equipment such as coordinate measuring machines. During this phase, commands are directed toward the equipment to select sensors, capture points of interest and return the results. Measurement plans may consist of thousands of individually acquired points, with coordinate systems set and branch points taken depending on intermediate results. The I++ DME specification covers the exchange of data between the execution software and the measurement equipment.

Once measurement data has been acquired, an analysis phase is performed in which the raw results are compared against the design requirements (e.g., dimensions and tolerances) so that quality conclusions can be made. A draft standard for reporting results is the Dimensional Markup Language (DML), being prepared by the Automation Industry Action Group.

While interoperability between these different phases of measurement is the overall goal, this paper focuses on validation testing of the I++ DME specification. The authors are conducting similar testing on STEP, DMIS and DML.

III. Challenges for standards-based measurement

A challenge for any standards-based activity is constraining the data exchange to a set that can be documented and thus standardized, while enabling vendors to innovate their products and thereby benefit manufacturers. For measurement, this challenge is made more difficult by the wide range of equipment used for measurement, and the many types of measurements done. For example, measurement equipment includes sensors such as touch-trigger probes, capacitance gages, lasers and other optical sensors; and machines ranging from small hand-moved portable arms through large granite-based fully automatic coordinate measuring machines. This technology continually evolves, and defining a set of capabilities to be used as the basis for a standard is difficult and requires compromise. In any case, there must be a process in place to revise the standard as technology improves and new sensors and measurement capabilities become available.

IV. The I++ DME Specification

The I++ committee is comprised of measurement equipment end users primarily from the automobile manufacturing sector. The I++ Dimensional Measuring Equipment (DME) specification [3] was written by I++ members and targeted toward equipment and software vendors. The goal was to enable manufacturers to pick best-in-class equipment and software reflecting their particular needs for sensor type, part size and measurement tasks.

I++ DME is a messaging protocol between measurement plan executors and measurement equipment. It uses TCP/IP sockets as the communication mechanism, and defines a message set and a client-server architecture. Clients are measurement plan executors, and servers are the equipment that carries out the measurements. For example, a client could read DMIS measurement plans produced by some upstream application, interpret the DMIS statements, send I++ DME messages to the measuring equipment, accumulate the measurement results that return as I++ DME messages from the server, and output a DMIS or DML measurement report. This is shown in Figure 1.

I++ DME consists of Unified Modeling Language (UML) descriptions of the messages, accompanied by natural language (English) that describes the semantics. Production rules in Backus-Naur Form (BNF) are provided that define the syntax of message composition. Numerous examples are provided as guidance to implementers. A sample I++ DME session is shown below, with messages from the client not underlined and responses from the server underlined.

```
00002 StartSession()
00002 &
00002 %
00003 GetDMEVersion()
00003 &
00003 # DMEVersion(1.4.2)
00003 %
00027 ChangeTool("ProbeB")
00027 &
00027 %
00078 SetProp(Tool.GoToPar.Speed(25.0))
00078 &
00078 %
00079 GoTo(X(2.626), Y(-4.656), Z(-4.100))
00079 &
00079 %
00094 PtMeas(X(2.47), Y(-4.13), Z(-5.10),
IJK(-0.01,-0.99,-0.00))
00094 &
00094 # X(2.44), Y(-4.64), Z(-5.99),
IJK(-0.019,-0.997,0.074)
00094 %
```

V. I++ DME Testing

As a product of a human endeavor, the I++ DME specification inevitably contains errors. The purpose of validation testing is to find the errors and suggest changes to the specification that fix the errors, before the specification is published and implementations are released. Validation ensures that the specification is complete, correct and

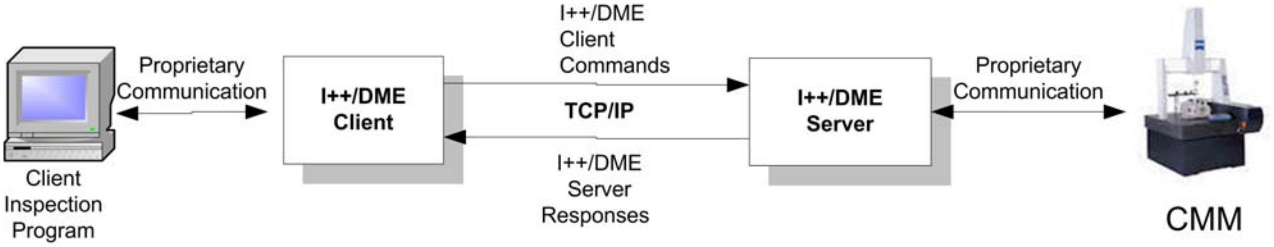


Fig. 1. The I++ DME activity model.

unambiguous. "Complete" means that it covers all the requirements set forth by the I++ members. Due to compromises, these may not completely satisfy the requirements of everyone. Nevertheless, it is the job of validation testing to discover any requirements that are not expressible in I++ DME. "Correct" means that there are no factual errors, including typographical errors but also inconsistencies in descriptions and conflicts with stated requirements. "Unambiguous" means that two readers of the specification will agree what is meant. This is difficult to achieve in practice, if for no other reason that the authors do not all speak the chosen natural language (English) as their native language. Ambiguity can be mitigated through the use of pictures or figures, and good examples.

Another objective of testing was to speed the commercialization of products that support I++ DME. This was achieved as a side effect of including vendors in the testing activities.

Testing can also lead to product conformance, if the testing tools persist after validation testing has concluded. In this case, all the hard work of testing can benefit newcomers, who can run the tests themselves privately and improve their products before releasing them.

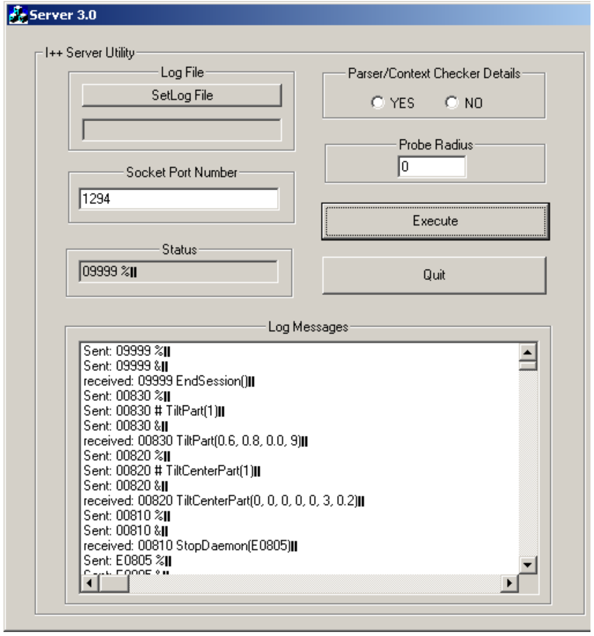
The approach to testing taken by the authors was to provide a software test suite that enables controlled, comprehensive testing, in source code, paired with a series of public interoperability tests and demonstrations at trade shows that included real products and real measurement tasks.

VI. The I++ DME Test Suite

The I++ DME Test Suite [4] was written by the authors as a utility to enable internal testing of conformance to the specification. It is comprised of two applications, a server and a client, many test scripts, and source code for a C++ class library and parsers that parse client and server messages. The source code is free and intended to help newcomers implement I++ DME without having to incur the tedium of developing message handling code.

Figure 2 shows the I++ Server Utility. The server simulates the response of measurement equipment to I++ commands, maintaining a coarse world model and simulation of a coordinate measuring machine and responding plausibly to requests from a client. Developers of client software typically use the Server Utility as a stand-in for real servers (e.g., coordinate measuring machines) that are expensive to obtain. Developers of client software can use the Server Utility to verify that their commands are valid, and to see what responses they should be prepared to receive. The Server first opens up a socket on a port specified by the user, and awaits connections from a client. Every message received or sent by the Server is logged, displayed in a window and written to a file. Some attributes of the simplified models are configurable, for example the radius of the probe.

Figure 3 shows the I++ Client Utility. The client simulates the actions of plan execution software, sending requests to the server to select sensors and measure attributes of the part, and collecting responses back for later analysis. Developers of server equipment typically use the Client Utility as a stand-in for execution software. This allows them to see what commands they are expected to handle, and to check that their responses are valid. The client connects to a running server on a socket specified by the user, who then loads a script file for reading and execution, similar to the excerpt shown below:



Probe Radius: 0

Socket Port Number: 1294

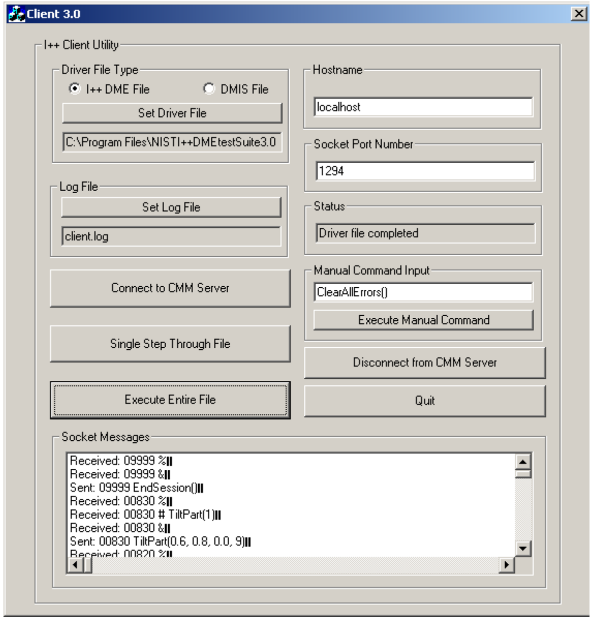
Status: 09999 %||

```
Sent: 09999 %||
Sent: 09999 &||
received: 09999 EndSession()||
Sent: 00830 %||
Sent: 00830 # TiltPart(1)||
Sent: 00830 &||
received: 00830 TiltPart(0.6, 0.8, 0.0, 9)||
Sent: 00820 %||
Sent: 00820 # TiltCenterPart(1)||
Sent: 00820 &||
received: 00820 TiltCenterPart(0, 0, 0, 0, 0, 3, 0.2)||
Sent: 00810 %||
Sent: 00810 &||
received: 00810 StopDaemon(E0805)||
Sent: E0805 %||
```

Fig. 2. The I++ DME Server Utility used is a surrogate for measuring equipment, used for testing client software.

```
AlignPart(1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 2.0)
AlignTool(0, 0, 1, 30)
CenterPart(2.0, 3.0, 4.0, 0.1)
ChangeTool("Probe1")
```

Each script file of I++ DME commands has an associated response file that is compared against what is received from the server. If responses don't match what is expected, errors are noted in the log file. These errors are not necessarily true errors, since the server messages in general include data points that vary depending on the actual sensed values of probe points. Strict comparisons against a pre-written response file may not match exactly yet still be valid. This is a challenge for automated testing, and one that requires balancing the difficulty of building an intelligent automated analysis tool against the value it provides, given that people will eventually be viewing the results and can be expected to make more difficult determinations of acceptability.



I++ Client Utility

Driver File Type: ● I++ DME File ○ DMIS File

C:\Program Files\NISTI++DMEtestSuite3.0

Log File: client.log

Hostname: localhost

Socket Port Number: 1294

Status: Driver file completed

Manual Command Input: ClearAllErrors()

Socket Messages

```
Received: 09999 %
Received: 09999 &
Sent: 09999 EndSession()
Received: 00830 %
Received: 00830 # TiltPart(1)
Received: 00830 &
Sent: 00830 TiltPart(0.6, 0.8, 0.0, 9)
Received: 00820 %
```

Fig. 3. The I++ DME Client Utility is a surrogate for measuring plan execution software, used for testing measuring equipment.

VII. Public Demonstrations

The I++ Test Suite allows developers to build compliant applications within their companies and test them before releasing them to their customers. At some point, applications will be run in production at customer facilities, and will interface with compliant applications from other vendors. It is important to have some experience with production interoperability prior to full release. This is the purpose of public demonstrations.

Three I++ public demonstrations have taken place, during the Control Shows in 2005, 2006 and 2007. The participants varied during each show, with the intent to include some number of client providers (e.g., measurement plan execution software developers) and some number of server providers (e.g., coordinate measuring machine builders). In 2007, the public demonstration included six clients and four servers, for 24 combinations possible for testing.

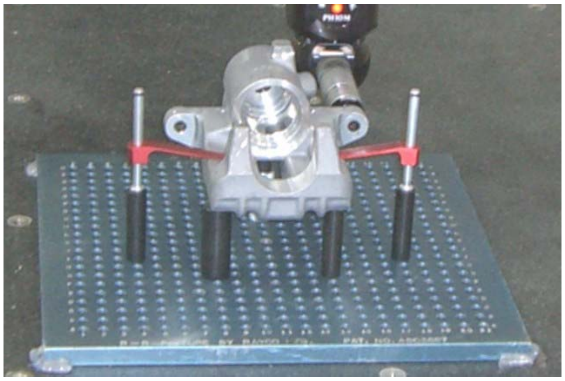


Fig. 4. Representative automobile part used for public demonstrations.

Unlike private testing with the I++ Test Suite, public demonstrations used real measurement plans (e.g., DMIS or some vendor proprietary plan formats) and real parts. A representative automobile part was selected, as shown in Figure 4. No test scripts were used, and thus no pre-written response files were written. Tests were done point-to-point, client-to-server, with people observing the measurement process on the machines and determining if the results of the measurement were acceptable.

The burden on the test judges was lessened somewhat by their experience with the test part. It was usually obvious when failures occurred, and where the source of the problem lay. If each test took place with a randomly-generated part, understanding what constitutes correct measurement would have been more difficult. The challenge is therefore to select a part with enough features to cover what is required by most manufacturers, simple enough to machine easily.

VIII. Recommendations

Practical experience with the I++ Test Suite and the series of public demonstrations has led to some recommendations for others who are undertaking similar validation efforts.

- Pre-testing components with simulated "mates" uncovers many simple errors that can be fixed early, saving time at the more expensive public demonstrations or installations on plant floors.
- Misinterpretation of specifications by people is to be expected. Formal methods of describing syntax and if possible semantics are preferred over natural language, especially when the audience members do not all speak the natural language natively.
- Examples should be provided where possible. Forgo the temptation to write all examples in the same style. For

example, if the specification allows variations in white space, examples should show this variation.

- Where the specification is ambiguous, expect that two developers will each interpret it differently. In cases where the resolution is a choice between two arbitrary options, each vendor will argue that their choice is the right one. There must be an arbiter whom all parties agree has the final word, and everyone must be prepared to go back to their benches and change.
- Standards validation is expensive, and should include line-by-line reading of the specification by experts; ongoing meetings to discuss revisions to the specification; development of testing tools to be shared by all participants; and commitment to a series of public interoperability testing under real-world conditions.

REFERENCES

[1] S. Kemmerer, Editor, “STEP: The Grand Experience.” NIST Special Publication 939, July 1999.

[2] Consortium for Advanced Manufacturing - International, “Dimensional Measuring Interface Standard,” Revision 3.0, ANSI/CAM-I 101-1995.

[3] International Association of CMM Vendors, “I++ DME,” Version 1.5. Available: www.isd.mel.nist.gov/projects/ metrology_interoperability/specs/idmespec.1.5.pdf

[4] J. Horst, T. Kramer, J. Falco, W. Rippey, F. Proctor and A. Wavering, “User's Manual for Version 3.0 of the NIST DME Interface Test Suite for Facilitating Implementations of Version 1.4 of the I++ DME Interface Specification,” October 4, 2002.
Available: www.isd.mel.nist.gov/projects/metrology_interoperability/ NISTI++DMEtestSuite3.0UsersManual.pdf

Virtual Mentor: A Step towards Proactive User Monitoring and Assistance during Virtual Environment-Based Training

Maxim Schwartz
Energetics Technology Center
P.O. Box 601
La Plata, MD, USA
mschwartz@etcmd.org

S.K. Gupta and
D.K. Anand
Center for Energetic Concepts Development
University of Maryland
College Park, MD, USA
skgupta@eng.umd.edu
dkanand@umd.edu

Robert Kavetsky
Energetics Technology Center
P.O. Box 601
La Plata, MD, USA
bob.kavetsky@verizon.net

Abstract**—**This paper describes a component of the Virtual Training Studio called the Virtual Mentor, which is responsible for interacting with the trainees in the virtual environment and proactively monitoring their progress. The Virtual Mentor is a component that is embedded in the Virtual Workspace. Some of the tasks it performs are driving the interactive simulation code generated by the Virtual Author, executing user testing, logging user actions in the virtual environment, detecting errors and providing detailed messages and hints, and assisting the instructor in tailoring the generated training material to increase training effectiveness. This paper presents some of the technical challenges and solutions as well as the rationale behind the Virtual Mentor design.

Keywords**:** *Virtual environment-based training, assembly modeling and simulation, proactive user monitoring and assistance in virtual training*

I. Introduction

Due to the rapid inflow of new technologies and their complexities, accelerated training is a necessity in order to maintain a highly productive manufacturing workforce. We believe that existing training methods can be improved in terms of cost, effectiveness and quality through the use of digital technologies such as virtual environments. Personal virtual environments (PVEs) offer new possibilities for building accelerated training technologies.

We are developing a virtual environment-based training system called Virtual Training Studio (VTS) [1]. The VTS aims to improve existing training methods through the use of a virtual environment-based multi-media training infrastructure that allows users to learn using different modes of instruction presentation while focusing mainly on cognitive aspects of training as opposed to highly realistic physics-based simulations.

The VTS system has two main goals. The first goal is the quick creation of virtual environment-based instructions for training personnel in the manufacturing industry so that an overall training cost reduction can potentially be realized by the use of our system. The second goal is to accelerate the training process through the use of adaptive, multi-modal instructions. With VTS, training supervisors have the option of employing a wide variety of multi-media instructions such as 3D animations, videos, audio, text and interactive simulations to create training instructions. The virtual environment enables trainees to practice instructions using interactive simulation and hence reduces the need for practicing with physical components. Our current system is designed mainly for training of cognitive skills: training workers to recognize parts, learn assembly sequences, and correctly orient the parts in space for assembly. The VTS is designed to be an affordable training tool. Hence we developed a low-cost wireless wand and use an off-the-shelf head mounted display (HMD).

The VTS system consists of the following three modules:

- **Virtual Workspace:** The objective of this component of the VTS is to provide the infrastructure for multimodal training and to incorporate the appropriate level of physics-based modeling that is suitable for the operation of a low-cost PVE. Virtual Workspace contains the necessary framework to allow manipulation of objects, collision detection, execution of animations, and it integrates the software with the hardware in order to give the user an intuitive, easy to use interface to the virtual environment. Virtual Workspace offers three primary modes of training: 3D animation mode, which allows users to view the entire assembly via animations; interactive simulation mode, which is a fully user-driven mode that allows users to manually perform the assembly tasks; and video mode, which allows users to view the entire assembly via video clips. Trainees can switch between these modes at any time with the click of a button.

- **Virtual Author:** The goal of the Virtual Author is to enable the user to quickly create a VE-based tutorial without performing any programming [2]. The Virtual Author package includes a ProEngineer assembly import function. The authoring process is divided into three phases. In the first phase, the author begins with a complete assembly and detaches parts and subassemblies from it, creating an assembly/disassembly sequence. In the process of doing this, the instructor also declares symmetries and specifies the symmetry types. In the second phase, the instructor arranges the parts on a table. In the third and final phase, the instructor plays back the generated assembly/disassembly sequence via animation. During this final phase, text instructions are generated automatically by combining data about collision detection and part motion.
- **Virtual Mentor:** The goal of the Virtual Mentor is to simulate the classical master/apprentice training model by proactively monitoring the actions of the user in the Virtual Workspace and assisting the user at appropriate times to enhance the user's understanding of the assembly/disassembly process. If users make repeated errors, then the system will attempt to clarify instructions by adaptively changing the level of detail and inserting targeted training sessions. The instruction level of detail will be changed by regulating the detail of text/audio instructions and regulating the detail level of visual aids such as arrows, highlights, and animations. This paper describes the Virtual Mentor module in detail.

II. Background

Development of the Virtual Mentor came about because of the need for an intelligent agent to operate inside the Virtual Workspace. Virtual Workspace was designed to be the basic infrastructure for running Virtual Author generated tutorials. It is capable of running animations, playing video clips, playing audio, and allowing the trainee to interact with objects in the virtual environment. It was also meant to give the trainee the capability to communicate with the Virtual Training Studio by manipulating virtual buttons on the virtual control panel and using wand commands by pressing buttons on the wand. Running interactive simulation, analyzing logs, and making intelligent decisions when generating tests, however, takes more complicated logic. Using a separate module to accomplish these tasks makes it easier to upgrade and tailor the intelligent behavior of the system. It also makes it easier to plug in the same functionality into other VTS components like the Virtual Author, if, for example, the instructor wants to simulate the training session on the fly within Virtual Author. The tasks of the Virtual Mentor can be divided into two categories: support for interactive simulation and adapting training material based on the performance of users.

A good amount of work has been done in this area in the past. Some have worked on techniques to detect errors made by trainees during training sessions and generate hints to provide them meaningful feedback. An example of a system that uses these techniques is the Georgia Tech Visual and Inspectable Tutor and Assistant, a tutoring system designed to teach satellite control and monitoring operations [3]. Lessons can be assigned one of many styles of tutoring ranging from demonstration via animation with little control of the lesson by the user-to-system monitoring of trainee progress with only occasional intervention by the system. In effect the tutor "fades" as the trainee progresses through the curriculum. Each lesson specifies performance requirements, which the student must satisfy to proceed to the next lesson. Another example of this type of system is Steve, an animated agent who helps students learn to perform procedural, physical tasks in a virtual environment [4]. Steve can demonstrate tasks, monitor students, and provide basic feedback when prompted by trainee. Steve signals mistakes with shaking of the head and saying "No". Yet another good example is a system designed by Abe et al., which teaches novices assembly and disassembly operations on mechanical parts inside a virtual environment by showing a technical illustration to trainees with lines representing assembly paths [5]. The hand motions of trainees are tracked and errors are detected. Trainees are alerted when they grasp wrong parts or move parts in the wrong direction. Monitoring errors and user actions in spatial manipulation tasks and providing highly descriptive feedback will require development of new types of algorithms.

Some work has also been done on intelligent adaptive tutorials. Various researchers have developed next generation tutorials that can adapt their instructions based on a user's capability and progress. Such systems, which adapt instructions to specific users, often use machine learning techniques from the artificial intelligence community. An example of this is AgentX, which uses reinforcement learning to cluster students into learning levels [6]. AgentX chooses subsets of all hints for a problem (instead of showing all possible hints) based on student's learning level. Students are grouped into levels based on pretests and their subsequent performance. If pretest data are not available for a student, then that student is automatically placed in level L4, which represents students who perform in the 50th percentile of the performance distribution.

Subsequent sections will explain the techniques used by Virtual Mentor and the rationale for those features. Section III presents all aspects of running interactive simulation. These include handling of part and assembly symmetries, detecting and reporting errors based on the symmetries, and using symmetry data to improve the quality of dynamic animations. Section IV discusses the initial testing that lead to the development of Virtual Mentor and the idea of an intelligent agent. Section V explains the technical details of logging, log analysis, and generating tests tailored to trainees. Finally, Section VI presents some concluding remarks and presents the future path of Virtual Mentor to achieving more autonomy in custom tailoring of tutorials.

III. Handling of Symmetries And Error Detection

A. Use of Part Symmetries to Check for Correct Placement

According to the case studies and the system testing conducted to the date invloving VTS, interactive simulation, involving manual assembly, turned out to be a popular system capability among users. An important aspect of a well-designed interactive simulation is the proper handling of symmetries. In real world mechanical assemblies, very often there are parts that are highly symmetric along certain planes or axes. Such symmetries often mean that there is more than one correct insertion position and insertion orientation. The challenge of this problem is that the system is not aware of any symmetries and the only information it has access to is the single position and single orientation of each part within the overall assembly. This position and orientation were declared when the assembly was put together by the instructor in the virtual environment. The challenge for VTS is to find out what types of symmetries exist and to calculate other possible positions and orientations during interactive simulation. This allows a user to place a part that is symmetric in some way at one of the alternate insertion locations as it could be done in real life without the system giving an error. It also allows the user to use one of many clones of a part in the assembly process at a particular step without the system requiring the use of a particular clone. Proper implementation of symmetries speeds up the training process by not forcing the user to attempt various correct insertion locations or orientations until the user finally uses only those that were declared during assembly sequence demonstration inside the Virtual Author run virtual environment.

Another reason why part symmetries need to be properly handled are animations. After the instructor demonstrates the assembly process in the Virtual Author monitored virtual environment, Virtual Author automatically generates the initial animation code, in the form of Python script, which will later be executed by Virtual Workspace where users train to create dynamic animations. The initial code, which does not take symmetries into account, will not produce efficient animations for parts that have symmetries. This is because the generated code will always instruct Virtual Workspace to animate the movement of a part to one particular position and orientation – declared by the instructor during the demonstrated attachment. In many cases, it would be better to animate the movement of a part to the nearest symmetric orientation or position. This speeds up the animation and reduces risk of confusing the trainee.

The Virtual Mentor is responsible for enforcing correct attachments and insertions involving part/assembly symmetries, though the Virtual Author is used to declare and categorize the symmetries. When creating tutorials via the Virtual Author, the instructor specifies for each part that exhibits symmetry the main symmetry axis of the part. The main symmetry axis is the axis around which the assembly has the greatest number of allowable orientations. By allowable orientations, we mean that the assembly looks the same and can be attached to the receiving assembly with that orientation. If we use a tube as an example, the main symmetry axis would be the axis of the cylinder because the tube can be rotated around that axis infinite number of ways and will still look the same. The instructor also specifies the number of different permissible orientations around this axis. We call this type of symmetry type A. In addition to this information, the instructor declares a second type of symmetry for each step, which we call type B. In type B symmetry, the instructor specifies one secondary symmetry axis, which is perpendicular to the main symmetry axis and also specifies a sub-type. By declaring the secondary symmetry axis, the instructor states that the assembly being attached may be flipped 180 degrees around this axis and the attachment would still be correct. In addition to declaring a secondary symmetry axis, the instructor also specifies a sub-type. The specified assembly sub-type informs the system about what types of rotations are allowed around the secondary symmetry axis and whether an alternate insertion position may be used for a particular step. The current version of the Virtual Mentor simplifies the problem by allowing only one alternate attachment location for the part being attached to an assembly and only one alternate orientation around the secondary axis. Sub-types for symmetry type B in the current version are:

- Sub-type B1: Allow primary position and primary orientation only
- Sub-type B2: No alternate position allowed, but alternate orientation for primary position is allowed
- Sub-type B3: Alternate position allowed but with primary orientation only (no alternate orientation for primary position)
- Sub-type B4: All combinations of (alternate/primary) positions and orientations are allowed
- Sub-type B5: Alternate position allowed but with alternate orientation only (no alternate orientation for primary position)

We came up with a method to handle placement of parts at alternate locations that is not computationally expensive. Our current method causes the animation to always attach parts to their unique, designated locations and orientations, which were declared during instructor's assembly sequence specification. This strategy simulates the placement of parts at their alternate locations and orientations, by rotating, swapping, and repositioning parts in a way that is least noticeable to the trainee before activating the animation mechanism, which is part of the Virtual Workspace infrastructure.

One example of such swapping is how identical parts are handled. Upon loading all the parts, the Virtual Author automatically detects and marks identical parts. It does this by comparing the number of vertices and the bounding boxes of the parts. At the end of interactive simulation, right before the animation that completes the step is activated, the system swaps clones depending on which clone was originally the designated attachment part for that particular step. This strategy once again allows the Virtual Workspace animation to always attach parts to their unique, designated locations and orientations.

After the check for clones is made, the Virtual Mentor checks if the position of the released part is close enough to the ideal position(s) relative to the receiving assembly. The correct position for the attaching part depends on the sub-type of symmetry type B. For sub-type B5, for instance, there are two allowed positions – primary and alternate. The primary position is specified by the instructor explicitly via the Virtual Author. The Virtual Mentor automatically ascertains the alternate position for sub-type B5 by first drawing a vector from the primary insertion location to the final location and then doubling that vector. A marker is placed at the tip of this vector. The Virtual Mentor then checks if the released part is close to the alternate position. An example of sub-type B5 symmetry is shown in Fig.1, where a primer retainer is being inserted into the inner tube. One interesting aspect of sub-type B5 symmetry is that if the alternate insertion position is used on the other side of the inner tube, then the primer retainer must have the alternate orientation relative to the receiving assembly so that it is once again facing the inner tube. Alternate orientation is achieved by rotating the primer retainer around the secondary symmetry axis 180 degrees. If the trainee has placed the primer retainer at its alternate position, then the Virtual Mentor checks if the primer retainer has the alternate orientation. If that is the case, the Virtual Mentor flips the receiving assembly/part, in this case the inner tube, 180 degrees around the instructor-specified secondary axis before passing control to the Virtual Workspace animation generating mechanism. By rotating the receiving assembly, the attaching subassembly is now at its primary insertion position and orientation, and as we already mentioned, all parts must be placed at their primary positions and orientations before animation is activated and the attaching part is inserted into the receiving part. In most cases the trainee does not notice this rotation.

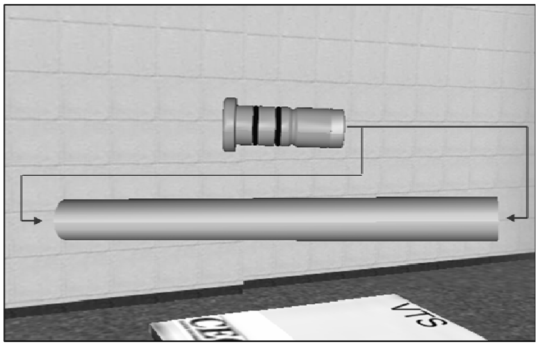


Fig. 1. Primer Retainer with Two Correct Insertion Position

The final check that the Virtual Mentor makes is the correctness of rotation around the primary symmetry axis. If the placement is correct, the Virtual Mentor rotates the attaching part in increments based on the number of permissible orientations. For example, if this number is 3, then the increment is 120 degrees. If the number is 4, then the increment is 90 degrees. The system must rotate in these increments to make sure the user does not notice a change in rotation. By rotating in these increments, the Virtual Mentor takes advantage of the attaching part’s symmetry to conceal the rotation. The reason why the attaching assembly must be rotated at all is because without such “setup rotation” the animation will be forced to rotate the part until it reaches its designated orientation within the assembly, slowing down the training in the process.

Fig. 2 shows an example of sub-type B1 symmetry. A front plate assembly containing the needle and needle valve is being attached to the engine block. There are no alternate orientations or positions. The trainee must place the front plate assembly very close to the primary orientation and positions declared by the instructor within theVirtual Author. Otherwise, an error message is given to the trainee describing the flawed orientation or position.

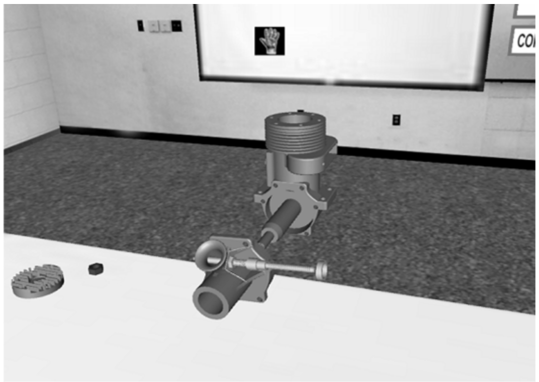
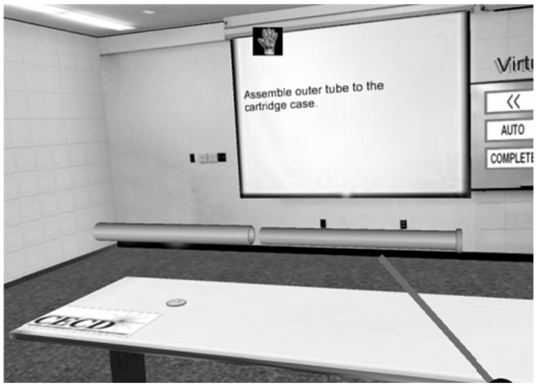


Fig. 2. An example of sub-type B1 symmetry where only the primary position and primary orientation are allowed. In effect, there is no symmetry.

Fig. 3 shows an example of sub-type B2 symmetry. An outer tube is being attached to the rest of the rocket motor assembly. Sub-type B2 symmetry says, “No alternate position allowed, but alternate orientation for primary position is allowed.” This means that the outer tube can only be attached from one side of the rocket motor assembly – the primary position declared during authoring. However, the outer tube is symmetric along the plane that is perpendicular to the outer tube’s main symmetry axis. The main symmetry axis of the outer tube is the axis of the cylinder. This means that if the instructor chooses a secondary symmetry axis that is perpendicular to the main symmetry axis and flips the outer tube 180 degrees around the secondary symmetry axis, then the outer tube will look the same and can be attached to the rocket motor assembly with that orientation. This flipped orientation is called alternate orientation and for sub-type B2, it is allowed. In this scenario, the mobile part, the part being attached, also has an infinite number of symmetric orientations around the main symmetry axis. Since this orientation has to do with the main symmetry axis, it is type A symmetry. The trainee can use any orientation around the main axis during placement and the Virtual Mentor will allow that instead of generating an error.



Assemble outer tube to the cartridge case.

Fig. 3. An example of sub-type B2 symmetry where the attachable part (outer tube) can only be inserted at the primary position, but with either primary or alternate orientation (around the secondary axis). Secondary axis is perpendicular to the main axis, which is the axis of the cylinder.

The two tutorials used in the latest case study contain twelve steps involving symmetries out of a total of nineteen steps. Three steps out of nineteen also involve the use of clones. During the case study, we observed users placing symmetric assemblies and parts at both their primary and alternate locations. The Virtual Mentor demonstrated 100 percent accuracy in detecting alternate correct placements and allowing users to proceed. One such case that we observed was step three of the ejection seat rocket tutorial in which one of the users had to place cartridge propellant grain into a cartridge case. The propellant grain was cylindrical while the case was a tube. The user placed the cartridge propellant grain on the other side of the cartridge case, which was not the original insertion location declared in the Virtual Author. The Virtual Mentor correctly gave the user a success message and correctly animated the propellant grain going into the case from the alternate location.

B. Error Messages

Detailed and precise error messages are important in the quick diagnosis and resolution of a problem, like for example an incorrect assembly attempt. In order to provide detailed error messages and helpful hints in the event of a mistake, the Virtual Mentor must first determine exactly what type of error was made. The current version of the Virtual Mentor is capable of detecting four types of error:

- Incorrect part used for a given step in the process
- Part was placed in an incorrect position
- Primary axis of the part is not correctly aligned
- Part is not correctly rotated around the primary axis of the part
- Primary axis of part is correctly aligned by object facing in the opposite direction

Whenever the Virtual Mentor gives the third, fourth, or fifth error to the user, it draws the primary axis through the part which the trainee attempted to assemble to another part or subassembly. This way the trainee knows exactly what axis is being referred to by the Virtual Mentor.

In the process of testing our system using volunteers, we observed that when trainees paid attention to the text error messages, they corrected their mistakes more quickly, on average, in order to complete the step. Trainees who, for whatever reason, did not pay attention to the text errors took significantly longer, on average, to correct their mistakes.

For the two tutorials used in our case studies, the Virtual Mentor reported a total of 146 errors during training. While monitoring the training of each trainee in VTS, no error detection or error classification mistakes on the part of the Virtual Mentor were observed. One of the instances of error detection and classification that we observed typified the detection and classification of an error by the Virtual Mentor. In the fourth step of the model airplane engine tutorial, a trainee had to place a cylinder head on top of the engine case. The cooling fins on the cylinder head had to be aligned parallel to the crankshaft. The user positioned the cylinder head correctly above the engine case but did not align the cooling fins with the crankshaft. After signaling to the Virtual Mentor to complete the assembly by pushing the "Complete" button, the trainee received a text error message saying, "Error: The object which needs to be inserted is not oriented correctly." The trainee then watched an animation of the step and completed it correctly.

IV. Annotation of Ambiguous Instructions

Once again the first major task of the Virtual Mentor is to provide support for interactive simulation by using information about part and assembly symmetries at each step in the assembly process to detect correct and incorrect part placements, report errors, and to prepare the part being attached for animation by performing a series of hidden rotations and translations. The second major task of the Virtual Mentor is to assist the instructor in adapting the training material based on the performance of trainees. The need for the second task came about as a result of some informal testing conducted early in the development of the VTS.

As the infrastructure of the VTS was built up to a certain level and a sample tutorial was created, we used six volunteers, consisting of graduate and undergraduate engineering students, to test the training effectiveness of the system and its user interface. At the time, the Virtual Author was not available so all the custom code needed for the tutorial was written manually in Python script by a programmer. The custom code included the text instructions, video and audio files, rules for dynamic animations, code to run interactive simulation, and variable detail visual hints to be used within interactive simulation. The six volunteers were trained inside the VTS to assemble a navy rocket that is a component of an ejection seat. After the virtual environment training with CAD models of these devices, the trainees were given actual parts and asked to assemble real devices. Even though most volunteers felt very confident after VTS training and felt they could easily assemble the real devices, a good number of them made some mistakes during the assembly of the real devices. What is interesting is that the errors were pretty consistently being made at a certain set of points in the assembly process. Rocket motor tutorial included an assembly step where the trainee must attach a small cap to one side of a rocket nozzle. The cap must be attached to the side of the nozzle with a relief. The animation that all volunteers saw during training showed the cap moving toward the side of the nozzle with a relief. Unfortunately, the limitations of the virtual reality display technologies used during testing made it difficult to see the relief due to a low 640 X 480 resolution. During physical testing some trainees attempted to attach the cap to the wrong side of the nozzle without the relief. There was another point in the assembly process that caused problems for several volunteers. Here the trainee must slide a rubber o-ring onto the right rectangular o-ring groove on the primer retainer. Some trainees slid the real o-ring onto the rounded grove next to it which is not designed for o-rings. The trainees who did this did not notice the difference between rounded and rectangular groves during virtual reality training.

After the initial testing, we added more detail to the tutorials to highlight the problem areas. The added details were in the form of additional text and audio instructions and more detailed animations. Animations were expanded in certain steps to include flashing 3D arrows that pointed out important features of the assembly. Fig. 4 shows the second scenario where an o-ring must be rolled on top of a rectangular o-ring grove.

After the changes were made, we conducted a second round of training and testing with another six volunteers. During the second round of testing, the new volunteers made fewer errors. These results showed that no matter how clear the instructor tried to be when generating the training material, certain flaws in the training material could only be detected after user testing and analysis of the results. This spurred the need for development of an intelligent agent operating inside the virtual environment that is capable of not only logging all the actions of the trainees during training sessions, but also capable of generating targeted tests, analyzing the results, and later even automatically adapting the tutorials. The more such tasks the Virtual Mentor can perform automatically, the less of a burden will be placed on the instructor. Current version of the Virtual Mentor performs logging during training sessions and tests within the virtual environment, analyzes the logs, generates tests that are customized for each trainee based on that trainee's performance, and provides recommendations to the instructor. We envision the Virtual Mentor not simply giving the instructor advice on what parts of the tutorials to adjust, but actually adapting the tutorials automatically with the instructor's approval.

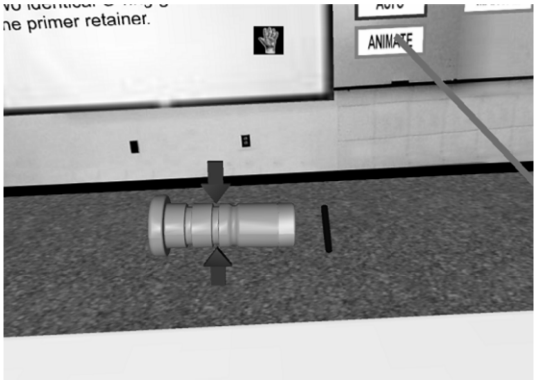


Fig. 4. Detail in the form of flashing arrows is added to the animation of rubber o-ring attachment to o-ring grove.

V. Logging, Analysis and Generation of Custom Tests

While trainees train to assemble a device in the Virtual Workspace and interact with the system, the Virtual Mentor logs a very wide range of events. Each event is logged with a timestamp representing the number of expired seconds since the beginning of the tutorial. Some of the events that the Virtual Mentor logs are:

- Activation of buttons on the virtual control panel
- Activation of animations
- Activation of hints
- Activation of video clips
- Browsing of steps in the assembly process by skipping to the next step or going back to a previous one
- Pick-up of objects
- Release of objects
- All errors detected during interactive simulation and the type of error
- Successful completions of steps
- Use of wand functions like rotation of objects with wand buttons and trackball

In addition to events, the Virtual Mentor also periodically logs the position of the user's head and the position of the wand. This information is logged in order to analyze the range of users' movement in the virtual environment. The amount of movement can later be used to determine the efficiency of the virtual room by answering such questions as:

- Are the parts on the table spread out too much or arranged inefficiently causing excessive wandering?
- Are the users moving and rotating objects manually by picking them up with the virtual laser pointer or are they using the wand buttons and trackball to rotate and move objects?
- Are the users looking at parts from a different perspective by walking around them or are they picking up and rotating them with the laser pointer?
- Should the size of the room be increased or decreased?

The logs are stored as text files in a format that can be loaded into Microsoft Excel. A new file is generated for each trainee.

After the training session inside the Virtual Workspace is over, the Virtual Mentor performs some analysis on the trainee's log in order to generate the appropriate test for the trainee. The trainee receives a message from the Virtual Mentor that is displayed on the projector screen that a test is being generated. The trainee remains inside the Virtual Workspace, while the Virtual Mentor analyzes the log and generates the test. After the Virtual Mentor finishes analyzing the log, it generates new random positions for all parts on the table and chooses a subset of the training session assembly steps for the trainee to perform in the Virtual Workspace. Certain features like text and audio instructions, hints, videos, and step browsing are disabled during the test mode. The subset of steps the trainee is tested on contains about 50 percent of the total number of tutorial steps. A certain number of steps is first chosen based on log data and the rest are picked randomly.

The process of choosing test steps based on log data begins with extraction of the following information from the log: number and type of errors made, number and type of hints used, and the number of times the animation has been played. Each of these pieces of data is extracted for each step in the tutorial. Next, the Virtual Mentor gives each step in the tutorial a difficulty rating. When calculating the difficulty rating the Virtual Mentor uses the occurrence and the weight of the extracted events. Errors have a weight of 3, hints have a weight of 2, and animations have a weight of 0 or 1. The first animation event has a weight of 0 while all subsequent animation events for a particular step have a weight of 1. The reason why multiple animation events are used to gauge step difficulty is because it was noticed during user testing and case studies that some trainees used animations as hints instead of using the hint feature in interactive simulation mode. Those trainees would switch to the auto mode, play an animation, and switch back to the interactive simulation mode. Next, the Virtual Mentor sorts all steps in descending order based on the difficulty rating.

After the steps have been sorted the Virtual Mentor must rearrange some steps depending on the error type of problem steps. There is only one error type that requires this – assembly sequence error. Assembly sequence error occurs when the trainee forgets what step to perform next by trying to attach the wrong part for a particular step. In order to test for assembly sequence memory, the Virtual Mentor must present the trainee with two steps – the step where the error occurred and the step before it. The only exception to that is if the step where this type of error occurred is the first step in the tutorial, in which case only the step where this error occurred will be used. To perform the rearranging of sorted steps, the Virtual Mentor visits each step in the queue where difficulties were detected. If a problem step S contains an assembly sequence error, then the Virtual Mentor moves step S – 1 in front of step S.

After the rearrangement has been done, the Virtual Mentor takes the steps in the top fifty percentile and uses them as steps the user will be tested on. If the number of problem steps makes up less than fifty percent of the total number of tutorial steps, then the Virtual Mentor chooses some random steps as a filler. This strategy ensures that all trainees are given tests of the same length to maintain consistency for future gathering of statistics.

The trainee is then put into interactive simulation mode and given the chosen test steps in the right sequential order. If the trainee performs a particular step correctly or makes three errors while in that step, the Virtual Mentor loads the next step in the queue. While the trainee is being tested, all of his actions are once again monitored and logged in a separate test log file. At the conclusion of the test, the Virtual Mentor analyzes the log file associated with the test and updates the master log associated with the used tutorial. The master log contains a sorted list of tutorial steps and errors associated with those steps. Steps at the top of the list have the highest occurrences of errors for all trainees. After updating the master log, the Virtual Mentor checks the top thirty three percentile of steps for changes in position. If a particular step in the top thirty three percentile advances to a higher position, the Virtual Mentor adds it to the list of steps to bring to the instructor's attention. At the end of the analysis, if the list of changed steps is not empty, the Virtual Mentor sends out an email to the instructor containing the list of steps of a particular tutorial which have advanced in difficulty level as well as the error types that caused this rise. The logging and testing process flow is summarized in Fig. 6.

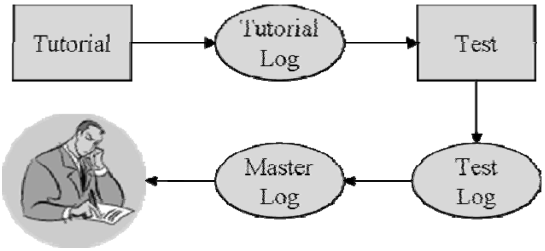


Fig.6. Flow of Information in the Log Analysis Process

VI. Conclusions

The Virtual Mentor is a software component embedded in the Virtual Workspace which is responsible for proactive monitoring of trainees, logging their progress, automatically generating customized tests, and sending out reports to the instructor. The need for the Virtual Mentor arose as a result of informal user testing conducted to determine the training effectiveness of VTS. We realized that tests given to the trainees at the end of a tutorial can reveal confusing areas of the tutorial which may need additional detail for clarification. Current version of the Virtual Mentor alerts the instructor of the detected problems of the tutorial. Future versions of the Virtual Mentor will take over the task of changing the level of detail, automatically adding more detail to tutorials when problems are detected, and removing details after long periods of good trainee performance. The Virtual Author always generates a maximum level of detail when it automatically produces text instructions. Currently, the instructor is responsible for removing too much detail from text instructions and adding arrows to animations when necessary. Future versions of the Virtual Mentor will automatically control the detail level of generated text instructions, the level of details in animations, and the level of details in the hints.

Acknowledgement

This research is supported in parts by the Center for Energetic Concepts Development at the University of Maryland and Naval Surface Warfare Center at Indian Head.

References

[1] J.E. Brough, M. Schwartz, S.K. Gupta, D.K. Anand, R. Kavetsky, and R. Pettersen, “Towards Development of a Virtual Environment-Based Training System for Mechanical Assembly Operations,” Accepted for publication in *Virtual Reality*.

[2] M. Schwartz, S.K. Gupta, D.K. Anand, J.E. Brough, and R. Kavetsky, “Using Virtual Demonstrations for Creating Multi-Media Training Instructions,” *CAD Conference*, Hawaii, June 2007.

[3] R.W. Chu, C.M. Mitchell, and P.M. Jones, “Using The Operator Function Model And Ofmspert As The Basis For An Intelligent Tutoring System: Towards A Tutor/Aid Paradigm For Operators Of Supervisory Control Systems,” *IEEE Transactions on Systems Man and Cybernetics*, 25(7):1054-1075, July 1995.

[4] J. Rickel and W. L. Johnson, “Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition and Motor Control,” *Applied Artificial Intelligence*, 13 (4-5): 343-382, June-August 1999.

[5] N. Abe, J.Y. Zhang, K. Tanaka, and H. Taki., A Training System Using Virtual Machines For Teaching Assembling/Disassembling Operations To Novices,” *Proceedings of IEEE International Conference on Systems Man and Cybernetics*, pp 2096-2101, 1996.

[6] K.N. Martin and I. Arroyo, “AgentX: Using Reinforcement Learning to Improve the Effectiveness of Intelligent Tutoring Systems,” *Lecture Notes in Computer Science*, 3220: 564-572, 2004.

Report on Panel Discussion on (Re-)Establishing or Increasing Collaborative Links Between Artificial Intelligence and Intelligent Systems

B Brent Gordon
NASA Goddard Space Flight Center
Information Science and Technology Research Group
Greenbelt, MD 20771 USA
bgordon@backserv.gsfc.nasa.gov

I. Introduction

A panel discussion on "(Re-)Establishing or Increasing Collaborative Links Between Artificial Intelligence and Intelligent Systems" was held on August 30, 2007, as a session of the *Workshop on Performance Metrics for Intelligent Systems* (PerMIS'07), 2007. The panelists were: James Albus, Senior Fellow, Intelligent Systems Division, National Institute of Standards and Technology; Ella Atkins, Associate Professor, Aerospace Engineering, University of Michigan; Henrik Christensen, KUKA Professor of Robotics in the Computer Science Department and Director of the Center for Robotics and Intelligent Machines, Georgia Institute of Technology; Lawrence Reeker, Computer Scientist, Information Technology Laboratory, National Institute of Standards and Technology; Alex Zalinsky, Director, CSIRO Information Communication Technology (ICT) Centre and Group Executive, Information and Communication Sciences and Technology. The moderator was Brent Gordon, Computer Scientist, Computer and Information Sciences and Technology Office, NASA Goddard Space Flight Center. Finally, audience participation, in the form of comments, opinions, or questions, was also encouraged. The discussion lasted 90 minutes.

The premise of the discussion was that at least some parts of the Artificial Intelligence community and Intelligent Systems community, as represented at the conference, i.e., mainly intelligent controls and robotics, have some fundamental goals in common, despite very different histories and approaches to them. Thus the main issues were to clarify what those goals are in a way that can be understood by all, identify the major difficulties in getting people from the different communities to talk to each other, and suggest ideas that might have a good chance of increasing communication and interaction, and perhaps lead to more collaboration.

The next section is a highly condensed summary of the discussion, following which we address what conclusions the discussion explicitly and implicitly supports.

II. Discussion synopsis

To begin with each panelist gave a short statement of their views on artificial intelligence and intelligent systems. (Except for Albus's, these statements were not prepared in advance.) Albus suggested that comparison with humans is a good metric against which to measure high level autonomous systems, and that it is time that the goal of developing a computational theory of mind be treated as a serious scientific problem. Atkins proposed that humans have not naturally evolved to perform well in air and space, i.e., there are some areas where machines can do better than humans, although we might want intelligent machines to emulate humans in their decision-making. Christensen emphasized real world autonomy, and described the robotic systems the EU-funded group he leads is working on as well as relevant examples. Reeker then brought up machine learning, and computational linguistics, as examples of issues more in the realm of intelligence than autonomy, and indicated that these areas took their inspiration from cognitive science, not strictly neuroscience. Zalinsky suggested taking a more bottom-up approach to problems that require intelligent behavior, and that the key ingredients for adaptation and learning are knowing how to represent information in a way similar to the brain, and embodiment.

A. *What are the goals or questions of common interest to the AI and intelligent systems communities?*

Christensen suggested that it depends on the project's time scale, since in such a collaboration up to a year may be required for everyone to become comfortable with a common vocabulary, and again emphasized the importance of working with embodied systems. Atkins brought up the dichotomy of symbolic modeling on the one hand and mathematical and physics models on the other, with the necessity, or at least common circumstance, of reasoning under uncertainty. Albus noted that intelligent autonomous robots need elements of both symbolic modeling and control theory. Christensen proposed that a systems perspective would be required for a perceptual system to be able to recognize chairs after seeing a single example of a chair. Louise Gunderson suggested that aiming at human-level intelligence directly is too high a goal, and advocated a roadmap strategy of starting with models of simpler vertebrates. Atkins presented the ideas of limiting perceptual algorithms in a domain-dependent way, and connecting perceptual problems with action problems. Reeker noted that we still don't know how children acquire ontologies. Steven Kalik observed that most people do and most machine systems don't represent a problem in multiple ways, and suggested building a system that would do so and then select the best approach to solving the problem. Albus favored the autonomous driving problem, as it is fundamentally locomotion, it requires understanding space, time, dynamics, environmental properties, and may require symbolic processing; and can attract funding. Zalinsky preferred to emphasize the problem of how to represent information. Christensen thought that manipulation might be a good problem, in that intelligent or automation systems of potential interest even to large manufacturing can be completely worked out in smaller scale.

B. What are the interesting scientific questions that might attract "both sides" if formulated in the right way?

James Gunderson pointed out that all organisms build models of the world and operate with reference to that internal model they have built. Raj Madhavan asked that if it all came down to a matter of systems integration, what would be the differences between an AI approach and an intelligent systems approach to integration? Christensen suggested that if integration is always feedforward while the situation is normal, and feedback is activated only when something fails, this might have a dramatic impact on how the system is designed. Reeker mentioned that machine learning is getting more attention and being more widely used. Someone from the audience suggested that there is need for a more abstract architecture that more easily allows new components to be plugged in. Christensen observed that we need to think about the right level of abstraction. Albus agreed that there are many levels of abstraction within the context of any any problem.

C. What can we do that might have the best chance of increasing the levels of communication, interaction, or collaboration among members of different communities with similar motivations?

Atkins suggested improving the educational system, since students now generally can't learn both computer science and physics, say, in depth. Christensen offered three specific ideas: first, projects of a nature that encourage multidisciplinary collaboration, and are long-term enough to allow participants to build a common vocabulary, with some mechanisms to encourage these projects; second, educational activities that cross traditional group lines, such as summer schools aimed at graduate students; third, other community-building mechanisms for meeting and communicating even in the absence of existing collaborations. Zalinsky emphasized the importance of computer scientists taking a multidisciplinary approach and finding challenges that appeal to politicians and the public. Albus mentioned the problems that exist in mastering domain vocabulary, let alone multiple domains, and expressed the need for an overarching architecture as a means for everyone to see how their vocabulary would fit together with everyone else's. Kalik pointed out that mechanical and electrical engineering both use the same vocabulary, even if computer engineering doesn't, there may be core components in common. Albus said it is a very small set of basic concepts. Atkins suggested that robotics may offer an option, as it exhibits high-level concepts in a real-world application. J. Gunderson indicated that other engineering disciplines have overcome the vocabulary problem, and seem able to work together. Atkins concluded with the observation that collaboration has physical, computational, and informational aspects to it.

III. Conclusions, and future work

As suggested in the introduction, the first important assumption underlying even the idea of having this panel discussion was that there are certain problems of common interest to segments of both the Artificial Intelligence and Intelligent Systems communities, and that in the long-run those sub-communities would benefit from greater interaction with each other. The first conclusion is that this important assumption is valid. For the panelists, as experts in their fields, were definitely smart enough and opinionated enough to challenge this assumption, which was stated explicitly at the time, if they disagreed with it. But not only did they not challenge it, they bought into it, both in their opening statements and throughout the discussion, in the suggestions and elaborations they proposed.

Continuing to consider the discussion as a whole rather than the details of what was said, the major take-away message is a resounding endorsement of the importance and the urgent need to do *something* to increase the level of interaction between the relevant sub-communities with common interests. Concerning what problems would draw the widest audience, what mechanisms to use, etc., there were a number of suggestions, but in the context of this panel discussion, all at the level of initial brainstorming.

Thus, the final conclusion is that the next step should be one or more planning meetings involving a modest number of experts, whose goal would be to put together some specific proposals along the lines of the suggestions that emerged from this panel.