Realization of the 5-Axis Machine Tool Digital Twin Using Direct Servo Control from CAM

Roby Lynn[1], Mukul Sati[2], Tommy Tucker[3], Jarek Rossignac[2], Christopher Saldana[1], Thomas Kurfess[1]

[1]George W. Woodruff School of Mechanical Engineering, Georgia Tech

[2]School of Interactive Computing, Georgia Tech

[3]Tucker Innovations, Inc.

NIST Model-Based Enterprise Summit
April 4, 2018

Introduction

- Machine tools are *difficult to operate*
  - Significant experience required for effective use
- Control
  - Decades old G-Code
  - Geometric primitives
  - Unidirectional data transfer
- Feedback
  - MTConnect
  - OPC UA

```
G0 G17 G20 G40 G49 G64 G80 G90 G98 G54
G54.4 P0
G53 G90 G0 Z0.
G53 G90 G0 B80. C90.
G65 P9544 H1
G54.1 P300
G54.4 P1
S4800 M3
G43 H#3020 X-3.2923 Y-7.6496 Z8. B80. C90.
Z4.2575
G05 P2
G61.1 P3
G1 Z3.2575 F.02
G41 D#3020 Y-7.5079 F.006
G3 X-3.7647 Y-7.0354 I-.4724
G2 X-4.6077 Y-6.1924 J.843
```

```
<Samples>
  <Load dataItemId="cl" timestamp="2017-01-31T22:16:39.623784Z"
name="Cload" sequence="1510344">0</Load>
  <Angle dataItemId="cpos" timestamp="2017-01-31T22:14:40.170085Z"
name="Cdeg" sequence="1509772" subType="ACTUAL">0</Angle>
  <RotaryVelocity dataItemId="cs" timestamp="2017-01 31T22:16:08.982719Z"
name="Cfrt" sequence="1510223" subType="ACTUAL">0</RotaryVelocity>
</Samples>
```

- *How can we use the digital twin concept to enable CNC machine tools to be operated as easily as 3D printers?*

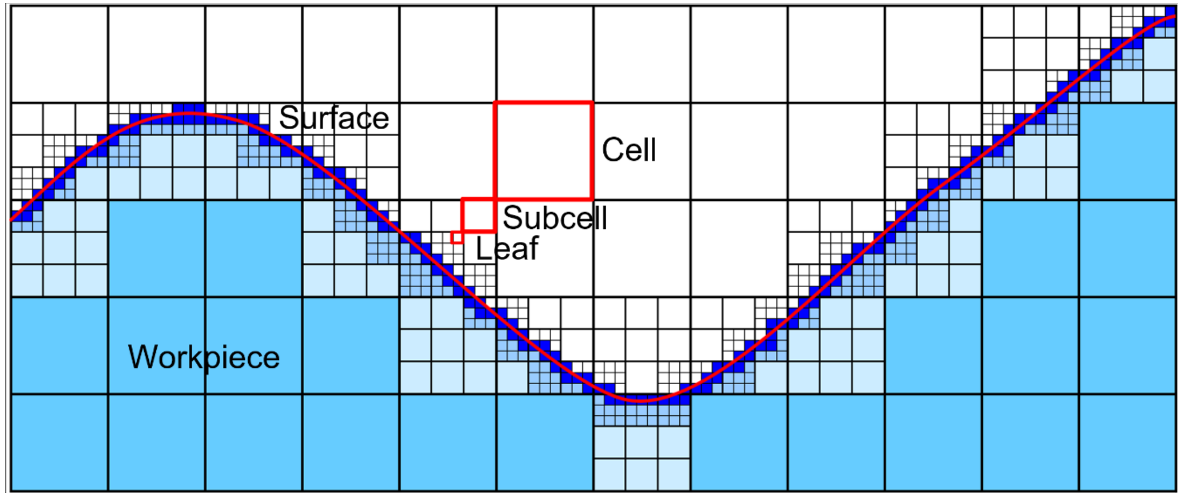
Current Industrial Practice: CAM to G-Code to CNC

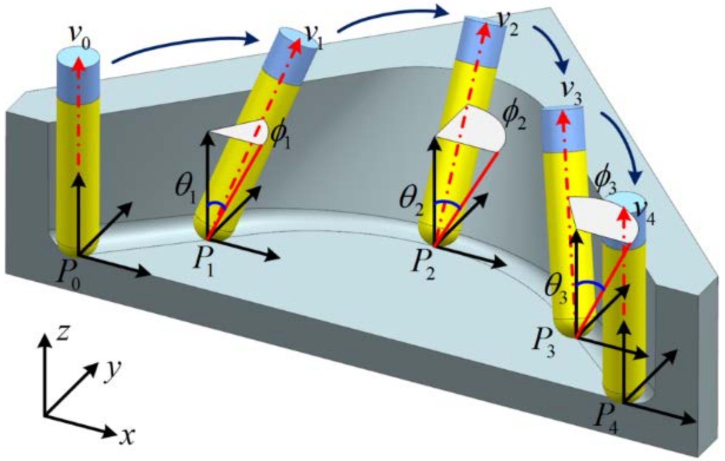
- CAM systems create G-Code using post processors
  - Machine specific instructions
- G-Code: text-based NC programming language, originally standardized in 1960s, still dominant today
  - Serial execution
  - Lines, arcs, and splines (primitives)
  - Maximum traversal velocity
- Industrial CNC systems
  - Proprietary architecture
  - Lack of interoperability
  - Options available – at a cost

Photo Credits: Okuma America, Mazak Corporation

Digital Volumetric Processing (DVP) Using High-Performance Computing

- Reformulation of the traditional toolpath planning problem
- Discrete Geometry
  - Part surface comprised of many small cubes called **voxels**
  - Voxel size *s* determines resolution of part surface
  - STLs, scans, point clouds
- High Performance Computing (HPC) using graphics processing units (GPUs)

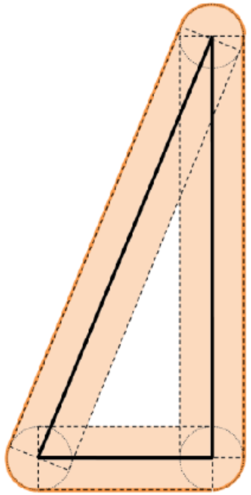




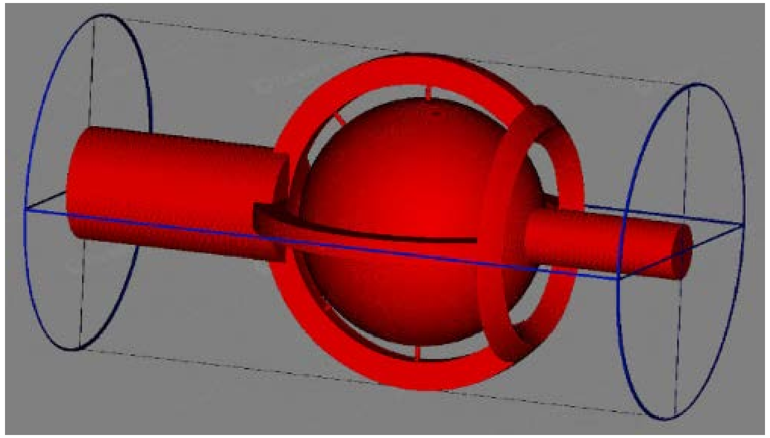
[1] R. Lynn, M. Dinar, N. Huang, J. Yu, J. Collins, C. Greer, T. Tucker, T. Kurfess. *Direct Digital Subtractive Manufacturing of Functional Assemblies Using Voxel-Based Models*. ASME Journal of Manufacturing Science and Engineering, 2017.

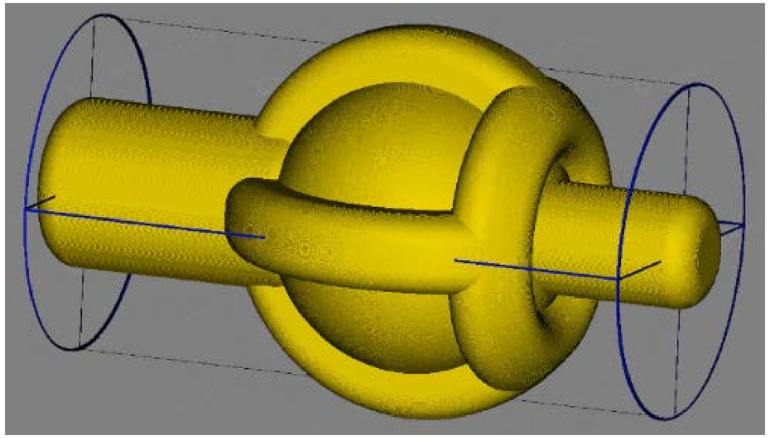
5-Axis Voxel-Based Path Planning: Contact Volume Generation

- Constraints on tool center position
  - Axial cutting depth
  - Final part geometry



- A sequence of volumetric offsets
  - Positive offset (expansion) of model by tool radius and cutting allowance
  - Negative offset (shrinkage) of stock volume by DoC
  - Union



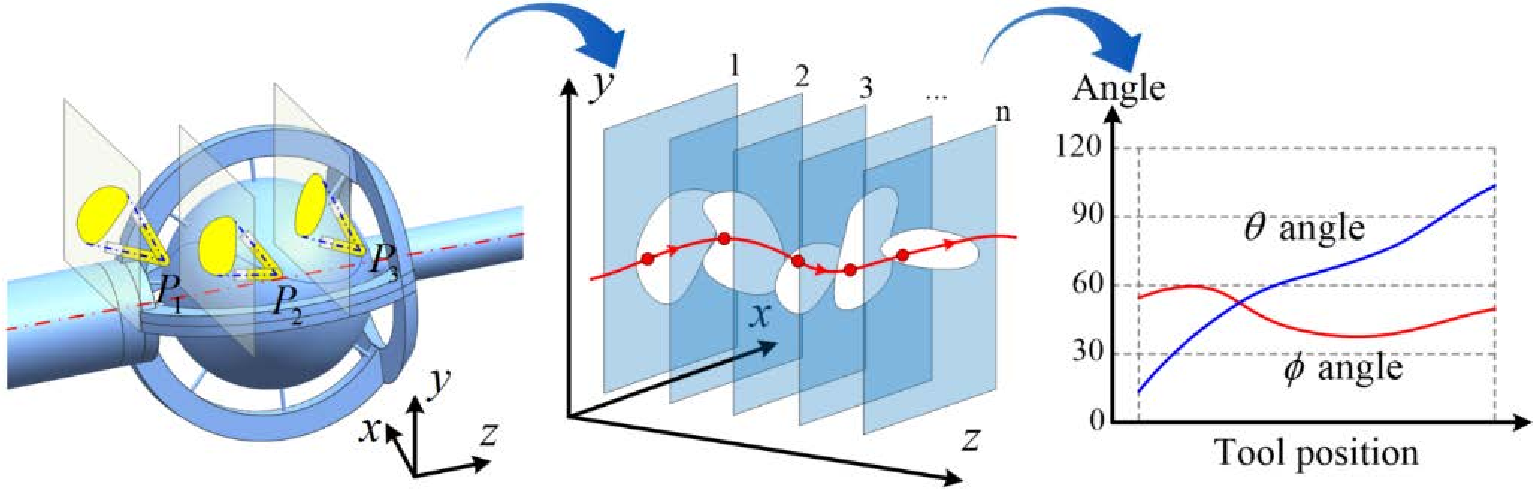


---

[2] R. Lynn, D. Contis, M. Hossain, N. Huang, T. Tucker, T. Kurfess. *Voxel Model Surface Offsetting for Computer-Aided Manufacturing Using Virtualized High-Performance Computing*. SME Journal of Manufacturing Systems, 2016.

[3] M. Hossain. *Voxel-Based Offsetting at High Resolution with Tunable Speed and Precision Using Hybrid Dynamic Trees*. Georgia Institute of Technology, 2016.

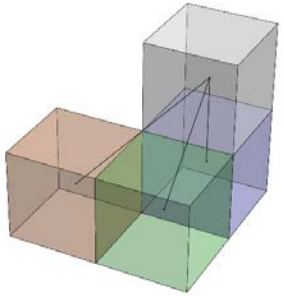
5-Axis Voxel-Based Path Planning: Accessibility Analysis



- Constraints on tool axis orientation
  - Collisions, travel constraints, surface normals

- Stack of binary bitmaps of unique orientations which are checked for collisions with workpiece and fixture
  - Access path through white regions on maps

- Result: toolpath consisting of *small, discrete, 5-axis movements* between voxels

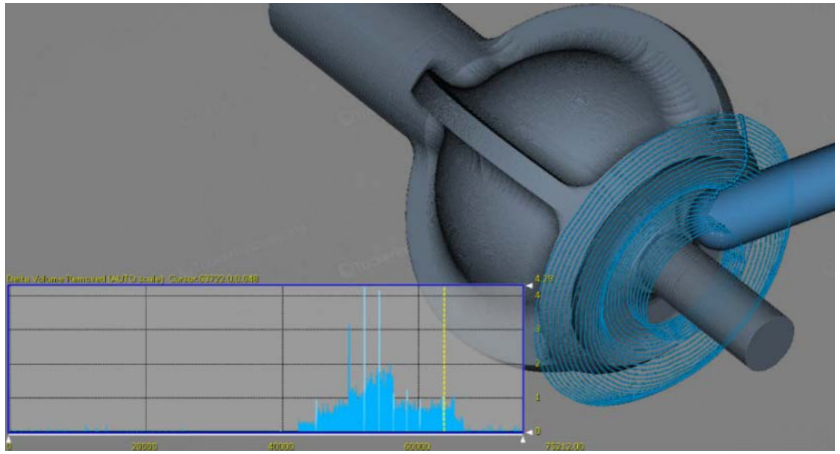


Process Intelligence: Material Removal Rate Analysis and Control Using Voxel-Based CAM

- Volume removed $\boldsymbol{V}_i$ is the sum of voxels of side length *s* swept by cutter envelope *C* over a step *i*

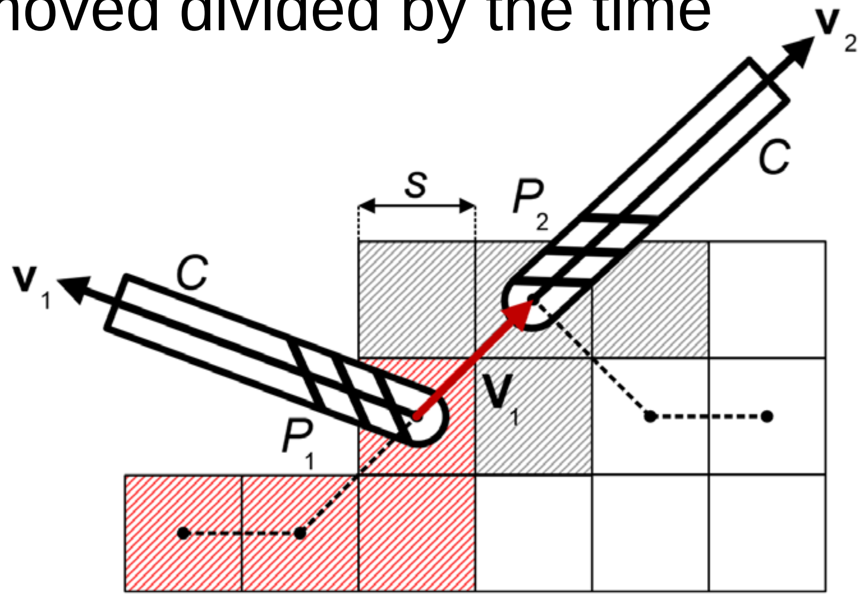
$$\boldsymbol{V}_i = \sum_i dv \times s^3 \mid dv \in C_i$$

- Average MRR for the step *i* is the volume removed divided by the time taken to complete the step, Δ*t*



$$MRR_i = \frac{\boldsymbol{V}_i}{\Delta t_i}$$

$$\Delta t_{i \to i+1} \geq \frac{\sum_i dv \times s^3}{\mathrm{MRR}_{\mathrm{Limit}}}$$

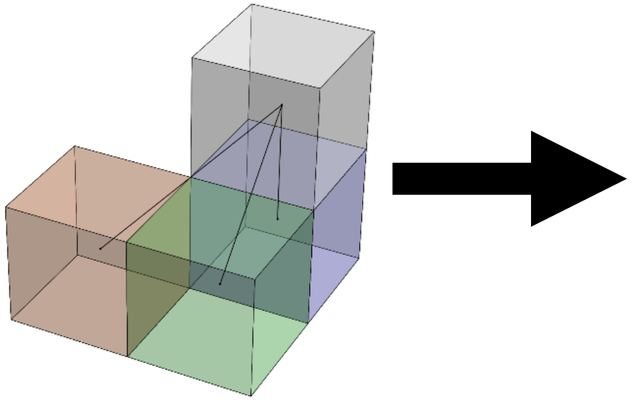


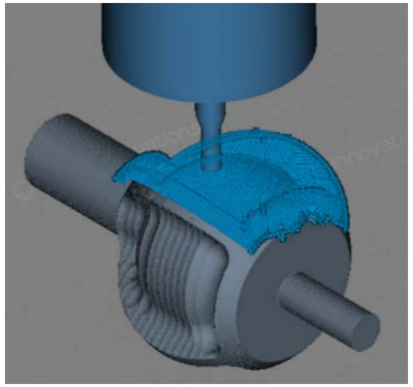
[4] Z. Nie, R. Lynn, T. Tucker, T. Kurfess. *Voxel-Based Analysis and Modeling of MRR Computational Precision Using GPGPU Computing (Under Review)*. ASME Journal of Manufacturing Systems, 2017.

Traditional Method: DVP to G-Code

- Typical 5-axis block: G1XxYyZzAaBbFf
  - x, y, z, a, and b are endpoints
  - f is *maximum* speed
  - Velocity constrained so all axes arrive at the same time

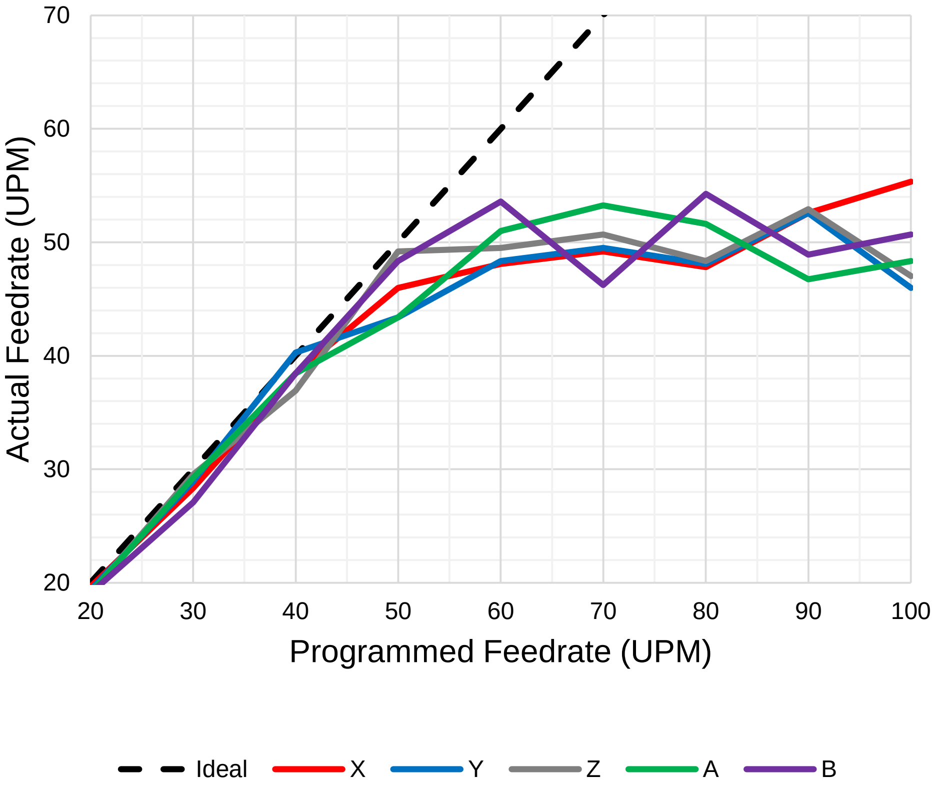
- For sufficiently small voxel sizes, a toolpath can consist of <u>millions of linear movements</u> (G1s) between voxel centers



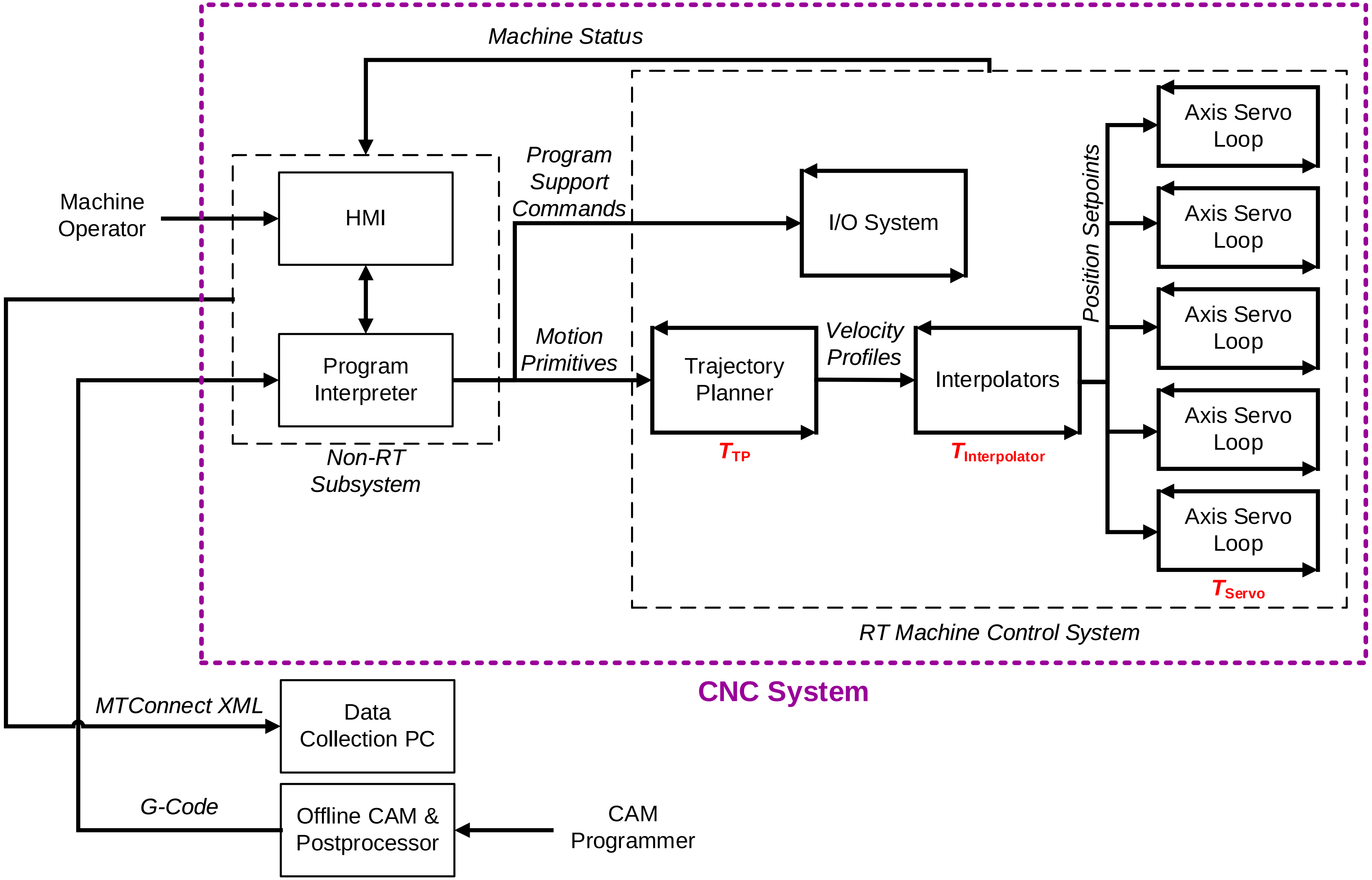


$$L_{\text{Shared Face}} = s, \quad L_{\text{Shared Edge}} = \sqrt{2}s, \quad L_{\text{Shared Vertex}} = \sqrt{3}s$$

2.5μm Movement Size

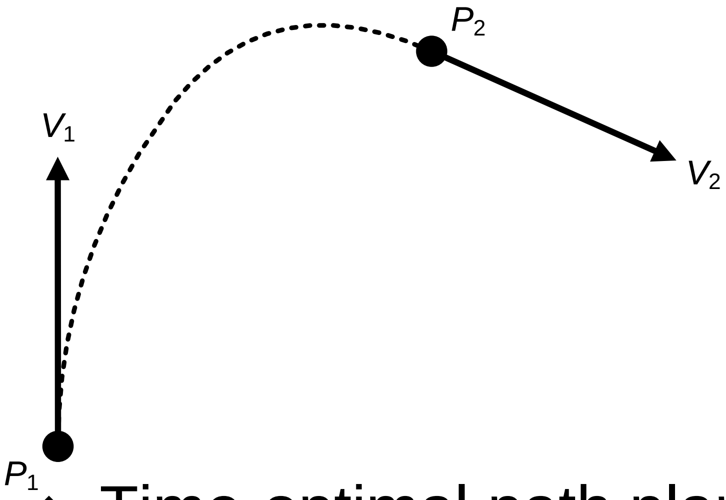


Traditional Method: Obfuscated Trajectory Planning



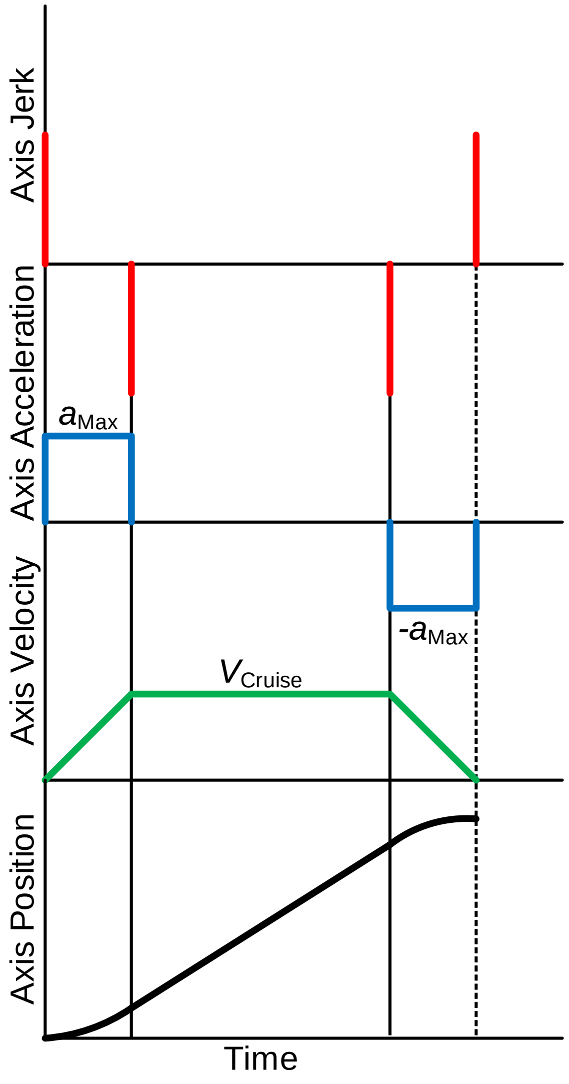
Direct Control Method: DVP to Servo Commands

- High speed communication of <u>servo setpoints</u> instead of <u>geometric primitives</u>



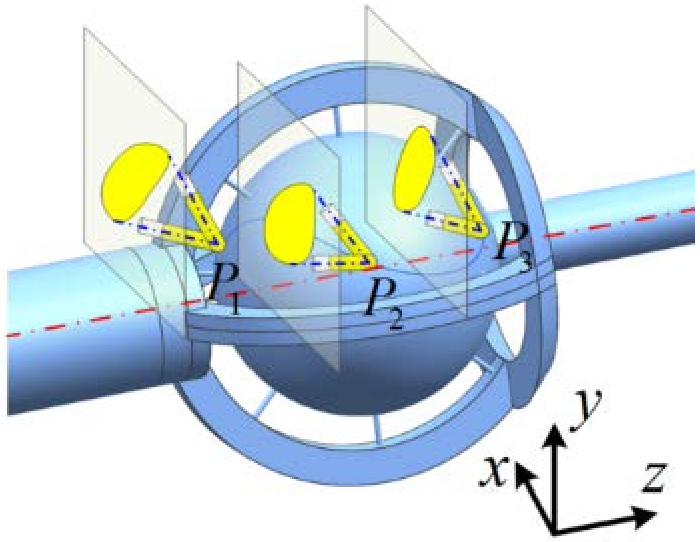
$$\Delta t_{i\to i+1} > 0 \qquad \frac{\Delta^2 P}{\Delta t_{i\to i+1}^{\;2}} \le A_{\mathrm{Max}}$$

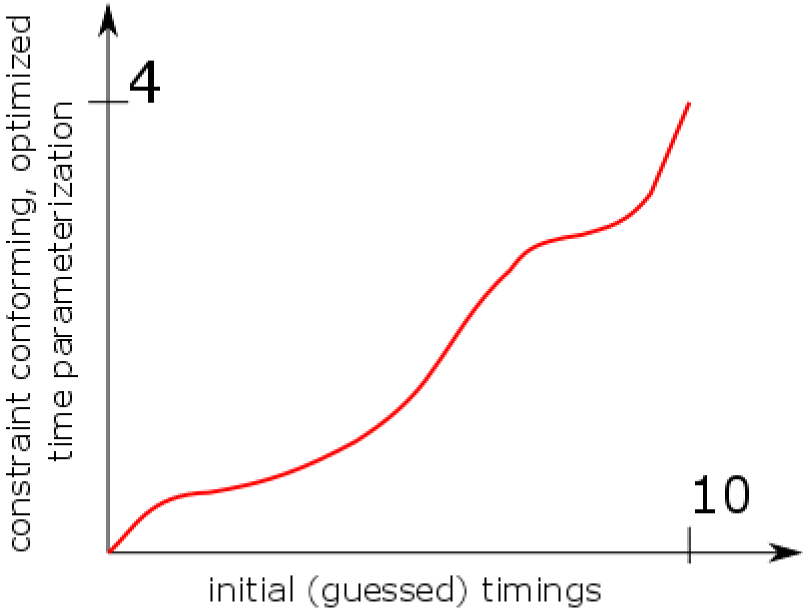
$$\frac{\Delta P}{\Delta t_{i\to i+1}} \le V_{\mathrm{Max}} \qquad \frac{\Delta^3 P}{\Delta t_{i\to i+1}^{\;3}} \le J_{\mathrm{Max}}$$

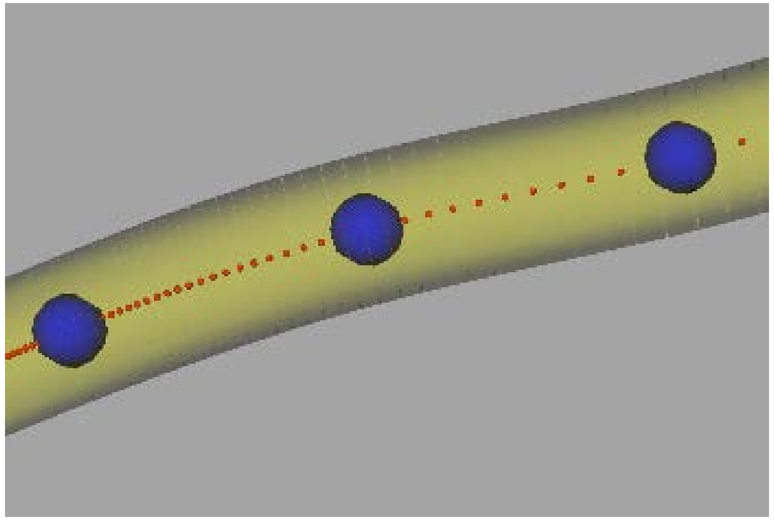


- Time-optimal path planning
  - Position constraints from CAM
  - Machine kinematic constraints
  - Spline fit to axis position commands
  - Multiple solutions of robotic path planning strategy

DVP to Servo Commands: Mathematical Formulation





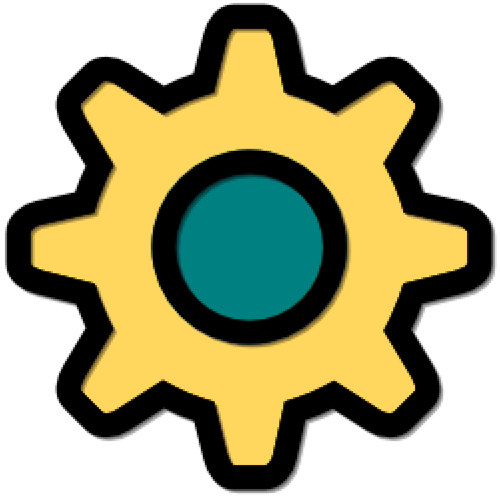


$$P_{T,i} = \begin{bmatrix} X_{T,i} \\ Y_{T,i} \\ Z_{T,i} \\ \theta_i \\ \phi_i \end{bmatrix} \quad P_{J,i} = \begin{bmatrix} X_{J,i} \\ Y_{J,i} \\ Z_{J,i} \\ A_i \\ B_i \end{bmatrix}$$

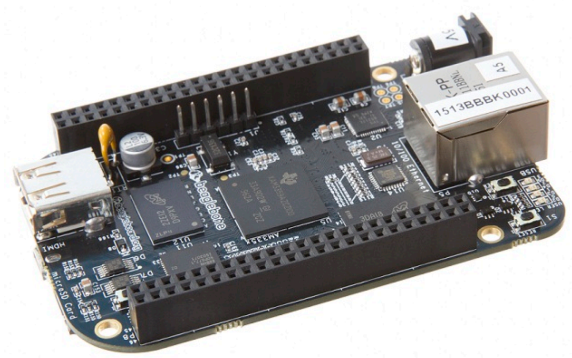
$$\boxed{r^* = \operatorname{argmin}_R \left\{ \int_0^1 \left(\frac{dr}{dt}\right)^{-1} du \;\middle|\; \begin{array}{l} \dfrac{dJ}{dt} \le V_{\text{Max}} \\ \dfrac{d^2J}{dt^2} \le A_{\text{Max}} \\ \dfrac{dV}{dt} \le \text{MRR}_{\text{Max}} \end{array} \right\}}$$

$$P'_{J,i} = \begin{bmatrix} X'_{J,i} \\ Y'_{J,i} \\ Z'_{J,i} \\ A'_i \\ B'_i \end{bmatrix}$$

Direct Servo Control Method: Requirements and Capabilities

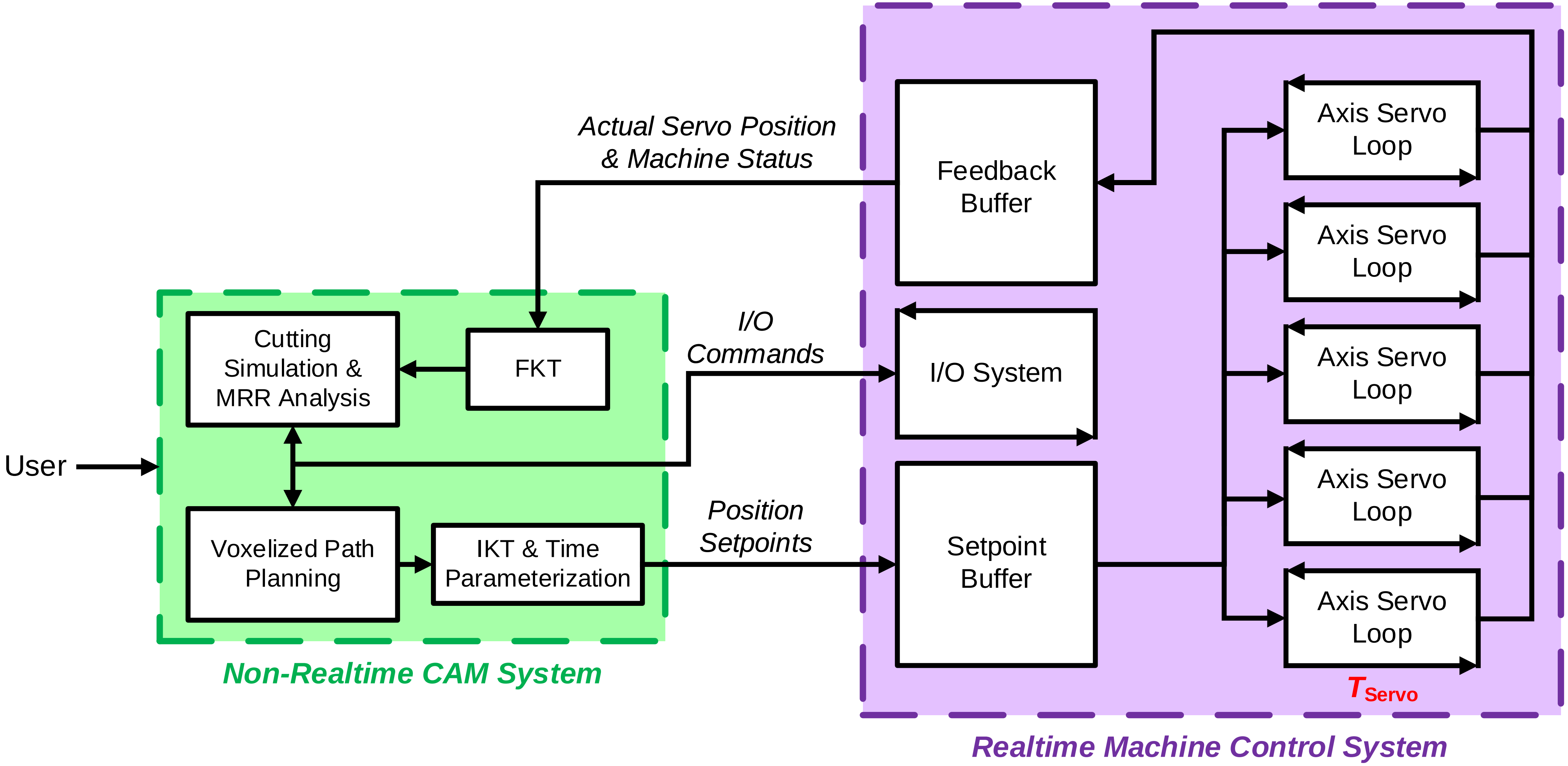


- Generalizable methodology for complete control of motion profiles directly from a CAM system
  - Pre-interpolated data with already-planned, optimized trajectories
  - Creation of joint space profiles at the servo rate with inverse kinematic transformation (IKT)
  - Use of open-source tools



- Dense feedback to CAM system to enable toolpath analysis and optimization
  - As-executed motion profile using forward kinematic transformation (FKT)
  - Positional derivatives along toolpath

Direct Servo Control from Voxel-Based CAM: Concept



Direct Servo Control from Voxel-Based CAM: Hardware Implementation

- <u>Current Platform</u>: PocketNC
  - $4000 5-axis desktop machine tool
  - Beaglebone Black, Machinekit
  - Desktop-sized open CNC research platform
- <u>WIP</u>: Mori Seiki Retrofit
- <u>Additional Capabilities</u>
  - Enhanced control of tool trajectory and MRR
  - Richer feedback information
  - Improved usability for complex machines, similar to AM

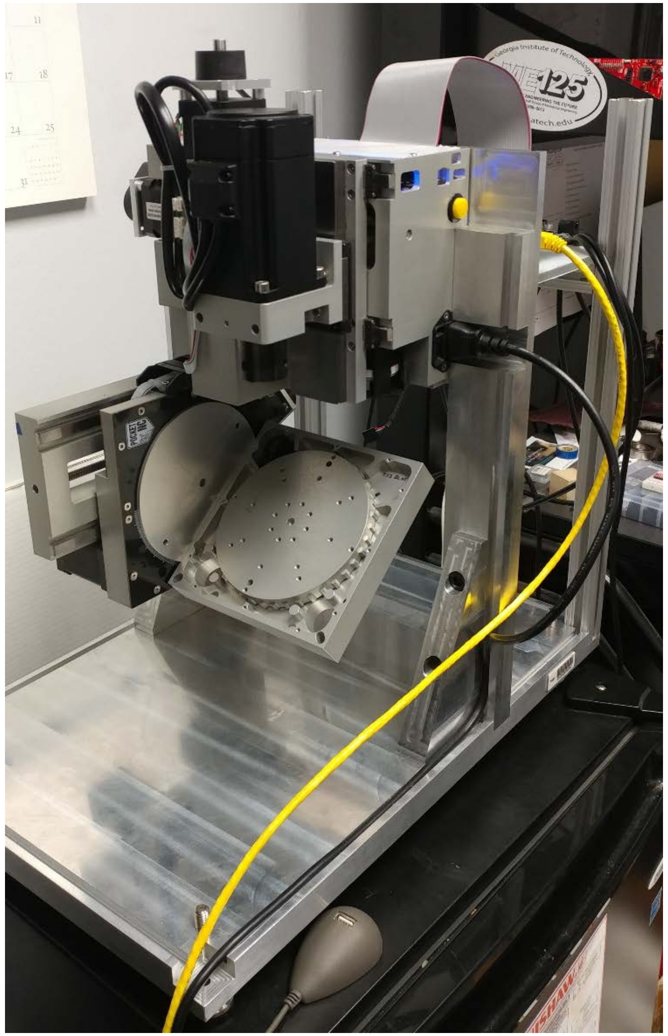
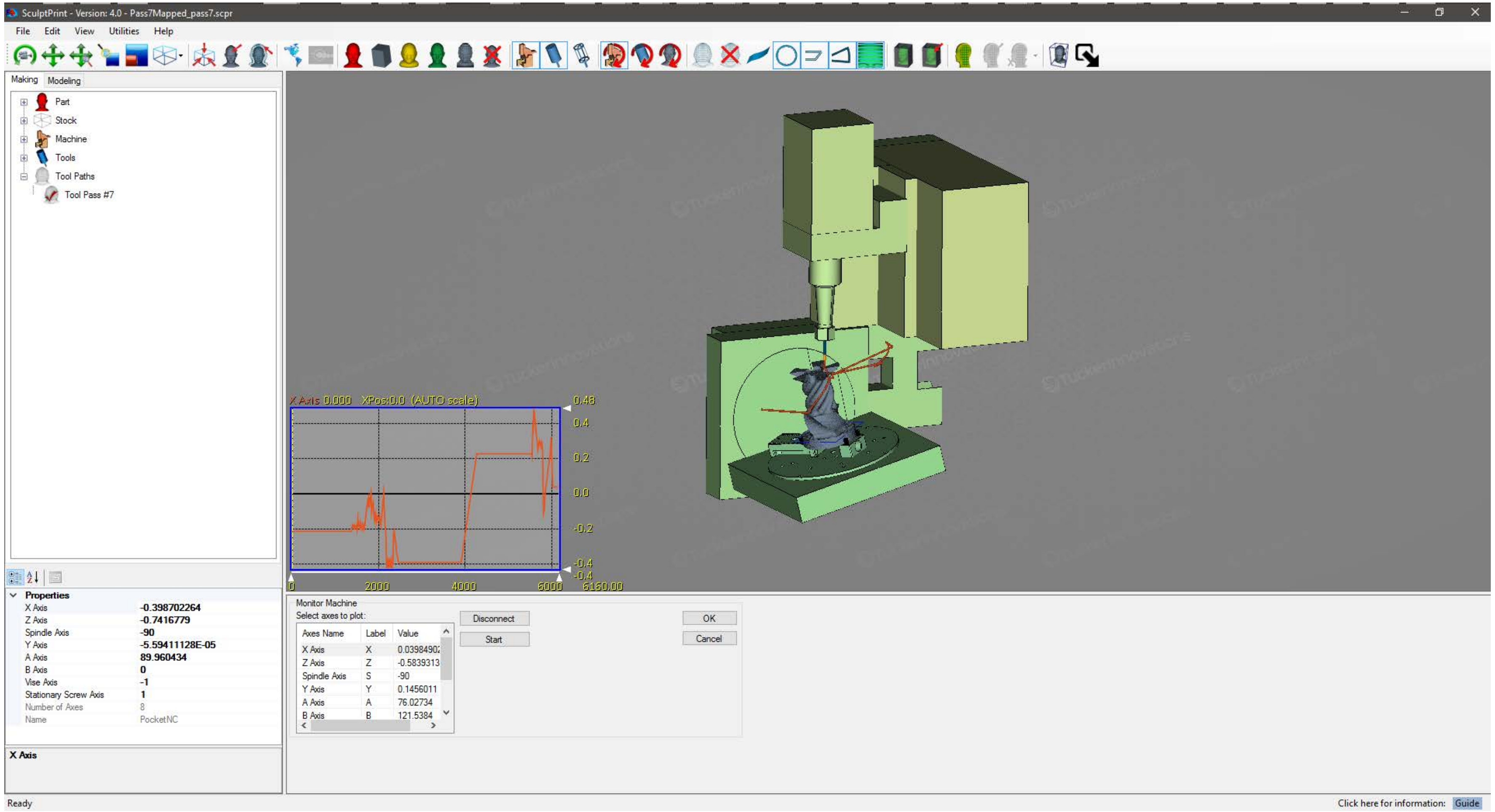


Photo Credit: Power Automation GmbH

Direct Servo Control from Voxel-Based CAM: The Machine Tool Digital Twin



Properties

X Axis	-0.398702264
Z Axis	-0.7416779
Spindle Axis	-90
Y Axis	-5.59411128E-05
A Axis	89.960434
B Axis	0
Vise Axis	-1
Stationary Screw Axis	1
Number of Axes	8
Name	PocketNC

X Axis

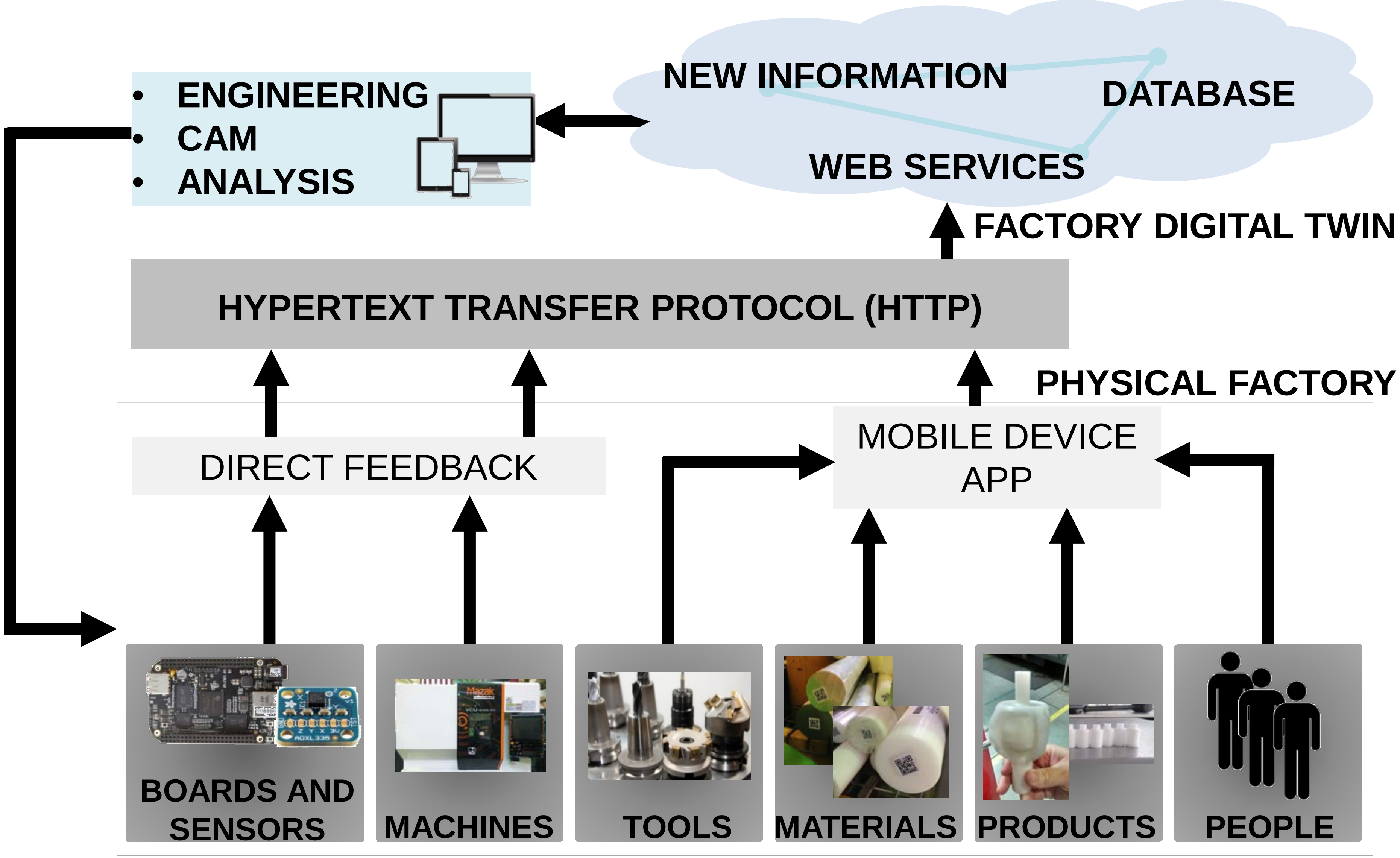
Ready

Monitor Machine

Select axes to plot:

Axes Name	Label	Value
X Axis	X	0.03984902
Z Axis	Z	-0.5839313
Spindle Axis	S	-90
Y Axis	Y	0.1456011
A Axis	A	76.02734
B Axis	B	121.5384

Integration into Complete Shop Floor Digital Twin



- ENGINEERING
- CAM
- ANALYSIS

NEW INFORMATION

DATABASE

WEB SERVICES

FACTORY DIGITAL TWIN

HYPERTEXT TRANSFER PROTOCOL (HTTP)

PHYSICAL FACTORY

DIRECT FEEDBACK

MOBILE DEVICE APP

BOARDS AND SENSORS

MACHINES

TOOLS

MATERIALS

PRODUCTS

PEOPLE

Implications & Conclusions

- Voxel-based CAM system
  - Intricate simulation, MRR analysis

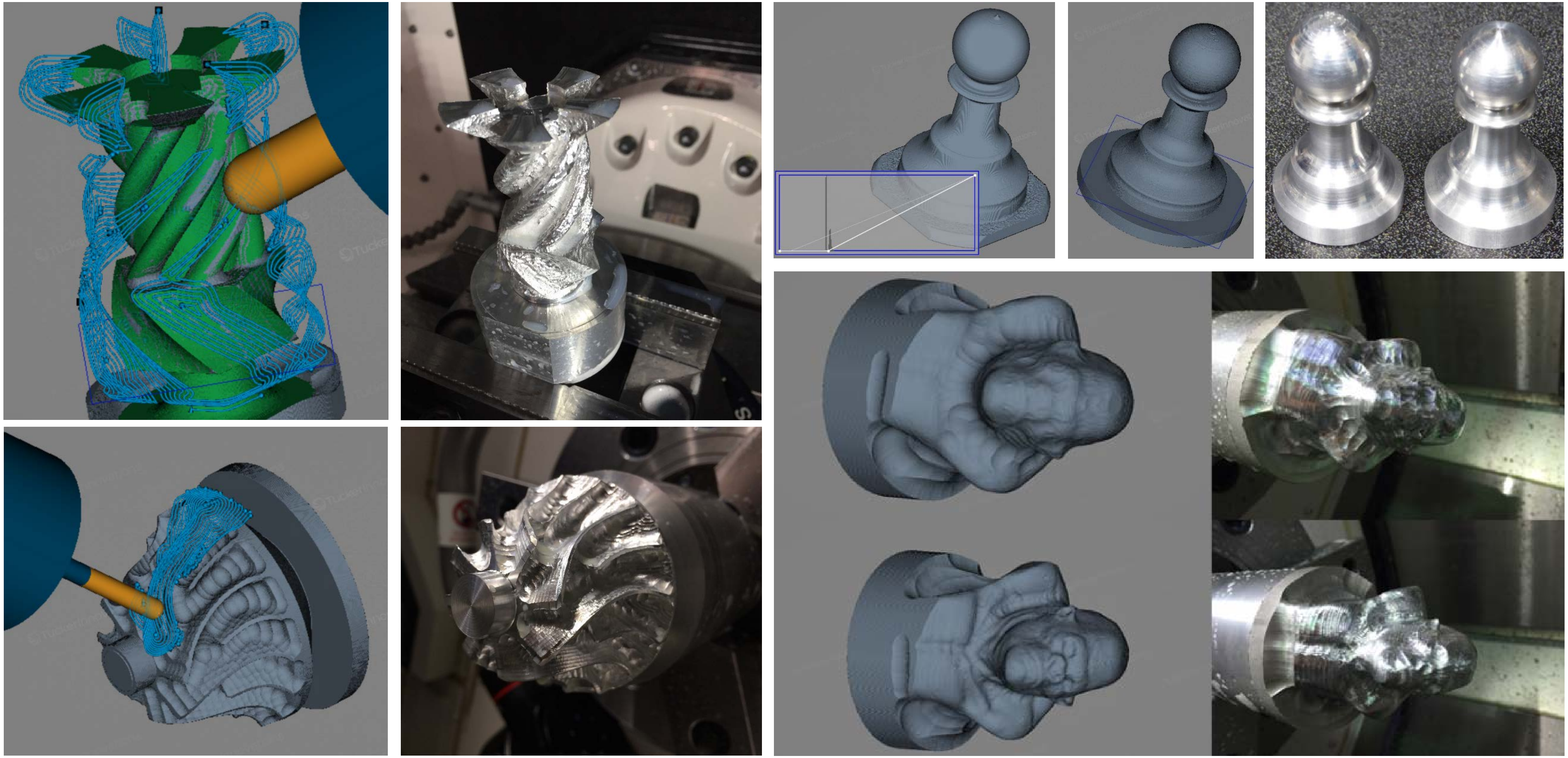
- Time parameterization of 5-axis toolpaths using position samples from voxel models
  - Minimum time path planning, kinematic and MRR limits

- New strategy for control and monitoring of a 5-axis machine tool directly from CAM
  - Enables usability similar to typical 3D printers
  - Allow for tighter integration into shop floor digital twin

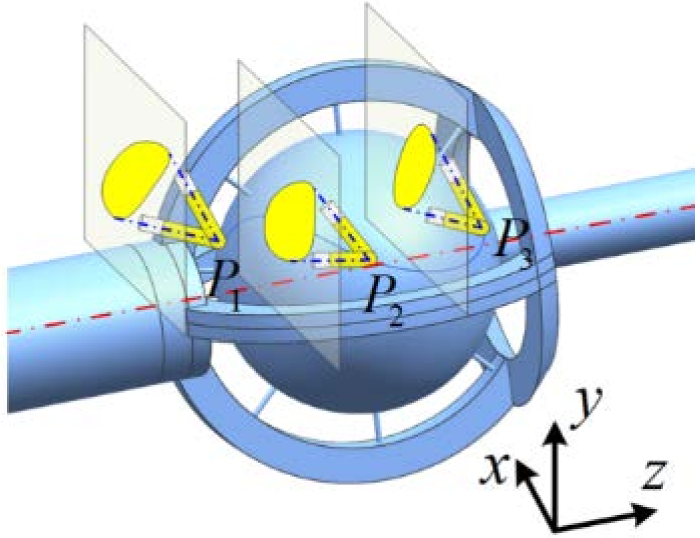
Questions?

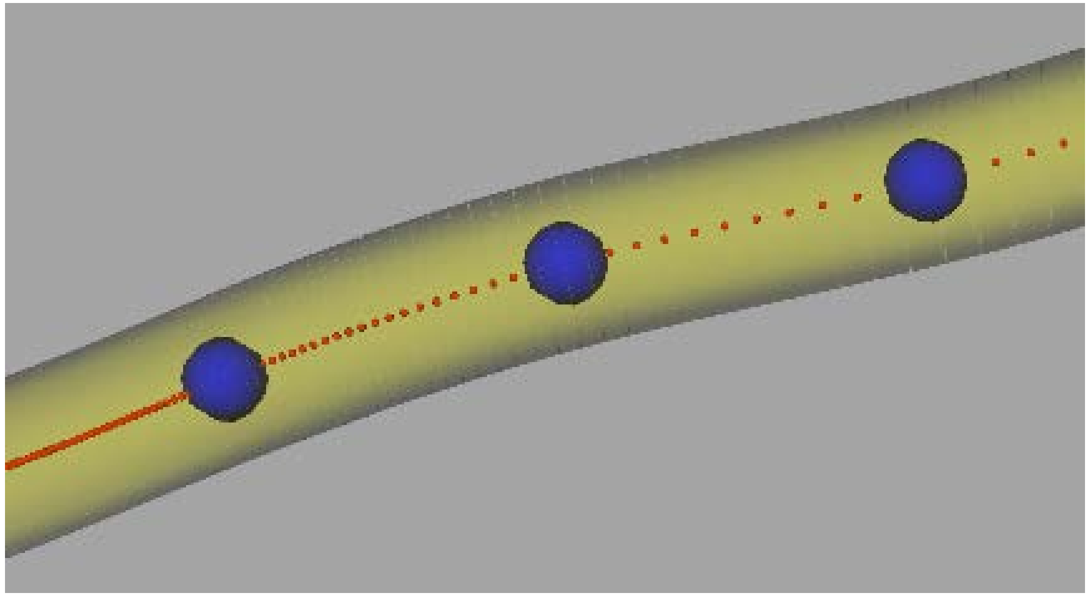
- Acknowledgements
  - NSF Grants CMMI-1329742, CMMI-1547093, and IIP-1631083
  - NSF GRFP DGE-1650044

DVP to Final Part



DVP to Servo Commands: Mathematical Formulation





$$P_{T,i} = \begin{bmatrix} X_{T,i} \\ Y_{T,i} \\ Z_{T,i} \\ \theta_i \\ \phi_i \end{bmatrix}$$

$$r^* = \operatorname{argmin}_R \left\{ \int_0^1 \left(\frac{dr}{dt} \right)^{-1} du \;\middle|\; \begin{array}{c} \frac{dJ}{dt} \leq V_{\text{Max}} \\ \frac{d^2J}{dt^2} \leq A_{\text{Max}} \\ \frac{dV}{dt} \leq \text{MRR}_{\text{Max}} \end{array} \right\}$$

$$P_{J,i} = \begin{bmatrix} X_{J,i} \\ Y_{J,i} \\ Z_{J,i} \\ A_i \\ B_i \end{bmatrix}$$

DVP to Servo Commands: Mathematical Formulation

$$P_{T,i} = \begin{bmatrix} X_{T,i} \\ Y_{T,i} \\ Z_{T,i} \\ \theta_i \\ \phi_i \end{bmatrix}$$

$$\underset{i \in \{1,\ldots,n\}}{M^{-1}\left(P_{T,i}\right)} = \begin{bmatrix} \sin\theta_i & -\sin\phi_i \cos\theta_i & \cos\phi_i \cos\theta_i \\ \cos\theta_i & \sin\phi_i \sin\theta_i & -\sin B \cos\phi_i \\ 0 & \cos\phi_i & \sin\phi_i \end{bmatrix}^{-1} \begin{bmatrix} -X_{T,i} - c_1 \cos\phi \cos\theta + c_2 \\ -Y_{T,i} + c_1 \sin\theta \cos\phi \\ -Z_{T,i} - c_1 \sin\phi - c_3 \end{bmatrix}$$

$$P_{J,i} = \begin{bmatrix} X_{J,i} \\ Y_{J,i} \\ Z_{J,i} \\ A_i \\ B_i \end{bmatrix}$$

$$u \in [0,1] \qquad r : t \to u, r^{-1} : u \to t \qquad R = \{ r \mid r \in C^1[\square \to \square] \}$$

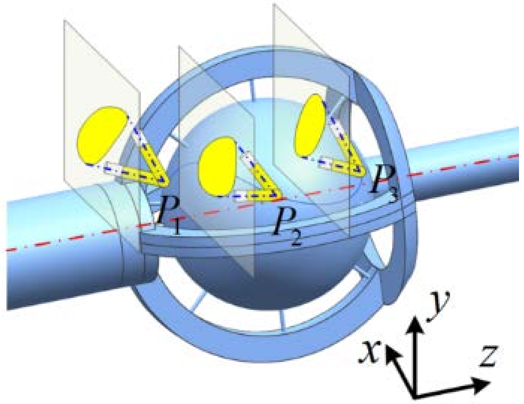
$$J : \begin{pmatrix} \square \to \square^5 \\ u \to J(u) \in P_J \end{pmatrix}$$

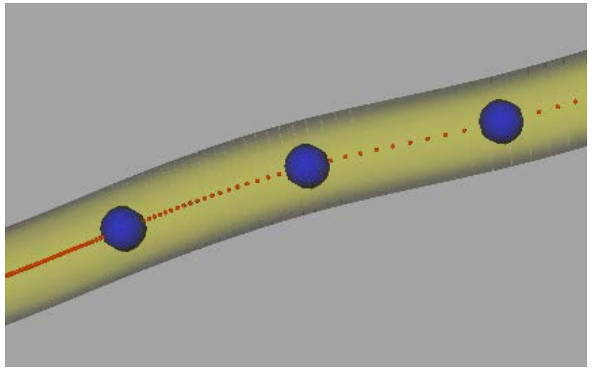
$$\boxed{r^* = \operatorname{argmin}_R \left\{ \int_0^1 \left(\frac{dr}{dt}\right)^{-1} du \;\middle|\; \begin{matrix} \dfrac{dJ}{dt} \le V_{\text{Max}} \\ \dfrac{d^2J}{dt^2} \le A_{\text{Max}} \\ \dfrac{dV}{dt} \le \text{MRR}_{\text{Max}} \end{matrix} \right\}}$$

$$T : \begin{pmatrix} \square \to \square^5 \\ u \to M(J(u)) \in P_T \end{pmatrix}$$

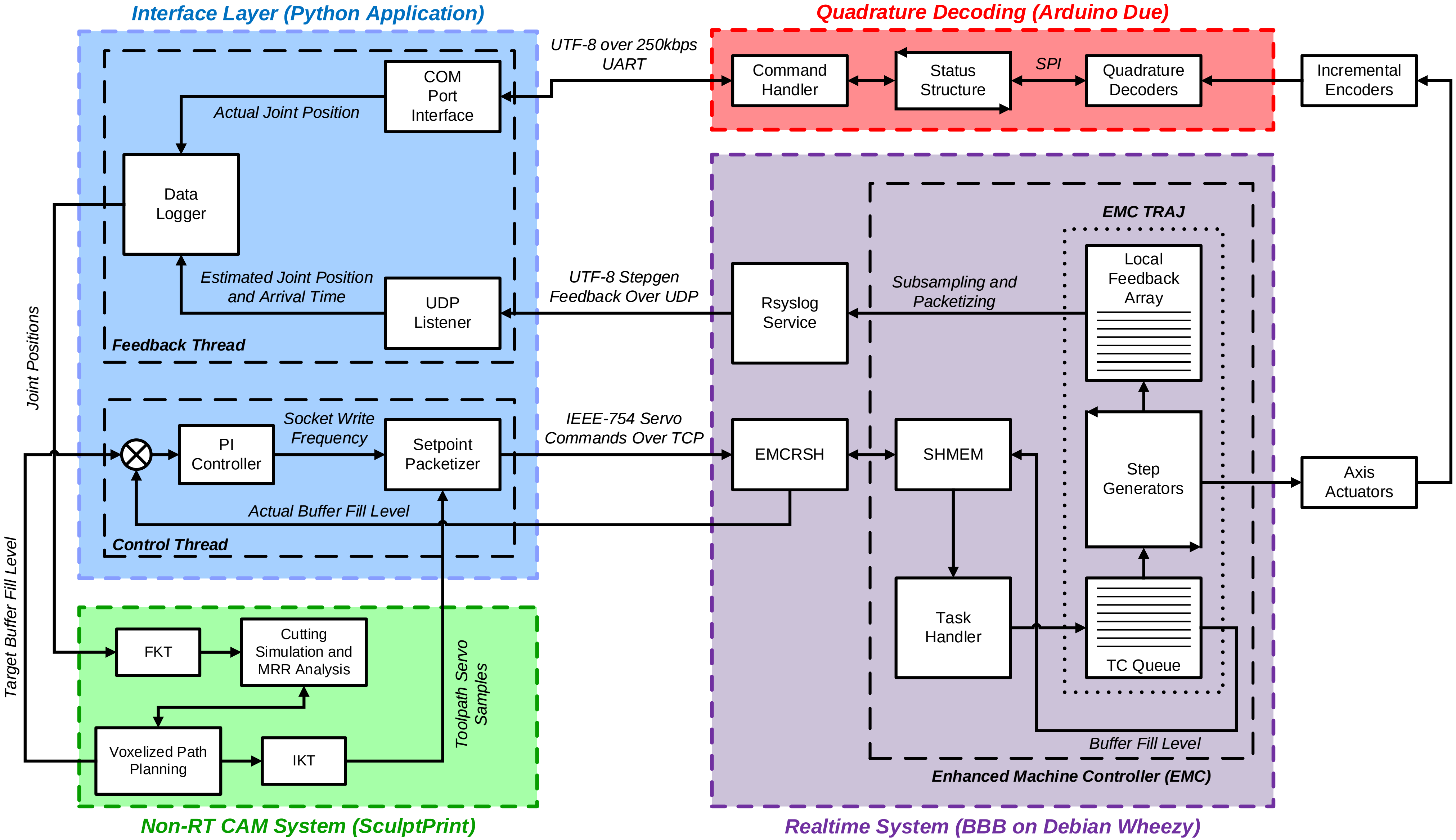
$$\frac{dJ}{dt} = \frac{dJ}{dr}\frac{dr}{dt} \qquad \frac{d^2J}{dt^2} = \frac{dJ}{dr}\frac{d^2r}{dt^2} + \frac{d^2J}{dr}\left(\frac{dr}{dt}\right)^2 \qquad \frac{dV}{dt} = \frac{dV}{dr}\frac{dr}{dt}$$

$$\boxed{T' : \begin{pmatrix} \square \to \square^5 \\ u \to M(J(u)) + \left[0\ 0\ 0\ \alpha(u)\ \beta(u)\right]^{ú} \end{pmatrix}}$$





Direct Servo Control from Voxel-Based CAM: Software



Interface Layer (Python Application)

COM Port Interface

Actual Joint Position

Data Logger

Estimated Joint Position and Arrival Time

UDP Listener

Feedback Thread

PI Controller

Socket Write Frequency

Setpoint Packetizer

Actual Buffer Fill Level

Control Thread

Joint Positions

Target Buffer Fill Level

FKT

Cutting Simulation and MRR Analysis

Voxelized Path Planning

IKT

Toolpath Servo Samples

Non-RT CAM System (SculptPrint)

UTF-8 over 250kbps UART

UTF-8 Stepgen Feedback Over UDP

IEEE-754 Servo Commands Over TCP

Quadrature Decoding (Arduino Due)

Command Handler

Status Structure

SPI

Quadrature Decoders

Incremental Encoders

Axis Actuators

Rsyslog Service

Subsampling and Packetizing

EMC TRAJ

Local Feedback Array

Step Generators

EMCRSH

SHMEM

Task Handler

TC Queue

Buffer Fill Level

Enhanced Machine Controller (EMC)

Realtime System (BBB on Debian Wheezy)