HVACSIM$^+$ Building Systems and Equipment Simulation Program Reference Manual

Daniel R. Clark

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Building Equipment Division
Gaithersburg, MD 20899

NBSIR 84-2996

HVACSIM$^+$ BUILDING SYSTEMS AND EQUIPMENT SIMULATION PROGRAM REFERENCE MANUAL

Daniel R. Clark

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Building Equipment Division
Gaithersburg, MD 20899

January 1985

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*
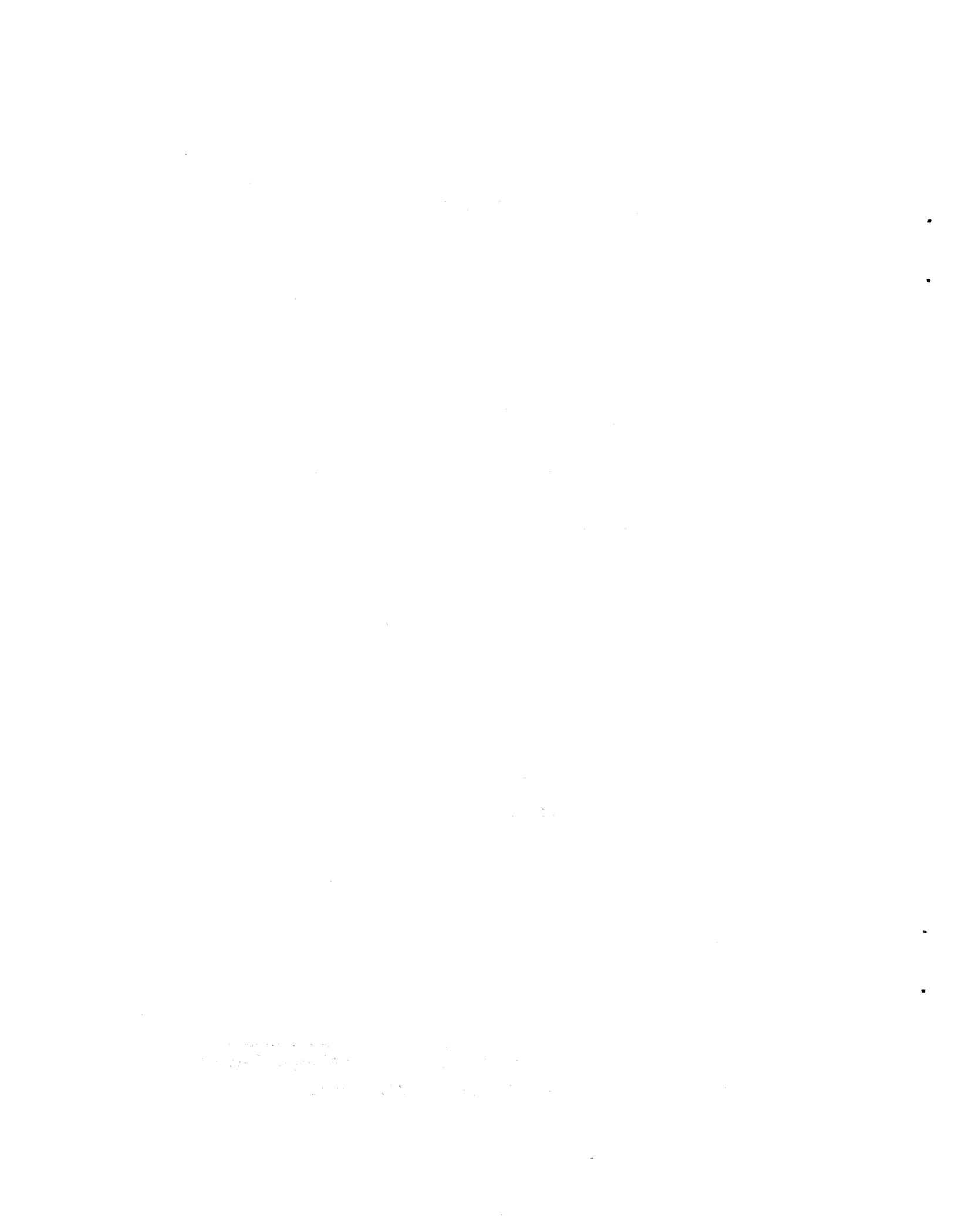
Abstract

HVACSIM$^+$ is a modular, non-proprietary computer simulation package developed at the National Bureau of Standards, designed to allow detailed simulation of entire building systems: the heating, ventilating, and air conditioning (HVAC) system, the equipment control system, the building shell, the physical plant, and the dynamic interactions among these subsystems.

The HVACSIM$^+$ package consists of a main simulation program, a library of subroutines containing mathematical models of building energy system components, and two programs used in preparing a description of the system to be simulated. Models representing the components of a physical plant, such as boilers and chillers, and a model representing a multizone building, are under development and will be added to the HVACSIM$^+$ package as they become available.

This document describes the operation, capabilities, and limitations of the main simulation program, presents descriptions of the existing library of component models for simulating HVAC systems and their controls, includes theoretical or experimental validation of the models wherever possible, and explains the procedure by which new component models may be written and added to the library.

KEY WORDS: building dynamics; building simulation; building system modeling; computer simulation techniques; control dynamics; dynamic modeling of building systems; dynamic performance of building systems; dynamic simulation; HVAC system simulation; HVACSIM+

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

vi

Acknowledgments

1. INTRODUCTION AND OVERVIEW

1.0 INTRODUCTION

HVACSIM$^+$ is a non-proprietary computer simulation package developed at the National Bureau of Standards (NBS), designed to allow detailed simulation of entire building systems: the heating, ventilating, and air conditioning (HVAC) system, the equipment control system, the building shell, the physical plant, and the dynamic interactions among these subsystems.

The HVACSIM$^+$ package consists of a main simulation program called MODSIM, a library of subroutines containing mathematical models of building energy system components, an interactive 'front end' program called HVACGEN which is used to create a description of the system to be simulated, and a program called SLIMCON which processes the output of HVACGEN and converts it into the form required by MODSIM.

MODSIM by itself is a general-purpose simulation program, employing advanced equation solving techniques and a hierarchical, modular approach to solve large sets of non-linear differential and algebraic equations. In principle, MODSIM could be used to simulate any system which can be represented as a set of discrete, interconnected components. The inclusion of a library of subroutines describing HVAC system components makes MODSIM specific to building energy systems. Models representing the components of a physical plant, such as boilers and chillers, and a model representing a multi-zone building, are under development and will be added to the HVACSIM$^+$ package as they become available.

This document is intended to serve as a reference manual for HVACSIM$^+$. A separate document [1] describes the operation of HVACGEN and the installation and use of the HVACSIM$^+$ package, and provides a series of examples to serve as a tutorial in the use of HVACSIM$^+$.

The remainder of this chapter describes the operation, capabilities, and limitations of MODSIM. Most of this information has been drawn from reference [2], which was written by the author of the MODSIM program. A brief summary of the entire simulation package is also provided.

Chapter two explains the interface between MODSIM and the individual component models. The purpose of this chapter is to describe how component models may be written and added to the library.

Chapter three describes a number of functions and subroutines which are used by some of the component models and are available for use by future component models. These routines represent transport delays, hysteresis effects, Bessel functions, linear least squares curve fitting, dry fin efficiencies, and the properties of fluids.

Chapter four provides descriptions of each of the component models presently included in HVACSIM$^+$. Wherever possible, theoretical or experimental verification of the models has been included.

1.1 AN OVERVIEW OF MODSIM

1.1.1 <u>Types, Units, Blocks, and Superblocks</u>

MODSIM stands for MODular SIMulation. Many ideas for the design of MODSIM were borrowed from the transient systems simulation program TRNSYS, which was developed at the University of Wisconsin Solar Energy Laboratory [3]. Both programs use separate subroutines, called TYPE subroutines, to represent each kind of system component. When a simulation is constructed, the components modeled by the TYPE subroutines are linked together. Two or more components of the same kind, such as two pumps, are modeled by the same TYPE subroutine. The distinction between components of the same kind is maintained by assigning a UNIT number to each component in the simulation. Each unit has distinct inputs, outputs, parameters, and saved workspace. As far as a user is concerned, a unit can be viewed as a black box which takes a set of inputs and produces a set of outputs.

A major difference between MODSIM and many other building simulation programs is the method used to obtain a self-consistent solution at each time step. Instead of the commonly used method of subsequent substitution, MODSIM uses a simultaneous nonlinear equation-solving package called SNSQ [4], which is based on a modification of the Powell hybrid method [5].

Large scale simulations require simultaneous solution of a large number of algebraic and differential equations. When the number of equations is much greater than about twenty or thirty, direct solution of the equation set using a program such as SNSQ becomes prohibitively time-consuming. MODSIM uses a hierarchical approach to this problem, by partitioning large sets of equations into smaller subsets. Between the entire simulation and the individual units, two intermediate levels of structure are defined: BLOCKS and SUPERBLOCKS.

A block is simply a set of units, more or less arbitrarily defined by the user. Usually, blocks should be chosen to represent functional units of moderate size, such as an air handler or the elements of a control loop. The interconnections among the units in a block define a set of simultaneous equations which is solved by the equation solver SNSQ. Some of the inputs to the units in a block will be solved simultaneously within the block. Any unit inputs which are not solved simultaneously are called 'block inputs.' These block inputs can be viewed as boundary conditions for the set of simultaneous equations being solved within the block. All the outputs of the units in a block can also be viewed as block outputs. The values of the block outputs will depend on the values of the block inputs. We can now view a block as a black box which takes a set of block inputs and produces a set of block outputs.

The next level of structure is the superblock, which consists of a set of blocks. Blocks within a superblock are exactly analogous to units within a block. Connections among the blocks in a superblock define a set of simultaneous equations which is solved by SNSQ. A superblock forms a functional unit of a building system, and a simulation may consist of several superblocks. For example, one superblock might contain several air handlers, another might consist of air conditioning zones within a building, and a third might represent the building shell. However, MODSIM assumes that the couplings among superblocks are weak enough so that each superblock can be treated as an independent subsystem. No simultaneous equations are defined between superblocks.

2

1.1.2 Time Steps and the Integration of Differential Equations

Differential equations limit the size of time steps which can be taken during the course of a simulation. If the time step is too large, inaccuracy or instability will result. This is particularly a problem when the components of a system have a wide range of characteristic times, or time constants, as is generally the case in building energy systems. A set of differential equations with widely varying time constants is referred to as a stiff set of equations.

The best integration algorithms for dealing with stiff equations are the so-called variable time step, variable order Gear algorithms [6,7]. These algorithms make use of backward differentiation formulas and the predictor-corrector method. The order of integration and the size of the time step are varied during the calculation to minimize the computation required for a specified degree of accuracy. The algorithm implemented in MODSIM is a modification of the algorithm described by Brayton [7].

Each superblock in a simulation is an independent subsystem in the sense that it proceeds forward in time independently. The time step and integration order are determined independently for each superblock. Thus, for example, rapid control actions in one superblock may cause that superblock to require very short time steps, but the time step taken by other superblocks can be much longer.

1.1.3 Time Dependent Boundary Conditions

In the context of MODSIM, a boundary condition or boundary variable is defined as a state variable which is external to the system being simulated, in the sense that it is not an output of any component of the system. A typical example is the outside air temperature. Boundary variables may be constant or time dependent. Any number of boundary variables may be specified as time dependent, and their values are stored in a data file. As a simulation progresses, data are read from this file and used to change the appropriate state variables.

The first column of the boundary variable data file must contain values of time. Time intervals need not be equal. MODSIM uses third order Lagrangian interpolation to find boundary values at the necessary times. The remaining columns contain values of the boundary variables at the specified times.

MODSIM begins all simulations at time zero. If the first time in the boundary variable file is 3600 seconds, for example, MODSIM will still simulate the time interval from 0 to 3600 seconds. During this time interval, the boundary variable values specified for the first time in the boundary file will be used. To avoid confusion, it is recommended that boundary data files always begin with time zero.

In certain cases, the change in a boundary variable may be discontinuous. A typical example is a set point change. In such a situation it is necessary to reset the differential equation integration algorithm at the time of the discontinuity. This is accomplished by including two data records with the same time. When MODSIM encounters two identical times in sequence in the boundary file, it integrates up to the repeated time and then reduces the

3

simulation time step to the minimum and reduces the integration algorithm to first order, beginning at the repeated time. This reset convention may be used to mark abrupt changes in any of the boundary conditions. There is no limit on the number of resets allowed in a simulation, but at least three records must occur between resets.

1.1.4 Initialization

Each time MODSIM is run, a complete description of the final state of the simulation is written to a file. An initialization option has been included which allows the final state of one simulation to be used as the initial state of another. This feature has two obvious uses. First, a series of simulations of a given system can be run from the same initial steady state, without having to repeat a computationally expensive startup transient for each run. Second, a long simulation can be run in several conveniently short pieces, with each piece beginning precisely where the previous piece ended.

As discussed in the previous section, MODSIM begins all simulations at time zero. Thus if a long simulation is to be run in several pieces, it will generally be necessary to prepare a separate boundary variable file (beginning at time zero) for each piece.

1.1.5 Report Generation

Two kinds of report are generated by MODSIM: a 'data file' containing raw data at each time step taken during the simulation, and a 'list file' containing a summary of the simulation configuration, any diagnostics generated during the computation, and an interpolated listing of the reported variables at equal time intervals. A reporting interval and a set of reported variables may be specified for each superblock. Reports are generated for each super-block individually as the simulation progresses. As discussed in the previous section, an output file containing the final state of the simulation is also written.

1.1.6 State Variable Freezing and Block Inactivation

During the course of a simulation, some of the state variables may reach steady state, i.e. cease to vary with time. If such a state variable is being solved simultaneously, a saving of computation time can be realized by removing it from the set of simultaneous equations. This is referred to as 'freezing' the variable. After a variable is frozen, it must be monitored so that it can be returned to the calculation, or 'unfrozen,' if necessary.

If all the simultaneous equations in a block are frozen, and all its block inputs are frozen, the block can be marked inactive. Once a block is inactivated, it is no longer necessary to monitor the frozen state variables in the block. A block is marked active as soon as one of its block inputs becomes unfrozen.

1.2 LIMITATIONS OF MODSIM

All simulation programs have inherent limitations, and MODSIM is no exception. In section 1.1, a number of techniques used to increase the computational efficiency of MODSIM over and above that of brute force calculation were described. Not surprisingly, these techniques involve tradeoffs. This

4

section discusses the drawbacks of the measures taken to achieve the goal of fast execution speed for very large scale simulations, and describes some additional measures taken to minimize the difficulties.

1.2.1 Convergence of the Equation Solver

Convergence is particularly a problem at the beginning of a simulation, when the equation solver must work from the set of user-supplied initial conditions chosen somewhat arbitrarily to start the calculation. If the initial conditions (or an intermediate solution) are too far from the final solution, the equation solver may fail to converge. In some cases, errors made by the equation solver can cause erratic solutions for several time steps, even if the solutions are convergent. The choice of the error tolerance for the equation solver is an important factor which influences this behavior. For further discussion of error tolerances, see section 1.3 below. The convergence properties of the equation solver are also strongly influenced by the block/superblock structure defined when a simulation is created.

The problem of slow and uncertain convergence to a self-consistent set of initial conditions is unavoidable, but the consequences can be minimized by the use of the initialization option. This permits a series of simulations to be run from the same initial state, without repeating an inaccurate and computationally expensive startup transient at the beginning of each run.

1.2.2 Limitations of Block Structure

It was stated above that the convergence properties of the equation solver depend on the block/superblock structure. This is not surprising since the block/superblock structure determines the equation sets to be solved during a simulation. This fact limits the flexibility of simulation construction. Some care must be exercised in defining blocks to obtain good convergence properties, particularly when control loops are involved. In general, it is good practice to have a control loop entirely contained in a single block.

1.2.3 Coupling Between Superblocks

Superblocks are assumed to be only weakly coupled. No simultaneous equations are defined between them, and they are allowed to evolve independently in time. Consequently, great care is required in dividing a system into superblocks.

When two superblocks are coupled, information is passed between them. If the coupling between superblocks involves a control loop, or variables which ideally should be solved simultaneously such as pressures and flow rates, the calculations tend to be inaccurate or unstable. To minimize this problem, special purpose subroutines have been designed for passing information (control signals and flow streams) between superblocks. However, the weak coupling of superblocks remains a fundamental limitation.

Difficulties can also arise due to the independent time evolution of superblocks, since the inputs to a superblock may change at a time when that superblock is not scheduled to be called. To mitigate this problem, a superblock input scanning option (INSOPT) is available. When this option is selected (INSOPT = 1), all superblock inputs are scanned after each time step.

5

If the inputs to a superblock which was not called during the time step have changed by an amount greater than an error tolerance, the superblock is called and its state is calculated based on the new values of its inputs. This option increases both the accuracy and the computational requirements of a simulation. When the input scanning option is not selected (INSOPT = 0), each superblock is called at times determined by the time step control algorithm, regardless of any changes in its inputs at intermediate times.

1.2.4 Time Step Control

Under certain conditions, the variable time step integration algorithm can produce erratic results. For the most part, these problems are eliminated by incorporating a mechanism for rejecting time steps which are too large, and by defining a minimum and a maximum time step. The choice of minimum and maximum time steps influences the accuracy and stability of the simulation, as well as the amount of computation required.

The time step selection algorithm depends on differential equation integration errors for its operation. A simulation which contains no differential equations will always proceed at the maximum time step. If all differential equations in a superblock become frozen during the course of a simulation, the length of each time step taken by the superblock is twice the length of the previous time step, until the maximum step size is reached.

1.2.5 State Variable Freezing and Unfreezing

A state variable is frozen if it changes less than a certain amount, determined by user-supplied error tolerances, from one time step to the next. Clearly, state variable freezing will introduce some errors into the computation. When a state variable is unfrozen within a block, the block equations are solved again with the unfrozen variable put back into the equation set. If this causes any additional variables to unfreeze, all of the variables solved simultaneously within the block are unfrozen and the computation is repeated a third time. Thus the unfreezing of variables forces extra computation, both by increasing the size of the equation set and by causing calculations to be repeated.

For superblock equations, three optional modes are implemented. The mode is determined by the value of the 'superblock variable unfreezing option,' IFZOPT. If a superblock equation is unfrozen and mode 0 is set, the corresponding state variable is not put back into the equation set until the next time step. If mode 1 is set, any unfrozen variables are put back into the superblock equation set and the calculation is repeated. If mode 2 is set, all superblock equations are put back into the equation set and the calculation is done again. The reason for these three modes is that superblock equations require much more computation than block equations. In some cases, depending on the block/superblock structure, the errors introduced by choosing mode 0 or 1 are acceptable, and these modes result in a significant saving of computation time. In other cases the simpler modes, particularly mode 0, result in serious errors, and mode 2 is required for satisfactory accuracy.

1.2.6 Size of Simulations

The size of simulations is limited by the dimensions assigned to a variety of arrays in MODSIM. These are not fundamental limitations, since dimen-

sion statements can be changed. As distributed, MODSIM has the following limits:

    15 inputs per unit
    15 outputs per unit
    10 differential equations per unit
    20 units per block
    50 inputs per block
    50 outputs per block
    30 simultaneous equations between units in a block
    10 blocks per superblock
    20 differential equations per superblock
    20 simultaneous equations between blocks in a superblock
    30 reported variables per superblock
   200 units per simulation
    50 blocks per simulation
    10 superblocks per simulation
    50 differential equations per simulation
   600 state variables per simulation
  1000 parameters per simulation
  1000 SAVED variables per simulation
    30 time-dependent boundary variables per simulation

1.3   AN OVERVIEW OF HVACSIM$^+$

This section provides a brief summary of the procedure for preparing and running simulations using HVACSIM$^+$. A more complete and detailed account can be found in reference [1].

First it is necessary to decide how the system to be simulated will be divided into units, blocks, and superblocks. A diagram of the system is indispensable at this stage. TYPE subroutines representing each unit must be selected, all state variables must be identified and numbered, and values for the parameters required by each component must be determined.

A description of the system to be simulated is then prepared using the interactive program HVACGEN. Information pertaining to the configuration of each TYPE component is contained in a file called TYPAR.DAT, which is read by HVACGEN. HVACGEN prompts the user to specify the TYPE of each component in the simulation, to provide index numbers for each state variable, and to enter values for all parameters. HVACGEN assigns a UNIT number to each component, to distinguish between components of the same type.

After all UNITS have been entered, HVACGEN asks the user to supply several error tolerances:

RTOLX, ATOLX – relative and absolute error tolerances. These tolerances are used in integrating differential equations and choosing the time step size, and in deciding when to freeze or unfreeze state variables. Some TYPE subroutines and utility functions also use these tolerances to decide when internal iterative procedures have converged sufficiently, and to decide when the freezing of differential equations should be permitted.

7

XTOL — an error tolerance used by the equation solver, SNSQ. The value of XTOL influences the convergence properties of the equation solver (see section 1.2.1). As a rule of thumb, XTOL should be greater than or equal to the sum of RTOLX and ATOLX.

TTIME — the time interval over which the integral of a differential equation must satisfy the error criterion defined by RTOLX and ATOLX.

HVACGEN supplies somewhat arbitrary default values for these error tolerances. The default tolerances are probably unnecessarily stringent for most purposes, and larger values could be used. However, it is important that XTOL be greater than or equal to the sum of RTOLX and ATOLX. In other words, the criterion for freezing and unfreezing variables must be more stringent than the convergence criterion for the equation solver. Otherwise variables may be frozen and unfrozen frequently and erratically, resulting in a slow and unstable simulation.

Next HVACGEN asks for initial values of all state variables, indices of time dependent boundary variables, a reporting interval for each superblock, and indices of variables to be reported.

It then prompts the user to select a superblock variable unfreezing option (IFZOPT) and a superblock input scan option (INSOPT). The variable freezing option determines the manner in which superblock equations are recalculated when a superblock equation unfreezes. Three modes are available, as described in section 1.2.5. The value of IFZOPT influences the strength of coupling among the blocks within a superblock, and has no effect on simulations consisting of a single block. The input scan option is mentioned in section 1.2.3. Enabling this option (mode 1) allows a superblock to be called whenever its inputs change, overriding the time step selection algorithm. When the input scan option is disabled (mode 0), a superblock is called only at times determined by the time step selection algorithm, regardless of any changes in the superblock inputs at intermediate times. The value of INSOPT influences the coupling between superblocks, and has no effect on simulations consisting of a single superblock.

HVACGEN also has extensive editing capabilities to facilitate the creation, testing, and modification of large system simulations. The operation and features of HVACGEN are the subject of a separate report [1]. The output of HVACGEN is a 'simulation work file.'

Before a simulation can be run, the program SLIMCON must be run. This program reads the simulation work file, performs some processing steps on the numbers, and writes a 'model definition file' which contains a complete system description in the form required by MODSIM.

When the simulation program is run, it prompts the user for the following information:

IPRINT — a non-zero value of IPRINT causes intermediate results of the equation solver to be written to the output 'list file.' This feature can be useful for debugging, but produces vast quantities of output. Normally IPRINT is set to zero, and only final values at each reporting time are written.

INIT - a non-zero value of INIT causes the initial state of the system to be read from a file. If INIT=0, the initial state vector in the model definition file defines the initial state of the system.

In addition, the program asks for a minimum time step, a maximum time step, and a stopping time. The choice of minimum and maximum time steps influences the accuracy and stability of the simulation, as well as the amount of computation required.

As the simulation progresses, MODSIM calls the TYPE subroutines representing the components of the system. Some of the TYPE subroutines call utility functions and subroutines described in chapter 3. The architecture of HVACSIM$^+$ is summarized in figure 1.3.1.
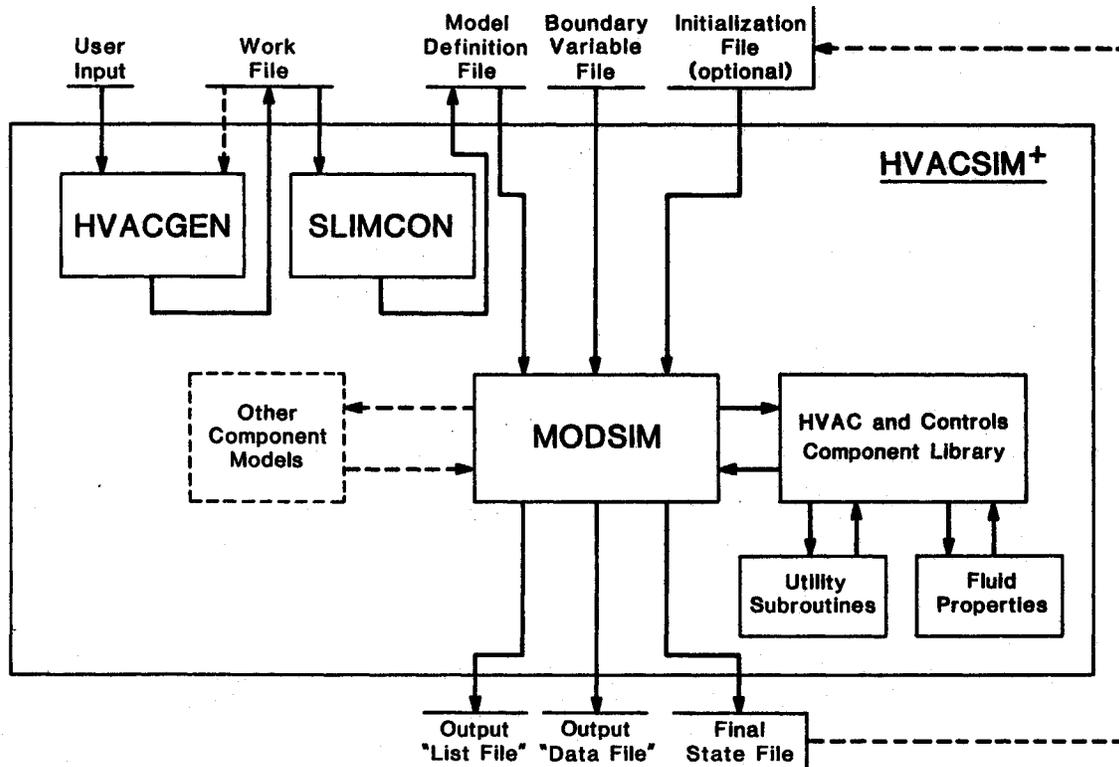
Figure 1.3.1   Architecture of the HVACSIM$^+$ package

2.0 FORMULATION OF COMPONENT MODELS

The library of component models described in this report includes most of the components of HVAC systems, including controllers for such systems. However, the library is by no means exhaustive, and situations will inevitably arise in which additional component models are needed.

In addition, it is worth mentioning that nothing in the main simulation program, MODSIM, limits it to building energy analysis. MODSIM is a general-purpose modular simulation program which can be used to study any dynamic system for which suitable component models are available. It is assumed that MODSIM will be used primarily as part of HVACSIM[+], by researchers involved in the analysis of HVAC systems and controls, though other applications of MODSIM are possible.

This section presents general information on the structure of component model TYPE subroutines consistent with the requirements of MODSIM. Development of mathematical models to represent a physical process is often a difficult undertaking, requiring a certain degree of expertise. However, this problem is not addressed here. Rather, emphasis is on features influencing the interface between TYPE subroutines and MODSIM. The TYPE 9 linear valve model described in section 4.9 is used as an example to point out important features of TYPE subroutines in general. A listing of the FORTRAN source code for the TYPE 9 subroutine appears at the end of section 2.2, and is referred to throughout this section. The reader should review section 4.9 before proceeding through the example. Familiarity with FORTRAN programming is assumed. HVACSIM[+] (including the TYPE subroutines) is written in ANSI Standard X3.9-1978 FORTRAN, which is compatible with FORTRAN VII compilers.

2.1 SOLUTION OF DIFFERENTIAL EQUATIONS

The model definition file informs MODSIM of the number of differential equations in each UNIT of the simulation. Provision has been made in the setup program, HVACGEN, to allow the number of differential equations in a given component to be determined by the values of one or more parameters. Thus different UNITS of a given TYPE may have different numbers of differential equations. This feature is discussed further in section 2.3.

For a UNIT with N differential equations, MODSIM expects the first N outputs to be first derivatives of the N dependent variables, rather than the dependent variables themselves. The dependent variables may be inputs to the component, and to any other component in the system. No limiting assumptions are made about the form of differential equations. The outputs must be first order derivatives, but this is not a fundamental limitation since higher order differential equations can be factored into a set of coupled first order differential equations.

2.2 EXAMPLE: LINE-BY-LINE DESCRIPTION

Line 1:

```
1          SUBROUTINE TYPE9(XIN,OUT,PAR,SAVED,IOSTAT)
```

The symbolic name of each TYPE subroutine is of the form TYPEn, where n is an integer less than or equal to 99. TYPE subroutines are called by

subroutine SELECT, which must include a CALL statement for each TYPE subroutine. The arguments of the subroutine are defined as follows:

XIN : an array of time-dependent inputs to the component
OUT : an array of time-dependent outputs from the component
PAR : an array of constant parameters defining the characterisics of the component
SAVED : a saved workspace specific to a particular implementation or UNIT of a component
IOSTAT : an array of information related to the freezing and unfreezing of state variables

Lines 5-6:

```
5         REAL K,XIN(4),OUT(2),PAR(4),SAVED(5)
6         INTEGER IOSTAT(4)
```

Since the arguments of the subroutine are all arrays, they must be dimensioned within the subroutine. IOSTAT has the same dimension as either XIN or OUT, whichever is larger. The use of IOSTAT is discussed below. SAVED need not be dimensioned if the saved workspace is not used.

Lines 8-10:

```
8         COMMON/CHRONO/ TIME,TSTEP,TTIME,TMIN,ITIME
9         COMMON/SOSCOM/ RTOLX,ATOLX,XTOL
10        COMMON/XINIT/ INIT,NSAVED
```

MODSIM uses common blocks extensively to pass information internally. Three common blocks are invoked by the TYPE9 subroutine.

CHRONO contains the following information:

TIME : the current simulation time
TSTEP : the current simulation time step
TTIME : a time interval associated with error criteria for integrating differential equations
TMIN : the minimum allowable simulation time step
ITIME : the number of time steps taken since the beginning of the simulation

SOSCOM contains the following information:

RTOLX : a relative error tolerance
ATOLX : an absolute error tolerance, used together with RTOLX to calculate error criteria for differential equations and bounds for variable freezing
XTOL : an error tolerance for simultaneous equation solving

XINIT contains the following information:

INIT : a flag determining whether the simulation is to be initialized from the initial state vector (INIT=0) or from an initialization file (INIT≠0)
NSAVED: the number of saved workspace variables in the entire simulation

11

```
12          P2=XIN(1)
13          W=XIN(2)
14          C=XIN(3)
15          CV=XIN(4)
16          K=PAR(1)
17          TCON=PAR(2)
18          CL=PAR(3)
19          HYS=PAR(4)
```

P2 is the water pressure at the valve outlet, W is the water mass flow
rate, C is a control signal specifying a requested valve position, and CV is
the actual position of the actuator. The value of CV is obtained from the
first output of this subroutine, as discussed below after lines 25 and 39.
The four parameters are the flow resistance when the valve is open, the time
constant of the actuator, a leakage parameter, and a hysteresis parameter,
respectively. Note that mnemonics for the parameters must be assigned on each
call to the subroutine, even though they do not change with time, since this
subroutine may be used to represent two or more valves with different parame-
ters in the same simulation.

Lines 21-24:

```
21          IF(C.LT.0.) C=0.
22          IF(C.GT.1.0) C=1.0
23          IF(CV.LT.0.) CV=0.
24          IF(CV.GT.1.0) CV=1.0
```

In the process of obtaining a solution, the MODSIM equation solver may
occasionally call the subroutine with values of C or CV greater than one or
less than zero. These lines enforce the requirement that the control signals
C and CV must lie between 0 and 1, inclusive.

Lines 25, 39, and 46:

```
25          IF(TCON.GE.1.0) GOTO 1000
:
39  1000    DCVDT=(C-CV)/TCON
:
46          OUT(1)=DCVDT
```

The description of the TYPE9 component in TYPAR.DAT informs MODSIM that
the component has one differential equation if the value of the second parame-
ter is greater than or equal to one, and no differential equations otherwise.
Thus if the time constant of the actuator is one second or more, lines 26-38
are skipped and the dynamics of the actuator are represented by line 39. Since
MODSIM expects one differential equation, it assumes the first output is the
time derivative of a state variable, rather than the state variable itself.
MODSIM takes care of integrating the derivative to evaluate the state vari-
able, in this case CV. If the actuator time constant is less than one second,
the differential equation is solved within the subroutine and the first output
is the value of CV, rather than its derivative.

Lines 26-29 and 45:

```
26              IF(ITIME.GT.1) GOTO 3000
27              IF(INIT.EQ.0)SAVED(4)=CV
28              IF(INIT.EQ.0 .OR. SAVED(3).GT.TIME) SAVED(3)=0.
29     3000     IF(TIME.GT.SAVED(3)) SAVED(5)=SAVED(4)
:
45              SAVED(4)=CV
```

The SAVED array is generally used to store previous values of variables whose history is required. Normally two spaces in the SAVED array are needed for each variable to be saved, plus one space for storing the simulation time.

In this particular case, SAVED(4) holds the value of CV from the previous call to the UNIT, which may be called repeatedly at a single simulation time. SAVED(5) holds the final value of CV from the last call of the previous time step, while SAVED(3) holds the simulation time from the previous call. Thus line 45 saves CV on each call, and line 29 updates SAVED(5) only on the first call of a new timestep. The utility function HYSTER, called in line 42, takes care of maintaining the simulation time in SAVED(3).

Line 26 serves to identify the first time step of the simulation. If an initialization file is used (INIT $\neq$ 0), the saved workspace initially contains final values from the initialization run. In this case SAVED(4) should not be changed, but SAVED(3) will initially contain the final time of the initialization run, and must be reset to zero on the first call of the first time step, as is done in line 28. If no initialization file is used, both SAVED(3) and SAVED(4) are initialized in lines 27 and 28.

Lines 30-38 and 46:

```
30              DCVDT=C
31              CV=C
32              A=TCON/TSTEP
33              IF(A.LT.0.05) GOTO 2000
34              DC=C-SAVED(5)
35              IF(ABS(DC).LT.1.E-10) GOTO 2000
36              CV=C-DC*EXP(-1./A)
37              DCVDT=CV
38              GOTO 2000
:
46              OUT(1)=DCVDT
```

If the actuator time constant is less than one second, the differential equation in line 39 is solved in lines 30-38. The resulting value of CV is assigned to the local variable DCVDT as a matter of convenience, allowing line 46 to be used regardless of whether the differential equation is solved inside or outside the subroutine. Note that care has been taken to avoid division by zero when the time constant is zero, and to avoid causing FORTRAN overflow or underflow errors when CV is very nearly equal to C.

<u>Lines 40-43 and 46-47</u>:

```
40  2000      S=1.
41            IF(W.NE.0.) S=SIGN(1.,W)
42            CH=HYSTER(CV,SAVED(1),SAVED(2),SAVED(3),HYS)
43            P1=P2+S*K*(W/((1.-CL)*CH+CL))**2
    :
46            OUT(1)=DCVDT
47            OUT(2)=P1
```

These lines complete the mathematical description of a linear valve with hysteresis, as described in Section 4.9, and place the results in the OUT array. The utility function HYSTER is described in Chapter 3.

<u>Lines 7 and 48-51</u>:

```
7             LOGICAL CON
48            CON=(IOSTAT(3).LT.-1).OR.(IOSTAT(3).EQ.2)
49            IOSTAT(1)=0
50            IF((ABS(C-CV).LE.RTOLX*ABS(CV)+ATOLX).AND.CON) IOSTAT(1)=1
51            IOSTAT(2)=1
```

IOSTAT is an input/output status vector. On entry this vector contains the status of the input variables. Possible values of IOSTAT and their meanings are as follows:

$$
\text{IOSTAT} = \begin{cases}
-4 & \text{time independent boundary condition} \\
-3 & \text{time dependent boundary condition} \\
-2 & \text{frozen output (not solved simultaneously)} \\
-1 & \text{output (not solved simultaneously)} \\
0 & \text{solved simultaneously, may not be frozen} \\
1 & \text{solved simultaneously, may be frozen} \\
2 & \text{frozen} \\
3 & \text{unfrozen}
\end{cases}
$$

On exit it contains flags which enable (IOSTAT=1) or disable (IOSTAT=0) variable freezing for the output variables. In general it is recommended that freezing be enabled for algebraic outputs, as in line 51. However, care must be taken in establishing criteria for enabling the freezing of dependent variables in differential equations. In line 48, the logical variable CON is true if the input control signal C is either a frozen variable or a boundary condition. In line 50, freezing of the dependent variable CV is enabled only if CON is true and CV is within error tolerances of C. This is to prevent the differential equation from freezing if CV happens to be nearly equal to C at a time when C is changing rapidly. IOSTAT should always be set to either 0 or 1 for each output.

14

```
1             SUBROUTINE TYPE9(XIN,OUT,PAR,SAVED,IOSTAT)
2   C
3   C         Linear valve model.
4   C
5             REAL K,XIN(4),OUT(2),PAR(4),SAVED(5)
6             INTEGER IOSTAT(4)
7             LOGICAL CON
8             COMMON/CHRONO/ TIME,TSTEP,TTIME,TMIN,ITIME
9             COMMON/SOSCOM/ RTOLX,ATOLX,XTOL
10            COMMON/XINIT/ INIT,NSAVED
11  C
12            P2=XIN(1)
13            W=XIN(2)
14            C=XIN(3)
15            CV=XIN(4)
16            K=PAR(1)
17            TCON=PAR(2)
18            CL=PAR(3)
19            HYS=PAR(4)
20  C
21            IF(C.LT.0.) C=0.
22            IF(C.GT.1.0) C=1.0
23            IF(CV.LT.0.) CV=0.
24            IF(CV.GT.1.0) CV=1.0
25            IF(TCON.GE.1.0) GOTO 1000
26            IF(ITIME.GT.1) GOTO 3000
27            IF(INIT.EQ.0)SAVED(4)=CV
28            IF(INIT.EQ.0 .OR. SAVED(3).GT.TIME) SAVED(3)=0.
29  3000      IF(TIME.GT.SAVED(3)) SAVED(5)=SAVED(4)
30            DCVDT=C
31            CV=C
32            A=TCON/TSTEP
33            IF(A.LT.0.05) GOTO 2000
34            DC=C-SAVED(5)
35            IF(ABS(DC).LT.1.E-10) GOTO 2000
36            CV=C-DC*EXP(-1./A)
37            DCVDT=CV
38            GOTO 2000
39  1000      DCVDT=(C-CV)/TCON
40  2000      S=1.
41            IF(W.NE.0.) S=SIGN(1.,W)
42            CH=HYSTER(CV,SAVED(1),SAVED(2),SAVED(3),HYS)
43            P1=P2+S*K*(W/((1.-CL)*CH+CL))**2
44  C
45            SAVED(4)=CV
46            OUT(1)=DCVDT
47            OUT(2)=P1
48            CON=(IOSTAT(3).LT.-1).OR.(IOSTAT(3).EQ.2)
49            IOSTAT(1)=0
50            IF((ABS(C-CV).LE.RTOLX*ABS(CV)+ATOLX).AND.CON) IOSTAT(1)=1
51            IOSTAT(2)=1
52            RETURN
53            END
```

2.3 INSTALLATION OF COMPONENT MODELS

Before a new TYPE subroutine can be used, a CALL statement for the subroutine must be added to subroutine SELECT, which is part of MODSIM. In addition, a description of the component must be added to the file TYPAR.DAT, which is read by HVACGEN. This section describes the format of the information required by HVACGEN. The TYPAR.DAT entry for the TYPE 9 linear valve model appears at the end of this section.

The first line must have an asterisk in the first column. This signals HVACGEN that a new TYPE description is about to begin. The remainder of the line is ignored.

The second line contains the TYPE number, followed by a description of the component in quotation marks.

The third line contains five integers, specifying the number of SAVED variables, differential equations, inputs, outputs, and parameters, respectively. In the case of the linear valve model, there are five saved variables, one differential equation, four inputs, two outputs, and four parameters.

Next comes a set of lines describing the outputs, followed by a set of lines describing the inputs. For the linear valve there are two lines for the outputs and four lines for the inputs. In general there is one line for each output and one line for each input. Each of these lines contains an integer followed by two character fields enclosed in quotation marks. The integer identifies the category of the input or output. Eight categories are available, as follows:

    1 - Pressure
    2 - Flow rate
    3 - Temperature
    4 - Control signal (or other fraction)
    5 - Rotation rate
    6 - Energy
    7 - Power
    8 - Humidity

The first character field provides a brief description of the input or output. The second character field gives recommended or required units for the variable.

After the output and input description lines, one or more optional lines may be used to modify the number of differential equations. Each such line consists of two integers, a relational operator in quotation marks, and a real number. The first integer is a parameter number. The second integer is a number to be subtracted from the number of differential equations if the specified condition is true. The truth or falsity of the condition is determined by applying the relational operator between the parameter and the real number. In the case of the linear valve model, the line

    2 1 'LT' 1.0

is equivalent to the FORTRAN statement

IF (PAR(2) .LT. 1.0) NDE = NDE - 1

where NDE is the number of differential equations.

        Any of the six FORTRAN relational operators may be used: 'EQ', 'NE',
'GT', 'GE', 'LT', or 'LE'. In addition, two or more conditions may be joined
with a logical AND, using as relational operators 'EQ.AND', 'NE.AND', and so
on. The second integer on any line using an AND operator is ignored. For
example, the two lines

        3 0 'GT.AND' 0.
        3 2 'LE' 1.

are equivalent to the FORTRAN statement

        IF ((PAR(3) .GT. 0.).AND.(PAR(3) .LE. 1.)) NDE = NDE - 2

        Lines modifying the number of differential equations are optional and may
be omitted entirely. There is no limit to the number of lines permitted, or
to the number of conditions which may be joined with logical ANDs. A number
sign (#) in column one tells HVACGEN not to expect any more conditions. The
number sign is required whether or not any conditions are present.

        Finally, one line for each parameter is required. These lines consist of
the parameter number and a single character field describing the parameter.
The set of parameter description lines may be followed by any number of blank
lines or comment lines, which will be ignored by HVACGEN.

Listing of TYPAR.DAT entry for the TYPE 9 linear valve model

```
*****************************************************************
9 'LINEAR VALVE WITH PNEUMATIC ACTUATOR'
5 1 4 2 4
4 'ACTUATOR RELATIVE POSITION' 'DIMENSIONLESS'
1 'INLET WATER PRESSURE' 'KPA'
1 'OUTLET WATER PRESSURE' 'KPA'
2 'WATER MASS FLOW RATE' 'KG/S'
4 'CONTROL VARIABLE FROM CONTROLLER' 'DIMENSIONLESS'
4 'ACTUATOR RELATIVE POSITION (SAME AS FIRST OUTPUT)' 'DIMENSIONLESS'
2 1 'LT' 1.0
#
1 'FLOW RESISTANCE [0.001/(KG-M)]'
2 'ACTUATOR TIME CONSTANT [SEC]'
3 'LEAKAGE PARAMETER [DIMENSIONLESS]'
4 'HYSTERESIS PARAMETER [DIMENSIONLESS]'
```

3. UTILITY FUNCTIONS AND SUBROUTINES

3.0 Introduction

This chapter describes a number of functions and subroutines which are available for use by TYPE components. Sections 3.1 and 3.2 describe two functions, DELAY and HYSTER, which are used by a number of component models to represent transport delays and hysteresis effects, respectively. Section 3.3 describes four subroutines which are used to determine heat exchanger fin efficiencies. Three of these subroutines, which compute modified Bessel functions and perform polynomial curve fitting, may also be useful for other applications. The remaining sections summarize functions and subroutines which calculate fluid properties. Most of these are not used by any of the existing component models, but are available for future use.

3.1 Transport Delays

General Description

Transport delays in conduits are modeled by the function DELAY, which takes the following arguments:

TIN   : the current value of the temperature (or other quantity) to be delayed
TAU   : the transport time, based on the current flow rate
A   : an array of six coefficients for a polynomial representing the temperature distribution in the conduit
TOLD   : the previous value of TIN (one time step ago)
DXOLD : the previous value of DX, the fractional distance the fluid moved in a time step
TIMOLD : the previous simulation time

The first two arguments, TIN and TAU, are supplied by the calling subroutine. The remaining arguments should be stored in the saved workspace by the calling subroutine. Thus the function might be used as in the following line:

    T = DELAY(T,TAU,SAVED(1),SAVED(7),SAVED(8),SAVED(9))

where SAVED(1) through SAVED(6) are used to store the array of 'A' coefficients.

Mathematical Description

Function DELAY uses a fifth-order time-dependent polynomial to model the temperature distribution in a pipe or duct:

$$T(x,t) = \sum_{i=0}^{5} a_i(t)x^i$$

where

a_i = time-dependent coefficients of the polynomial

x = normalized position in the pipe or duct $(0 \le x < 1)$

At time $t'=t+\Delta t$, the temperature distribution is shifted by a distance $\Delta x=\Delta t/\tau_x$, where τ_x is the transport time through the conduit:

$$T(x,t') = T(x-\Delta x,t) \; , \; \Delta x < x \leq 1$$

For positions between $x=0$ and $x=\Delta x$, linear interpolation is used to evaluate temperatures:

$$T(x,t') = T(0,t')+[T(0,t)-T(0,t')]\frac{x}{\Delta x} \; , \; 0 < x < \Delta x$$

At each time step, temperatures are evaluated at five points and a new set of coefficients for the polynomial are found by Gaussian elimination. It has been found empirically that for a stable solution, the evaluation points should be equally spaced along the duct or pipe with the following exceptions:

1. One of the points should be located at $x=\Delta x$.

2. If $\Delta x > 1/5$, there should be one or more points in the region $0<x<\Delta x$.

3. If $\Delta x > 1/2$, a linear temperature distribution is used.

4. If $\Delta x > 1$, a uniform temperature is used.

This method of representing transport delays is necessary because the simulation program selects its own variable time step between user-supplied minimum and maximum values. Since a minimum time step of a tenth of a second is not unreasonable, representing transport delays by simply storing a table of values could require an excessively long table.

3.2 Hysteresis Effects

Hysteresis effects are modeled by the function HYSTER. In the context of HVACSIM, HYSTER represents the relative position of a valve or damper, which may differ from the relative position of the actuator controlling the valve or damper position. Function HYSTER has the following arguments:

C   : a control signal between 0 and 1, representing the relative position of an actuator
CR : a variable related to the position of a valve or damper
CRO : the value of CR at the previous simulation time
TOLD: the previous simulation time
HYS : the hysteresis parameter

C can be thought of as the relative position of an actuator which moves a valve or damper, and HYS represents mechanical slack in the coupling between the actuator and the valve or damper. More specifically, HYS is the fraction of the actuator's range over which HYSTER remains constant when the actuator's direction of travel reverses. Like C, the value of HYSTER is always between 0 and 1. The relationship between C and HYSTER is illustrated in figure 3.2.1.

The values of C and HYS are supplied by the calling subroutine. The other three arguments should be stored in the saved workspace by the calling subroutine. Thus the function might be used as in the following line:

```
CH = HYSTER(C,SAVED(1),SAVED(2),SAVED(3),HYS)
```
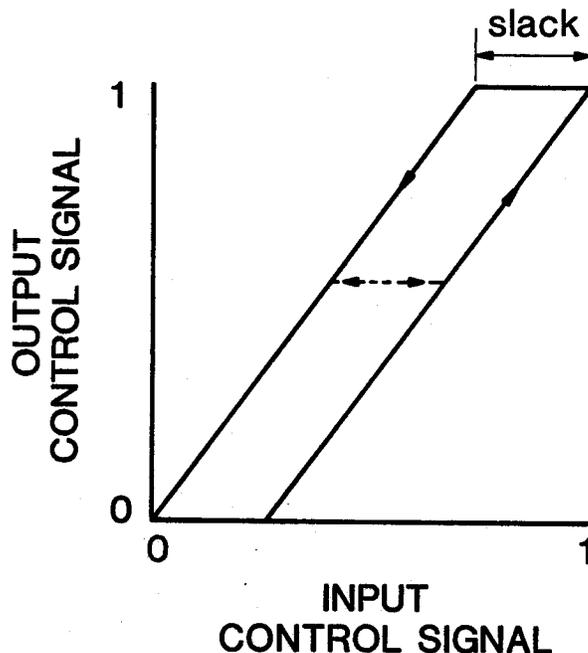
Figure 3.2.1. Hysteresis model.

3.3 Bessel Functions, Polynomial Curve Fitting, and Fin Efficiencies

The subroutines described in this section are taken, with minor modifications, from reference [8]. The first two subroutines, described briefly in sections 3.3.1 and 3.3.2, calculate the modified Bessel functions $I_n(x)$ and $K_n(x)$. Section 3.3.3 describes a subroutine used for polynomial curve fitting. The subroutine described in section 3.3.4 uses the first three subroutines to find heat exchanger fin efficiencies under any operating conditions, provided that condensation does not occur.

3.3.1 SUBROUTINE BESI(X,N,BI,IER)

This subroutine computes the modified Bessel function $I_n(x)$. The arguments of the subroutine are defined below.

X: the argument of the Bessel function
N: the order of the Bessel function (integer $N \geq 0$)
BI: the resultant value of the Bessel function
IER: the resultant error code:
        IER = 0   no error
        IER = 1 $N < 0$ (BI set to 1.)
        IER = 2 $X < 0$ (BI set to 1.)
        IER = 3 $BI < 10^{-38}$ (BI set to 0.)
        IER = 4 $X > 90.$ (BI set to 10^{38})

20

3.3.2   SUBROUTINE BESK(X,N,BK,IER)

This subroutine computes the modified Bessel function $K_n(x)$. The arguments of the subroutine are defined below.

X: the argument of the Bessel function
N: the order of the Bessel function (integer $N \geq 0$)
BK: the resultant value of the Bessel function
IER: the resultant error code:

IER = 0	no error	
IER = 1	$N < 0$	(BI set to 0.)
IER = 2	$X \leq 0.$	(BI set to 0.)
IER = 3	$X > 85.$	(BI set to 0.)
IER = 4	$BI > 10^{38}$	(BI set to 10^{38})

3.3.3   SUBROUTINE POLFIT(X,Y,N,TOL,LAST,COF)

This subroutine fits a polynomial to the ordered pairs of data points (x,y) and returns the coefficients of the polynomial. The arguments of the subroutine are defined below.

X: an array of up to 60 x-coordinates
Y: an array of up to 60 y-coordinates
N: the number of data points (X,Y) ($N \leq 60$)
TOL: an error tolerance
LAST: the maximum order of the fitted polynomial ($1 \leq LAST \leq 9$)
COF: an array of (LAST+1) coefficients to the polynomial

The subroutine begins by fitting a first order polynomial to the set of points. If the standard deviation of the correlation is greater than the error tolerance TOL and the order of the polynomial is less than LAST, the order is increased by one and a new set of coefficients is determined. This procedure is repeated until either the error tolerance is met or the maximum order specified by LAST is reached.

3.3.4   SUBROUTINE SUFED(ROH,COF)

The purpose of this subroutine is to obtain coefficients for a polynomial representing heat exchanger fin efficiencies (in the absence of condensation) as a function of the fin surface heat transfer coefficient and the fin geometry. The first argument of the subroutine, ROH, is the ratio of tube outside diameter to fin diameter:

$$ROH = D_{to}/D_f$$

For rectangular plate fins, D_f is the diameter of a circular fin having the same area per tube as the rectangular fin. The subroutine returns a set of five coefficients in the array COF. These coefficients can be used to determine heat exchanger dry fin efficiencies:

$$\eta_{dry} = C_1 + C_2 F + C_3 F^2 + C_4 F^3 + C_5 F^4$$

21

where

$$F = 0.5 \ (D_f - D_{t_o}) \ (2h/k\delta)^{0.5}$$

h = dry surface heat transfer coefficient

k = fin thermal conductivity

δ = fin thickness

The subroutine determines the fin efficiency by the method of Gardner [9] at sixty operating points, using the subroutines BESI and BESK, and calls subroutine POLFIT to determine the coefficients.

3.4 Refrigerant Properties

A number of functions are available for determining refrigerant properties. The equations are derived from real gas equations of state with coefficients for twelve different refrigerants [10]. The last argument of each function is an index, IR, coding for the refrigerant. The meaning of IR is as follows:

IR	Refrigerant	IR	Refrigerant
1	R-11	7	R-23
2	R-12	8	R-113
3	R-13	9	R-114
4	R-14	10	R-500
5	R-21	11	R-502
6	R-22	12	R-318

The functions require SI units on input, and return values in SI units. Internally, inputs are converted to English units, and results are converted back to SI units. The required units are as follows:

Variable	Symbol	Units
Pressure	P	kPa
Specific Volume	V	m^3/kg
Temperature	T	^{o}C
Entropy	S	$kJ/(kg^oC)$
Enthalpy	H	kJ/kg
Liquid Density	ρ	kg/m^3

Several of the functions are iterative and require a relative error tolerance, RTOL. With one exception, the tolerance is applied to a Fahrenheit temperature. An absolute tolerance of 0.001 is built into the functions. The single exception is the function which returns specific volume. In this case, the relative error tolerance RTOL is applied to the density (i.e., the inverse of the specific volume), and no absolute tolerance is used. All of the iterative functions are derived from the basic equations in reference [10], using the Newton-Rapheson method.

The list of available functions and their meanings is as follows:

Function Name	(Arguments)	Value Returned
PSAT	(T, IR)	P_{sat}
TSAT	(P, RTOL, IR)	T_{sat}
TVSAT	(V, RTOL, IR)	T_{sat}
PGAS	(V, T, IR)	P
VGAS	(P, T, RTOL, IR)	V
HGAS	(P, V, T, IR)	H
SGAS	(V, T, IR)	S
HPS	(P, S, RTOL, IR)	H
TPH	(P, H, RTOL, IR)	T
TVH	(V, H, RTOL, IR	T
DHLAT	(P, V, T, IR)	ΔH_{lat}
RHOLIQ	(T, IR)	ρ
CV	(V, T, IR)	C_v

In addition there is one subroutine which takes V, T, and IR, and determines C_p, C_v, the ratio C_p/C_v, and the velocity of sound in the refrigerant:

SUBROUTINE CPCV (V, T, IR, CP, CV, GAMMA, SONIC)

3.5 Steam and Liquid Water Properties

The functions described in this section compute water and steam properties. Some care is required in their use, since many of the functions are valid over a narrow range of conditions (e.g. only at saturation). Descriptions of the functions are divided into four sections: superheated steam, saturated steam, saturated liquid water, and water at atmospheric pressure. Three of the functions are listed in both of the last two sections. These functions are based on data for saturated water at temperatures greater than 100 °C, and water at atmospheric pressure for temperatures from 0 to 100. The latent heat of vaporization has been included in the saturated steam category.

In all of the water and steam property functions, temperatures are in Celsius, pressures are in kilopascals, specific volumes are in cubic meters per kilogram, and entropy is in kilojoules per kilogram per degree Celsius.

3.5.1 Superheated Steam

Function Name and Arguments		Value Returned	References
VS	(P, T)	V (m^3/kg)	eq. from [11]
HS	(P, T)	H (kJ/kg)	eq. from [11]
SS	(P, T)	S [kJ/(kg K)]	eq. from [11]
CPS	(T)	C_p [kJ/(kg K)]	eq. from [12]
TPSS	(P, S)	T (°C)	uses SS and CPS
VISSPH	(T)	μ [kg/(m s)]	data from [13]
STEAMK	(T)	K [W/(m K)]	data from [13]
CVS	(V, T)	C_v [kJ/(kg K)]	adapted from [14]

3.5.2 Saturated Steam

Function Name and Arguments	Value Returned	References
TSATS (P)	T (C)	eq. from [11]
PSATS (T)	P (kPa)	eq. from [11]
VSATS (P, T)	V (m^3/kg)	eq. from [11]
HSATS (T)	H (kJ/kg)	eq. from [11]
SSATS (T)	S [kJ/(kg K)]	eq. from [11]
VISSV (P)	μ [kg/(m s)]	data from [13]
HFG (T)	ΔH_{fg} (kJ/kg)	eq. from [11]

3.5.3 Saturated Liquid Water

Function Name and Arguments	Value Returned	References
VSATW (T)	V (m^3/kg)	eq. from [11]
HSATW (T)	H (kJ/kg)	eq. from [11]
SSATW (T)	S [kJ/(kg K)]	eq. from [11]
WMU (T)	μ [kg/(m s)]	data from [15] for T > 100 C.
WCP (T)	C_p [kJ/(kg K)]	data from [15] for T > 100 C.
WK (T)	K [kW/(m K)]	data from [15] for T > 100 C.

3.5.4 Liquid Water at 1 Atmosphere

Function Name and Arguments	Value Returned	References
WMU (T)	μ [kg/(m s)]	eq. from [16]
WCP (T)	C_p [kJ/(kg K)]	eq. from [16]
WK (T)	K [kW/(m K)]	data from [16]
WRHO (T)	ρ (kg/m^3)	eq. from [16]

3.6 Air Properties

Five functions and one subroutine are available for computing properties of air [11], which can be treated as an ideal gas. The subroutine argument list is as follows:

SUBROUTINE CPCVA(TCA, CPA, CVA, GAMMA, SONIC)

where TCA is the Celsius air temperature at which the remaining arguments are computed, CPA and CVA are specific heats at constant pressure and constant volume, respectively, in kJ/(kg K), GAMMA is the ratio CPA/CVA, and SONIC is the speed of sound in meters per second.

The five air property functions are as follows:

AKA (T) : Thermal conductivity [KW/(m K)] as a function of temperature (C).

VISCA (T): Dynamic viscosity [kg/(m s)] as a function of temper-ature (C).

HA (T)   : Enthalpy (kJ/kg) as a function of temperature (C).

PHIA (T) :   The entropy function (defined below) [kJ/(kg K)] as a function of
             temperature (C).

TPHIA (T):   Temperature (C) as a function of the entropy function [kJ/(kg K)].

The entropy function, ϕ, is defined by

$$\phi = \int_{T_0}^{T} \frac{C_{p(T)}}{T} \, dT$$

where T_0 is a reference temperature at which entropy is zero. Information on
the use of the entropy function can be found in reference [11] or in standard
thermodynamics textbooks.

4. COMPONENT MODELS

4.0 INTRODUCTION

This chapter is devoted to descriptions of component models developed for use with MODSIM. Wherever possible, experimental or theoretical verification of the models has been included with the descriptions. In Table 4.0.1, the available component models are listed in functional categories. Several of the components (TYPES 12, 13, 16, and 17) have been included in two categories.

The remainder of this section relates to the calculation of pressures and flow rates, which is common to many component models.

Pressure and Flow Calculations

Component models generally relate pressure losses to flow rates using equations of the following form:

$$\Delta P = Kw^2$$

where ΔP = pressure drop across the device
             K = flow resistance or pressure loss coefficient
             w = mass flow rate

On the basis of how pressure and flow calculations are handled, the component models can be divided into four categories:

1. No fluid flows through the component.
2. Fluid flows through the component but no pressure drop is calculated. Any flow resistance may be modeled by adding it to the flow resistance of an adjacent component.
3. Outlet pressures and inlet flow rates are used to calculate inlet pressures. If necessary, outlet flow rates are calculated by mass conservation.
4. Inlet and outlet pressures are used to calculate flow rates.

The connections among pressures and flow rates in a simulation define a set of equations which must be solved simultaneously. In setting up a simulation, the user must be careful to ensure that the set of equations is neither under-determined nor overdetermined, but has a unique solution. In general, one component from the fourth category is needed in each flow circuit, to determine a flow rate. The inlet pressure of this component will generally be a boundary condition (i.e. not an output of any other component), providing a fixed pressure basis for the determination of all other pressures in the circuit. For further discussion of pressure and flow rate calculations, refer to the example simulation chapter of reference [1]. In Table 4.0.2, the component models are organized into the four categories listed above.

Table 4.0.1 Component models arranged in functional categories

Fans and Pumps
TYPE   1: Fan or Pump

Conduits
TYPE   2: Conduit (Duct or Pipe)
TYPE   3: Inlet Conduit (Duct or Pipe)

Flow Splits and Merges
TYPE   4: Flow Merge
TYPE   6: Flow Split
TYPE 13: Three-way Valve with Actuator
TYPE 17: Mixing Dampers and Merge
TYPE 18: Plenum
TYPE 21: 'Grounded' Flow Split

Valves, Dampers, and Flow Restrictors
TYPE   5: Damper or Valve
TYPE   9: Linear Valve with Actuator
TYPE 13: Three-way Valve with Actuator
TYPE 17: Mixing Dampers and Merge
TYPE 23: Steam Nozzle
TYPE 24: Ideal Gas Nozzle

Sensors and Controllers
TYPE   7: Temperature Sensor
TYPE   8: Proportional-Integral Controller
TYPE 16: 'Sticky' Proportional Controller
TYPE 20: High or Low Limit Controller
TYPE 26: Control Signal Inverter

Heat Exchangers
TYPE 10: Hot Water to Air Heating Coil (Simple)
TYPE 11: Hot Water to Air Heating Coil (Detailed)
TYPE 12: Cooling or Dehumidifying Coil
TYPE 25: Steam to Air Heat Exchanger

Humidifiers and Dehumidifiers
TYPE 12: Cooling or Dehumidifying Coil
TYPE 14: Evaporative Humidifier
TYPE 22: Steam Spray Humidifier

Conditioned Zones
TYPE 15: Room

Superblock Coupling Components
TYPE 16: 'Sticky' Proportional Controller
TYPE 19: Flow Balance Control

Table 4.0.2 Component models arranged by pressure/flow calculation method

Category 1: No Fluid Flow

    TYPE 7: Temperature Sensor
    TYPE 8: Proportional-Integral Controller
    TYPE 16: 'Sticky' Proportional Controller
    TYPE 20: High or Low Limit Controller
    TYPE 26: Control Signal Inverter

Category 2: No Pressure Drop

    TYPE 12: Cooling or Dehumidifying Coil
    TYPE 14: Evaporative Humidifier
    TYPE 15: Room
    TYPE 18: Plenum
    TYPE 22: Steam Spray Humidifier
    TYPE 25: Steam to Air Heat Exchanger (Air Side)

Category 3: Inlet Pressure Calculated from Flow Rate and Outlet Pressure

    TYPE 1: Fan or Pump
    TYPE 2: Conduit (Duct or Pipe)
    TYPE 4: Flow Merge
    TYPE 5: Damper or Valve
    TYPE 6: Flow Split
    TYPE 9: Linear Valve with Actuator
    TYPE 10: Hot Water to Air Heating Coil (Simple)
    TYPE 11: Hot Water to Air Heating Coil (Detailed)
    TYPE 13: Three-way Valve with Actuator
    TYPE 19: Flow Balance Control

Category 4: Flow Rate Calculated from Inlet and Outlet Pressures

    TYPE 3: Inlet Conduit (Duct or Pipe)
    TYPE 17: Mixing Dampers and Merge
    TYPE 21: 'Grounded' Flow Split
    TYPE 23: Steam Nozzle
    TYPE 24: Ideal Gas Nozzle
    TYPE 25: Steam to Air Heat Exchanger (Steam Side)

Flow Resistance Coefficients

    As stated above, component models generally relate pressure losses to
flow rates using equations of the following form:

$$\Delta P = Kw^2$$

When ΔP is in kPa and w is in kg/s, K has dimensions of $g^{-1}m^{-1}$. If ΔP is in
Pascals, K is in $kg^{-1}m^{-1}$.

    A dimensionless loss coefficient, K', commonly reported in the litera-
ture, is defined as follows:

$$K' = \frac{\text{head loss}}{\text{velocity head}} = \frac{2\Delta P}{\rho v^2}$$

where ρ is the density of the fluid and v is its velocity. For K' to be dimensionless, ΔP must be in pascals when ρ is in kg/m^3 and v is in m/s. The relationship between the dimensionless parameter K' and the dimensioned parameter K is

$$K = \frac{0.001 \; K'}{2\rho A^2}$$

where A is the cross-sectional flow area and the factor of 0.001 converts K for use with pressures in kilopascals.

K may also be expressed as a function of the dimensionless Darcy friction factor, f:

$$K = \frac{8fL}{\rho \pi^2 d^5}$$

where d is the hydraulic diameter of the pipe or duct and L is its length.

Valves and Dampers

For valves and dampers, K is a function of a control signal defining the position of the valve or damper, and of parameters defining the characteristics of the valve or damper. Two parameters common to the valve and damper models are λ, a fractional leakage rate at unit ΔP when the valve or damper is closed, and K_o, a flow resistance when the valve or damper is open. The flow-handling capacity of a control valve is commonly represented in the literature by a flow coefficient or valve constant, C_v, defined as follows:

$$C_v = (q)(\Delta P)^{-0.5}$$

where q is the volumetric flow rate in gallons per minute and ΔP is the pressure drop in pounds per square inch when the valve is fully open. When ΔP is in kPa and w is in kg/s,

$$K_o = 1.732 \times 10^9 \; (\rho C_v)^{-2}$$

A valve is essentially a variable fluid resistance. The manner in which the flow rate varies with valve stem position is known as the valve characteristic, or inherent characteristic, since it is always measured in practice with a constant pressure drop across the valve. For a linear valve, the flow rate is directly proportional to the valve stem position when the pressure drop across the valve is constant.

It is important to distinguish between the inherent characteristic of a valve and the installed characteristic. The latter depends on both the inherent characteristic and the authority of the valve, a:

$$a = \frac{\Delta P_v}{\Delta P_v + \Delta P_s} = \frac{K_o}{K_o + K_s}$$

29

where ΔP_v = pressure drop across the open valve,

ΔP_s = pressure drop across the remainder of the circuit,

K_o = flow resistance of the open valve, and

K_s = sum of flow resistances around the remainder of the circuit.

For three-way valves, ΔP_s and K_s refer only to that portion of the circuit where the flow rate is being controlled by the inlet port in question, and the two inlet ports may have different authorities.

The installed characteristic of a valve is strictly identical to the inherent characteristic only when the authority is one, that is, when the valve represents the only pressure drop in the system. A valve with low authority has little effect on the flow rate over most of its operating range.

The leakage flow rate through a closed valve also depends on the valve's authority, as well as on the leakage parameter, λ. For the valve and damper models described in this chapter, the normalized or fractional leakage rate is given by

$$\frac{w_{min}}{w_{max}} = \lambda[a + \lambda^2(1 - a)]^{-0.5}$$

Dampers also have inherent characteristics, installed characteristics, and authority. The effects of valve and damper authority are accurately represented by the valve and damper models described in this chapter.

4.1 TYPE 1: FAN OR PUMP

General Description

The pump or fan model calculates a pressure rise and an efficiency as functions of a mass flow rate, using dimensionless performance curves. The performance curves are represented by polynomials, with empirically determined coefficients which must be supplied by the user. The efficiency is used to compute a temperature rise across the fan or pump, as well as power consumption. The last parameter, MODE, determines whether air or water properties (density and specific heat) are to be used.

Nomenclature

C_f — dimensionless flow coefficient
C_h — dimensionless pressure head coefficient
C_p — specific heat of fluid (kJ/kg-C)
D — diameter of blades or impeller (m)
E — power consumption (kW)
N — rotational speed (rps)
P_i — inlet pressure (kPa)
P_o — outlet pressure (kPa)
T_i — inlet temperature (C)
T_o — outlet temperature (C)
w — mass flow rate (kg/s)
η — efficiency
ρ — fluid density (kg/m³)

Mathematical Description

The dimensionless flow coefficient and the dimensionless pressure head coefficient are defined as follows:

$$C_f = \frac{w}{\rho N D^3}$$

$$C_h = \frac{1000 \Delta P}{\rho N^2 D^2}$$

where ΔP is the pressure rise, $P_o - P_i$, in kilopascals. The factor of 1000 converts the pressure difference from pascals to kilopascals for dimensional consistency. For this reason, the units given in the nomenclature section must be used.

The model determines the head coefficient, C_h, and the efficiency, η, as functions of the flow coefficient, C_f:

$$C_h = a_0 + a_1 C_f + a_2 C_f^2 + a_3 C_f^3 + a_4 C_f^4$$

$$\eta = e_0 + e_1 C_f + e_2 C_f^2 + e_3 C_f^3 + e_4 C_f^4$$

The 'a' and 'e' coefficients must be supplied as parameters by the user. The inlet pressure is then obtained from the definition of C_h:

$$P_i = P_o - 0.001C_h\rho N^2 D^2$$

The efficiency is used to compute power consumption, and a temperature rise is calculated on the assumption that all inefficiencies heat the fluid:

$$E = \frac{w\Delta P}{\eta\rho}$$

$$T_o = T_i + \frac{(P_o - P_i)}{\rho C_p} \left(\frac{1}{\eta} - 1\right)$$

TYPE 1 Component Configuration

Inputs		Description
1	w	— fluid mass flow rate
2	P_o	— outlet pressure
3	N	— rotational speed
4	T_i	— inlet temperature

Outputs		Description
1	P_i	— inlet pressure
2	T_o	— outlet temperature
3	E	— power consumption

Parameters		Description
1-5	a_i	— coefficients for head vs. flow curve
6-10	e_i	— coefficients for efficiency curve
11	D	— diameter
12	Mode	— air = 1 , water = 2

4.2 TYPE 2: CONDUIT (DUCT OR PIPE)

General Description

The conduit model is designed to account for three effects: thermal losses to ambient conditions, transport delays, and dynamics due to thermal capacitance. The model has four modes, determined by the last parameter. If the absolute value of MODE is 1, air properties are used and the model represents a duct. If the absolute value of MODE is 2, water properties are used and the model represents a pipe. The sign of MODE determines the method used to calculate thermal capacitance effects. The merits of the two methods are discussed below.

Nomenclature

A — surface area of conduit
C_m — thermal capacitance (mass * specific heat) of conduit
C_p — specific heat of fluid
h — heat transfer coefficient
K — flow resistance coefficient
P — pressure
T — temperature
t — time
U — overall heat transfer coefficient
V — volume of conduit
w — mass flow rate of fluid
ρ — density of fluid
τ — time constant
τ_c — capacitance term of time constant
τ_x — transport time

Subscripts

amb   — ambient
i   — inlet or inside
o   — outlet or outside
ss   — steady state

Mathematical Description

The pressure at the inlet of the conduit is calculated from the outlet pressure and the mass flow rate, using a flow resistance coefficient, K, which is assumed constant:

$$P_i = P_o + \text{sign}(w)Kw^2 \qquad (4.2.1)$$

The steady state outlet temperature is given by

$$T_{ss} = T_{amb} + (T_i - T_{amb})\exp(-\gamma) \qquad (4.2.2)$$

where $\qquad \gamma = \dfrac{UA}{wC_p}$

If MODE is positive, the temperature dynamics are represented by the following equation:

$$\frac{dT_o}{dt} = \frac{T_{ss} - T_o}{\tau} \qquad (4.2.3)$$

where

$$\tau = \left[\frac{h_i}{h_i + h_o} \right] \frac{C_m}{wC_p}$$

A derivation of this equation can be found in reference [17]. This equation is solved analytically within the subroutine, bypassing the MODSIM non-linear differential equation solver:

$$T_o^+ = T_{ss} + (T_o^- - T_{ss})\exp(-\frac{\Delta t}{\tau}) \qquad (4.2.4)$$

This result is then sent to the DELAY subroutine, which returns the current outlet temperature:

$$T_o = \text{DELAY}(T_o^+, \tau_x) \qquad (4.2.5)$$

where τ_x is the transport time through the conduit:

$$\tau_x = \frac{\rho V}{w}$$

If MODE is negative, the temperature dynamics are represented by the following equation:

$$\frac{dT_o}{dt} = \frac{-\exp(-\alpha)}{\tau_c}\{ T_o - \text{DELAY}[T_{ss} + \tau_c \exp(-\gamma-\alpha)\frac{dT_i}{dt}] \} \qquad (4.2.6)$$

where

$$\tau_c = \frac{C_m}{h_i A}$$

$$\alpha = \frac{\gamma h_i}{2h_o}$$

and γ is defined above. This equation is derived by inverse transformation of an approximate transfer function given in reference [18], and is solved outside the subroutine by the MODSIM non-linear differential equation solver.

Model Verification

The approximate transfer function from which equation 4.2.6 is derived has been compared with a more exact transfer function [18]. The error of the approximation is a function of α and $\tau_c\omega$, where ω is an oscillation frequency, but the maximum error is a function of α alone. When α is less than or equal to one, the error never exceeds 10% in magnitude and 8° in phase angle. (As discussed below, some additional error will be introduced by the transport delay subroutine.) For further details see reference [18]. Since α is directly proportional to the length of the conduit, accuracy can be improved by dividing a long conduit into two or more shorter conduits.

34

For a unit step increase in the inlet temperature, equation 4.2.6 yields an outlet temperature step increase of magnitude $\exp(-h_iA/wC_p)$ at the end of the delay time, followed by an exponential rise with a time constant of $\tau_c\exp(-a)$. The time constant of the simpler model, τ, is essentially always smaller than this, which helps to compensate for the lack of an output step change in the simpler model. A comparison between the two models is provided by figure 4.2.1. The parameter values in this comparison were calculated for a 0.508 meter (20 inch) square duct, 65.8 meters (216 feet) in length, carrying air at 1.7 cubic meters per second (3600 cubic feet per minute). The transport delay in this example is ten seconds.
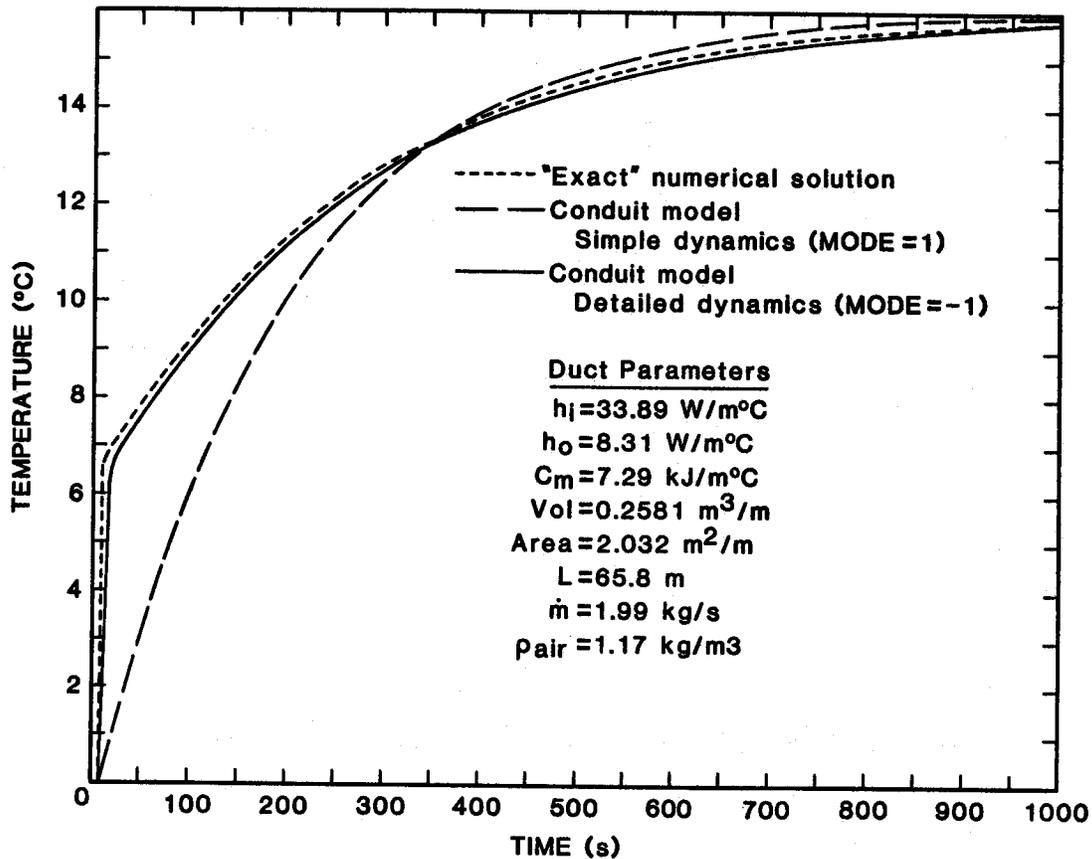
Figure 4.2.1   Conduit model response to a step change in the fluid inlet temperature at time zero.

35

Limitations of the Model

Although a step change in the inlet temperature has been applied here for purposes of illustration, use of the detailed mode with changes in the inlet conditions which are rapid relative to the transport time is not generally recommended. This is due to the intrinsic limitations of the transport delay subroutine, which is required to pass the derivative of the inlet temperature in the detailed mode. A step change in the inlet temperature produces a spike in the term sent to DELAY, and a ramp input produces a step change in DELAY. The detailed mode is generally unreliable for step change inputs, and its accuracy diminishes when the flow rate is such that the transport time exceeds the duration of a ramp input or the period of a sinusoidal input. The characteristics of DELAY also limit the accuracy of the simple mode when the transport time exceeds the time interval between two step changes or the period of a sinusoidal input.

TYPE 2 Component Configuration

Inputs		Description
1	w	- fluid mass flow rate
2	P_o	- outlet pressure
3	T_i	- inlet temperature
4	T_{amb}	- ambient temperature
5	T_o	- outlet temperature

Outputs		Description
1	T_o	- outlet temperature
2	P_i	- inlet pressure

Parameters		Description
1	h_iA	- inside surface heat transfer coefficient * area
2	h_oA	- outside surface heat transfer coefficient * area
3	C_m	- thermal capacitance of conduit material
4	V	- volume of conduit
5	K	- flow resistance coefficient
6	H	- height of outlet above inlet
7	MODE -	-2 => water, detailed dynamics (one D.E.)
		-1 => air, detailed dynamics (one D.E.)
		1 => air, simple dynamics (no D.E.)
		2 => water, simple dynamics (no D.E.)

4.3 TYPE 3: INLET CONDUIT (DUCT OR PIPE)

General Description

In the TYPE 2 Conduit model, the inlet pressure is calculated from the outlet pressure and the mass flow rate. In the TYPE 3 Inlet Conduit model, the mass flow rate is calculated from the inlet and outlet pressures. In all other respects the two models are identical.

Nomenclature

K — flow resistance coefficient

ΔP — inlet pressure minus outlet pressure

w — mass flow rate

Mathematical Description

Equation 4.2.1 in the TYPE 2 Conduit model is replaced by the following equation:

$$w = \text{sign}(\Delta P) \ (\Delta P/K)^{0.5}$$

Temperatures are calculated as described for the TYPE 2 Conduit model.

TYPE 3 Component Configuration

Inputs		Description
1	P_i	— inlet pressure
2	P_o	— outlet pressure
3	T_i	— inlet temperature
4	T_{amb}	— ambient temperature
5	T_o	— outlet temperature

Outputs		Description
1	T_o	— outlet temperature
2	w	— mass flow rate

Parameters	Description
1	h_iA – inside surface heat transfer coefficient * area
2	h_oA – outside surface heat transfer coefficient * area
3	C_m – thermal capacitance of conduit material
4	V – volume of conduit
5	K – flow resistance coefficient
6	H – height of outlet above inlet
7	MODE –

$-2 \Rightarrow$ water, detailed dynamics (one D.E.)
$-1 \Rightarrow$ air, detailed dynamics (one D.E.)
$1 \Rightarrow$ air, simple dynamics (no D.E.)
$2 \Rightarrow$ water, simple dynamics (no D.E.)

4.4 TYPE 4: FLOW MERGE

General Description

This component models the merging of two flow streams. It calculates the mass flow rate and temperature of the single outlet flow stream, and the pressures at the two inlets. The model assumes that the flow resistance parameter is the same for all three branches.

Nomenclature

K   — flow resistance coefficient
P   — pressure
T   — temperature
w   — mass flow rate

Subscripts

1   — first inlet
2   — second inlet
3   — outlet

Mathematical Description

The outlet mass flow rate is simply the sum of the inlet flow rates:

$$w_3 = w_1 + w_2$$

The outlet temperature is a weighted average of the inlet temperatures:

$$T_3 = \frac{w_1 T_1 + w_2 T_2}{w_3}$$

Inlet pressures are calculated from the outlet pressure, the flow rates, and the flow resistance coefficient:

$$P_1 = P_3 + \frac{K}{2}[sign(w_1)w_1^2 + sign(w_3)w_3^2]$$

$$P_2 = P_3 + \frac{K}{2}[sign(w_2)w_2^2 + sign(w_3)w_3^2]$$

TYPE 4 Component Configuration

Inputs		Description
1	w_1	— flow rate at first inlet
2	w_2	— flow rate at second inlet
3	P_3	— pressure at outlet
4	T_1	— temperature at first inlet
5	T_2	— temperature at second inlet

Outputs		Description
1	w_3	– flow rate at outlet
2	P_1	– pressure at first inlet
3	P_2	– pressure at second inlet
4	T_3	– temperature at outlet

Parameters		Description
1	K	– flow resistance coefficient

4.5 TYPE 5: DAMPER OR VALVE

General Description

A damper or valve is essentially a variable fluid resistance. The manner in which the flow rate varies with damper or valve position is known as the damper or valve characteristic, or inherent characteristic. The inherent characteristic is always measured in practice with a constant pressure drop across the damper or valve. For a linear valve, for example, the flow rate is directly proportional to the valve stem position when the pressure drop across the valve is constant.

This model represents dampers or valves having inherent characteristics which are linear, exponential, or intermediate between linear and exponential. The characteristic is determined by the value of the third parameter, W_f. When W_f is zero, the model represents an exponential valve or damper. When W_f is one, a linear valve or damper is modeled. Intermediate values of W_f produce intermediate characteristics.

The model computes the inlet pressure as a function of mass flow rate and relative damper or valve position. A non-zero leakage parameter is required to prevent infinite flow resistance when the damper or valve is fully closed. Parameter values for modeling opposed leaf dampers and parallel leaf dampers are given in the model validation section below.

It is important to distinguish between the inherent characteristic of a damper or valve and the installed characteristic. The latter depends on both the inherent characteristic and the authority of the damper or valve, as discussed in Section 4.0. The installed characteristic of an inherently linear valve is strictly linear only when the authority is one, that is, when the valve represents the only pressure drop in the system. A valve with low authority has little effect on the flow rate over most of its operating range. When used in a system with other components, the TYPE 5 component accurately models the effects of authority.

Nomenclature

        C   - relative position
        K   - flow resistance coefficient [$\sqrt{1000}$/(kg m)]
        M   - mode: M = 0 => damper closed when C = 0.
                       M = 1 => damper closed when C = 1.
        P   - pressure (kPa)
        w   - mass flow rate (kg/s)
        W_f - weighting factor for linear term of flow resistance coefficient
        λ - leakage parameter: fractional leakage when ΔP = 1 kPa.

Subscripts

        o   - full open
        1   - inlet
        2   - outlet

Mathematical Description

If $M \neq 0$ then $C = 1. - C$

$$K = \frac{W_f K_o}{[(1 - \lambda)C + \lambda]^2} + (1 - W_f)K_o\lambda^{(2C - 2)}$$

$$P_1 = P_2 + \text{sign}(w)Kw^2$$

K is the flow resistance as a function of C, the relative position ($0 \leq C \leq 1$). The inherent characteristic of the damper or valve model is determined by the parameter W_f. When W_f is zero, the model represents an exponential valve or damper. When W_f is one, a linear valve or damper is modeled. Intermediate values of W_f produce intermediate characteristics.

Model Validation

Reference [19] gives curves of dimensionless flow resistance versus damper position for parallel leaf dampers and for opposed leaf dampers, which are the two types of dampers most commonly used in HVAC control applications. The above equation for K was converted to dimensionless form as described in section 4.0, and a nonlinear regression program was used to find values for the parameters K_o, λ, and W_f which minimized the difference between the above equation for K and the published curves. The following values were obtained:

	Opposed	Parallel
$2\rho A^2 K_o$	.52274	.54205
λ	.01091	.012214
W_f	.61785	.8689

Reference [20] contains curves of flow versus position at several authorities for parallel leaf and opposed leaf dampers. As an independent check on the model, a simulation was run with the inlet conduit model (TYPE 3) joined to the TYPE 5 component with the parameters given above, and the authority was varied by varying the conduit resistance. Figures 4.5.1 and 4.5.2 compare the simulation results with published curves of the same damper authorities for opposed and parallel leaf dampers respectively [20]. For parallel leaf dampers, agreement between the model and the published curves is excellent. For opposed leaf dampers the results are less dramatic, but better agreement might be obtained by decreasing the leakage parameter used in the simulations.
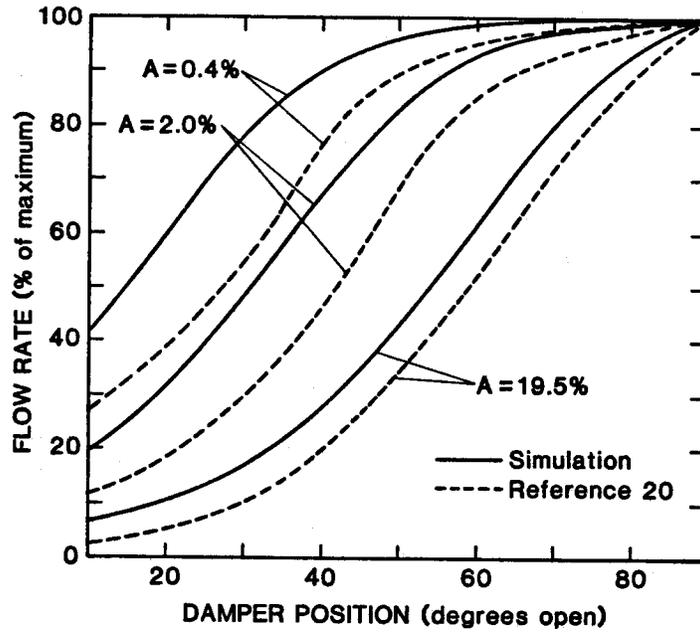
Figure 4.5.1   Flow versus position for an opposed leaf damper, as a function of damper authority, A.
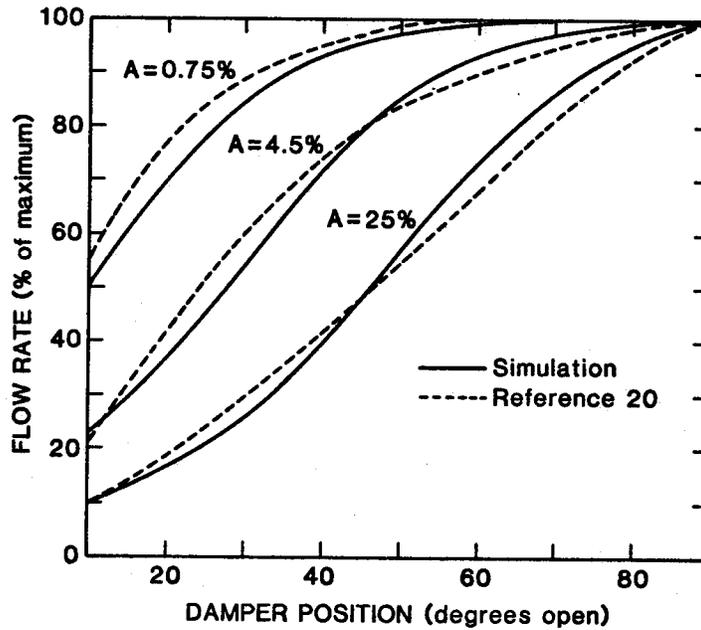
Figure 4.5.2   Flow versus position for a parallel leaf damper, as a function of damper authority, A.

43

TYPE 5 Component Configuration

Inputs	Description
1 w	– mass flow rate [kg/s]
2 P_2	– outlet pressure [KPa]
3 C	– relative position of damper or valve [–]

Outputs	Description
1 P_1	– inlet pressure [KPa]

Parameters	Description
1 K_o	– full open resistance [$1000./(kg-m)$]
2 λ	– leakage parameter [–]
3 W_f	– weighting factor for linear term of flow resistance [–]
4 M	– mode: M = 0 => damper closed when C = 0. M = 1 => damper closed when C = 1.

4.6 TYPE 6: FLOW SPLIT

General Description

This component models the division of a flow stream into two flow streams. It calculates the mass flow rates at the two outlets, and the pressure at the single inlet. The model assumes that the flow resistance parameter is the same for all three branches. Care has been taken to ensure that correct results are obtained even when the flow direction of one or more flow streams is reversed.

Nomenclature

K - flow resistance coefficient
P - pressure
w - mass flow rate

Subscripts

0   - center of component
1   - inlet
2   - first outlet
3   - second outlet

Mathematical Description

The model is derived from the following set of equations:

$$P_1 - P_0 = 0.5 \; K \; sign(w_1) \; w_1^2$$

$$P_0 - P_2 = 0.5 \; K \; sign(w_2) \; w_2^2$$

$$P_0 - P_3 = 0.5 \; K \; sign(w_3) \; w_3^2$$

$$w_1 = w_2 + w_3$$

where P_0 is the pressure at the center of the component and 0.5 K is the flow resistance between the center and an inlet or outlet. The set of equations is solved for w_2 and w_3, and the solution is expressed in the following form:

$$w_2 = \frac{w_1}{2} + [sign(w_1)] \frac{\Delta P}{K w_1} \qquad \text{if } S \geq 0.$$

$$w_2 = \frac{w_1}{2} + [sign(\Delta P)] \left[\frac{abs(\Delta P)}{K} - \frac{w_1^2}{4} \right]^{0.5} \qquad \text{if } S < 0.$$

where $\Delta P = P_3 - P_2$

and $S = w_1^2 - \dfrac{2 \; abs(\Delta P)}{K}$

$$w_3 = w_1 - w_2$$

45

When S is positive, all three flow rates have the same sign, and w_1 represents the single inlet (positive) or outlet (negative) flow rate of a split or merge. When S is negative, w_2 and w_3 do not have the same sign, and w_1 represents one of two inlets (w_1 positive) or outlets (w_1 negative).

The pressure at the nominal inlet, P_1, is then calculated:

$$P_1 = P_3 + \frac{K}{2}[\text{sign}(w_1)w_1^2 + \text{sign}(w_3)w_3^2]$$

TYPE 6 Component Configuration

Inputs		Description
1	w_1	– flow rate at inlet
2	P_2	– pressure at first outlet
3	P_3	– pressure at second outlet

Outputs		Description
1	w_2	– flow rate at first outlet
2	w_3	– flow rate at second outlet
3	P_1	– pressure at inlet

Parameters		Description
1	K	– flow resistance coefficient

46

4.7 TYPE 7: TEMPERATURE SENSOR

General Description

Temperature sensors are modeled by a simple first-order differential equation with a single time constant. The input to the component is modified using a gain and an offset supplied as parameters, allowing the output to be treated as a voltage level, a control signal, or a temperature in either Fahrenheit or Celsius. If a Celsius temperature output is desired, an offset of zero and a gain of one are used. If a control variable between zero and one is required for use as a controller input, the offset is the minimum allowable temperature and the gain is the maximum allowable temperature minus the minimum temperature.

Nomenclature

C_i — temperature input modified by gain and offset
C_o — output signal (e.g. sensed temperature)
T_i — temperature input
T_g — gain
T_o — offset
Δt — time step
τ — time constant

Mathematical Description

$$C_i = \frac{T_i - T_o}{T_g}$$

$$\frac{dC_o}{dt} = \frac{C_i - C_o}{\tau}$$

If the time constant, τ, is greater than or equal to one second, the differential equation is solved outside the subroutine by the MODSIM differential equation solver. If τ is less than one second, the differential equation is solved within the subroutine:

IF $(\tau/\Delta t) < 0.05$ OR $|C_i - C_o^-| < 10^{-10}$ THEN

$$C_o = C_i$$

ELSE

$$C_o = C_i - (C_i - C_o^-)\exp(-\Delta t/\tau)$$

where C_o^- is the value of C_o one time step ago.

TYPE 7 Component Configuration

Inputs		Description
1	T_i	– temperature input
2	C_o	– temperature sensor output

Outputs		Description
1	C_o	– temperature sensor output

Parameters		Description
1	τ	– temperature sensor time constant
2	T_o	– offset
3	T_g	– gain

4.8 TYPE 8: PROPORTIONAL-INTEGRAL CONTROLLER

General Description

This subroutine models an analog proportional-integral controller. Proportional control alone can be modeled by setting the integral gain to zero. A time constant is used to model the response time of the controller itself. The feedback signal, generally representing a temperature, must be a control variable, which can be obtained by passing a temperature through the TYPE 7 temperature sensor model.

Nomenclature

C — controller output ($0 \le C \le 1$)
C_f — feedback signal (controlled variable)
C_{set} — set point
C_{ss} — steady state controller output
E — error signal
I — integral portion of controller output
K_i — integral gain
K_p — proportional gain
τ — time constant associated with controller response

Mathematical Description

The error signal is defined by

$$E = C_{set} - C_f$$

The integral portion of the output signal is given by

$$\frac{dI}{dt} = K_i E$$

The output control signal is then determined:

$$\frac{dC}{dt} = \frac{C_{ss} - C}{\tau}$$

where

$$C_{ss} = K_p E + I$$

This differential equation is solved by MODSIM unless the time constant, τ, is less than one second, in which case it is solved within the subroutine:

IF $(\tau/\Delta t) < 0.05$ OR $|C_{ss} - C^-| < 10^{-10}$ THEN

$$C = C_{ss}$$

ELSE

$$C = C_{ss} - (C_{ss} - C^-)\exp(-\Delta t/\tau)$$

where C^- is the value of C one time step ago.

49

TYPE 8 Component Configuration

Inputs		Description
1	C_f	– controlled variable
2	C_{set}	– set point for controlled variable
3	I	– integral portion of control signal (from first output)
4	C	– output control signal (from second output)

Outputs		Description
1	I	– integral portion of control signal ($0 \leq I \leq 1$)
2	C	– output control signal ($0 \leq C \leq 1$)

Parameters		Description
1	K_p	– proportional gain
2	K_i	– integral gain
3	τ	– controller time constant

4.9 TYPE 9: LINEAR VALVE WITH PNEUMATIC ACTUATOR

General Description

A valve is essentially a variable fluid resistance. The manner in which the flow rate varies with valve stem position is known as the valve characteristic, or inherent characteristic, since it is always measured in practice with a constant pressure drop across the valve. For a linear valve, the flow rate is directly proportional to the valve stem position when the pressure drop across the valve is constant.

It is important to distinguish between the inherent characteristic of a valve and the installed characteristic. The latter depends on both the inherent characteristic and the authority of the valve, as discussed in Section 4.0. The installed characteristic of an inherently linear valve is strictly linear only when the authority is one, that is, when the valve represents the only pressure drop in the system. A valve with low authority has little effect on the flow rate over most of its operating range. When used in a system with other component models, the TYPE 9 valve accurately models the effects of valve authority. Figure 4.9.1 illustrates the effect of valve authority on the installed characteristic of a linear valve. This figure was obtained using the TYPE 9 linear valve model in conjunction with a TYPE 3 pipe model.
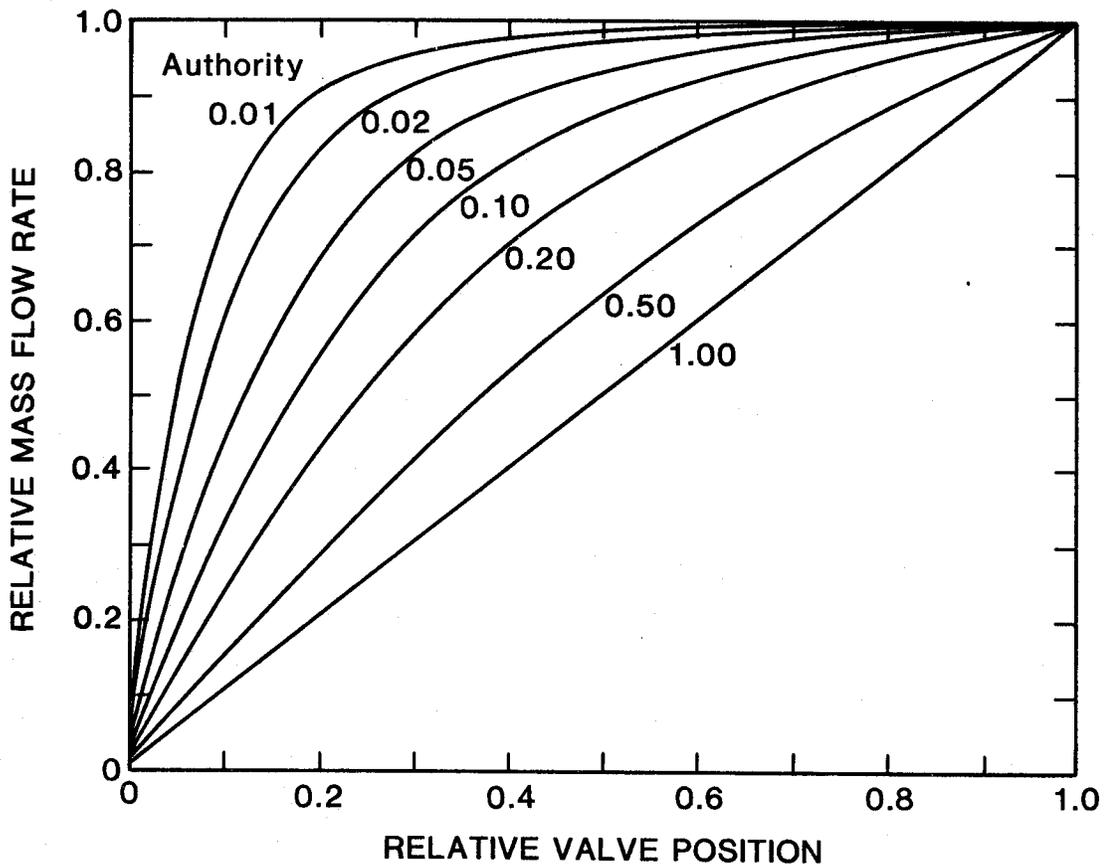
Figure 4.9.1   Linear valve characteristics as a function of authority, A.

The valve model includes a pneumatic actuator model. Actuator hysteresis is represented using the utility subroutine HYSTER, described in section 3.2. The dynamic response of the actuator is modeled by a first order differential equation. The actuator can be effectively removed by setting the time constant and the hysteresis parameter to zero.

When the flow rate through the valve goes to zero, the pressure drop across the valve becomes indeterminate. To prevent this problem, a non-zero leakage parameter is required by the model.

Nomenclature

C – requested relative valve position ($0 \leq C \leq 1$)
C_a – relative actuator position ($0 \leq C_a \leq 1$)
C_h – relative valve position ($0 \leq C_h \leq 1$)
hys – fraction of actuator's range over which C_h remains constant
       when actuator's direction of travel reverses
K – flow resistance parameter when valve is open ($C_h = 1$)
w – mass flow rate
λ – leakage parameter: fractional leakage when $\Delta P = 1$.
τ – actuator time constant

Mathematical Description

The relationship between the input control signal, C, and the actuator position, C_a, is defined by:

$$\frac{dC_a}{dt} = \frac{C - C_a}{\tau}$$

This differential equation is solved by MODSIM unless the time constant, τ, is less than one second, in which case the following solution is used:

IF $(\tau/\Delta t) < 0.05$ OR $|C_{ss} - C^-| < 10^{-10}$ THEN

$$C_a = C$$

ELSE

$$C_a = C - (C - C_a^-)\exp(-\Delta t/\tau)$$

where C_a^- is the value of C_a one time step ago.

The actuator position differs from the valve position, C_h, due to hysteresis effects, which are determined by the utility function HYSTER described in section 3.2.

$$C_h = \text{HYSTER}(C_a, \text{hys})$$

The inlet pressure is given by

$$P_i = P_o + \text{sign}(w)Kw^2[(1-\lambda)C_h + \lambda]^{-2}$$

52

TYPE 9 Component Configuration

Inputs		Description
1	P_o	— pressure at outlet
2	w	— mass flow rate
3	C	— input control signal ($0 \leq C \leq 1$)
4	C_a	— actuator relative position

Outputs		Description
1	C_a	— actuator relative position
2	P_i	— pressure at inlet
3	C	— valve stem relative position

Parameters		Description
1	K	— flow resistance when valve is open
2	τ	— actuator time constant
3	λ	— leakage parameter
4	hys	— hysteresis parameter

4.10   TYPE 10:   HOT WATER TO AIR HEATING COIL (SIMPLE)

General Description

The TYPE 10 subroutine is the simplest of three water-to-air heat exchanger models available at present in the TYPES library. A constant overall heat transfer coefficient (UA) is assumed. Steady state air and water outlet temperatures are calculated using an approximate effectiveness relationship for cross-flow heat exchangers. Dynamic outlet temperatures are calculated using first-order differential equations, in which the time constant is a function of both the coil flush time and the thermal capacitance of the coil. However, the outlet temperature is not delayed by the flush time of the coil.

Nomenclature

C_a – capacitance rate (specific heat times mass flow rate) of air
C_{min} – minimum of C_a and C_w
C_w – capacitance rate (specific heat times mass flow rate) of water
K_a – flow resistance parameter on air side of coil
K_w – flow resistance parameter on water side of coil
NTU   – number of transfer units
R   – ratio of minimum to maximum capacitance rate
UA   – overall heat transfer coefficient
w_a – mass flow rate of air
w_w – mass flow rate of water
ε – effectiveness
τ – coil time constant
τ_c – capacitive term of coil time constant
τ_x – coil flush time

Subscripts

a   – air
i   – inlet
o   – outlet
ss   – steady state
w   – water

Mathematical Description

The heating coil model uses an approximate equation for effectiveness as a function of NTU (number of transfer units) for a cross-flow heat exchanger with both fluids unmixed [21] to determine steady state air and water outlet temperatures:

$$\varepsilon = 1 - \exp\{[\exp(-Rn\{NTU\})-1]/Rn\}$$

where

$$n = (NTU)^{-0.22}$$

and NTU is defined by

$$NTU = \frac{UA}{C_{min}}$$

The steady state air and water outlet temperatures are found using the definition of ε:

$$T_{aoss} = T_{ai} + (T_{wi} - T_{ai})\varepsilon C_{min}/C_a$$

$$T_{woss} = T_{wi} - (T_{aoss} - T_{ai})C_a/C_w$$

A coil time constant is also calculated by the model, using the following equation:

$$\tau = (\tau_c^{-1} + \tau_x^{-1})^{-1}$$

where τ_c is a capacitive term supplied as a parameter and τ_x is the coil flush time, which is determined from the coil volume:

$$\tau_x = \frac{\rho Vol}{w_w}$$

The equation for the time constant was chosen to give the correct qualitative behavior of τ as a function of water flow rate (see reference [22], figure 7). Air and water dynamic outlet temperatures are both calculated using this time constant:

$$\frac{dT_{wo}}{dt} = \frac{T_{woss} - T_{wo}}{\tau}$$

$$\frac{dT_{ao}}{dt} = \frac{T_{aoss} - T_{ao}}{\tau}$$

If both τ_c and the coil volume are zero, the differential equations are eliminated from the model and the steady state temperatures are used.

Finally, inlet pressures are calculated from the outlet pressures and flow rates:

$$P_{ai} = P_{ao} + K_a w_a^2$$

$$P_{wi} = P_{wo} + K_w w_w^2$$

TYPE 10 Component Configuration

Inputs		Description
1	P_{ao}	– outlet air pressure
2	w_a	– air mass flow rate
3	T_{ai}	– inlet air temperature
4	T_{ao}	– outlet air temperature (from first output)
5	P_{wo}	– outlet water pressure
6	w_w	– water mass flow rate
7	T_{wi}	– inlet water temperature
8	T_{wo}	– outlet water temperature (from second output)

Outputs		Description
1	T_{ao}	– outlet air temperature
2	T_{wo}	– outlet water temperature
3	P_{ai}	– inlet air pressure
4	P_{wi}	– inlet water pressure

Parameters		Description
1	UA	– overall heat transfer coefficient (times area)
2	K_a	– flow resistance parameter, air side
3	K_w	– flow resistance parameter, water side
4	Vol	– water side volume of coil
5	τ_c	– capacitive term of coil time constant

4.11 TYPE 11: HOT WATER TO AIR HEATING COIL (DETAILED)

General Description

Like the TYPE 10 component model, this subroutine represents a water-to-air cross-flow heating coil. Unlike TYPE 10, a constant overall heat transfer coefficient is not assumed: separate inside and outside heat transfer coefficients are calculated as functions of the water and air flow rates. In addition, a more complicated method is used to represent temperature dynamics. The differential equations used to calculate temperature dynamics in this model are derived from transfer functions for the responses of the air and water outlet temperatures to perturbations in the water inlet temperature.

Nomenclature

C_a — capacitance rate (specific heat times mass flow rate) of air
C_m — thermal capacitance (specific heat times mass) of coil material
C_{min} — minimum of C_a and C_w
C_w — capacitance rate (specific heat times mass flow rate) of water
K_a — flow resistance parameter on air side of coil
K_w — flow resistance parameter on water side of coil
NTU   — number of transfer units
R   — ratio of minimum to maximum capacitance rate
s   — Laplace operator
UA   — overall heat transfer coefficient
w_a — mass flow rate of air
w_w — mass flow rate of water
ε — effectiveness
η — fin efficiency
τ — coil time constant
τ_c — capacitive term of coil time constant
τ_x — coil flush time

Subscripts

a   — air
i   — inlet
o   — outlet
ss   — steady state
w   — water

Mathematical Description

Separate inside and outside heat transfer coefficients are calculated as functions of the water and air mass flow rates, w_w and w_a:

$$(\eta hA)_a = aw_a^{\,b}$$

$$(hA)_w = cw_w^{\,d}$$

The parameters a, b, c, and d may be determined from experimental data or calculated from existing correlations (e.g reference [13], pp. 341 and 351), which are generally given in terms of the Reynolds number. The overall heat transfer coefficient is then given by

$$UA = [\ (\eta hA)_a^{-1} + (hA)_w^{-1}\ ]^{-1}$$

and the NTU (number of transfer units) value is, by definition,

$$NTU = \frac{UA}{C_{min}}$$

If a constant UA is desired, b and d may be set to zero.

The coil model uses an approximate equation for effectiveness, ε, as a function of NTU, for a cross-flow heat exchanger with both fluids unmixed [21]:

$$\varepsilon = 1 - \exp\{[\exp(-Rn\{NTU\})-1]/Rn\}$$

where $\quad n = (NTU)^{-0.22}$

The steady state air and water outlet temperatures are found using the definition of ε:

$$T_{aoss} = T_{ai} + (T_{wi} - T_{ai})\varepsilon C_{min}/C_a$$

$$T_{woss} = T_{wi} - (T_{aoss} - T_{ai})C_a/C_w$$

Coil time constants are also calculated by the model, using an assumption that all dynamics are due to changes in the water inlet temperature. Gartner and Harrison [23] give the following approximate transfer function for the water outlet temperature response to perturbations in the water inlet temperature:

$$g_1 = \frac{T'_{wo}(s)}{T'_{wi}(s)} \simeq \frac{\{\exp(-\gamma) + \exp(-\beta_1)\tau_w s\}\exp(-\tau_x s)}{1 + \tau_w s}$$

where

$$\tau_w = \frac{\tau_x}{\alpha} \exp(\frac{\beta_1\beta_3}{2\alpha})$$

$$\alpha = \beta_3 + \frac{2\beta_2}{2+\beta_4}$$

$$\gamma = \beta_1 - \frac{\beta_1\beta_3}{\alpha}$$

$$\beta_1 = \frac{(hA)_w}{C_w}$$

$$\beta_2 = \frac{(\eta hA)_a\tau_x}{C_m}$$

$$\beta_3 = \frac{(hA)_w\tau_x}{C_m}$$

$$\beta_4 = \frac{(\eta hA)_a}{C_a}$$

and η, C_a, C_w, and C_m are defined in the nomenclature section. T'_{wo} and T'_{wi} represent perturbations from some base temperature. Using the air inlet temperature as the base temperature, they are replaced by $(T_{wo} - T_{ai})$ and $(T_{wi} - T_{ai})$, respectively. Treating T_{ai} as a constant, and replacing the steady state solution with the steady state temperature calculated from the coil effectiveness, the following equation can be derived by inverse transformation of the transfer function:

$$\frac{dT_{wo}}{dt} = \frac{1}{\tau_w} \{T_{ai} - T_{wo} + DELAY[\tau_w \exp(-\beta_1)\frac{dT_{wi}}{dt} + T_{woss} - T_{ai}]\}$$

where DELAY is the transport delay function described in section 3.1. Under some conditions the accuracy and reliability of the model may be limited by the characteristics of DELAY. For further discussion of the limitations see section 4.2.

A similar procedure is applied to the air side of the coil. Bhargava et al. [24] give an approximate transfer function of the following form:

$$g_2 = \frac{T'_{ao}(s)}{T'_{wi}(s)} = \frac{a_1}{1 + a_2 \tau_x s + a_3 \tau_x^2 s^2}$$

Reference [24] gives closed-form expressions for the constants a_1, a_2, and a_3, which are derived from a more exact transfer function by the Pade polynomial method. This approximate transfer function can be inverse transformed into a second order differential equation, which can be separated into a pair of coupled first order equations. Assuming a constant air inlet temperature as before, the steady-state air temperature can be substituted into the equations. The resulting pair of first-order differential equations is

$$\frac{dT_d}{dt} = \frac{(T_{aoss} - T_{ao})}{\tau_x}$$

$$\frac{dT_{ao}}{dt} = \frac{T_d - a_2 T_{ao}}{a_3 \tau_a}$$

where

$$a_2 = \frac{\alpha + \beta_1}{\alpha \gamma} - \gamma' e_1$$

$$a_3 = a_2^2 - \frac{(\alpha + \beta_1)^2 - \alpha \gamma}{\alpha^2 \gamma^2} + e_1 [\gamma'(\frac{\gamma'}{2} + \frac{\gamma'}{\gamma} + \frac{2}{\alpha}) - \frac{1}{\alpha}]$$

$$\gamma' = 1 + \frac{\beta_1 \beta_3}{\alpha^2}$$

$$e_1 = \frac{\exp(-\gamma)}{1 - \exp(-\gamma)}$$

and the parameters α, γ, and the β parameters are defined above. T_d is a dummy variable, introduced in the process of separating the second order equation into two first order equations.

Model Validation

The TYPE 11 heating coil model was tested by using experimentally measured temperatures and flow rates as inputs to the model, and comparing the measured air and water outlet temperatures with the calculated values. In figure 4.11.1, simulated and measured air and water outlet temperatures are compared for an experiment in which step changes in the control valve position were used to produce sharp changes in the water inlet temperature. In figure 4.11.2, the dynamics are due to step changes in the water flow rate. In all cases the agreement between simulated and measured outlet temperatures is quite good. Similar results are obtained for step changes in the air inlet temperature and the air flow rate. At least some of the differences must be attributed to experimental error, since the rate of energy transfer calculated from the measured water flow rate and steady state temperatures is not quite equal to the energy transfer rate calculated from the measured air flow rate and steady state temperatures.
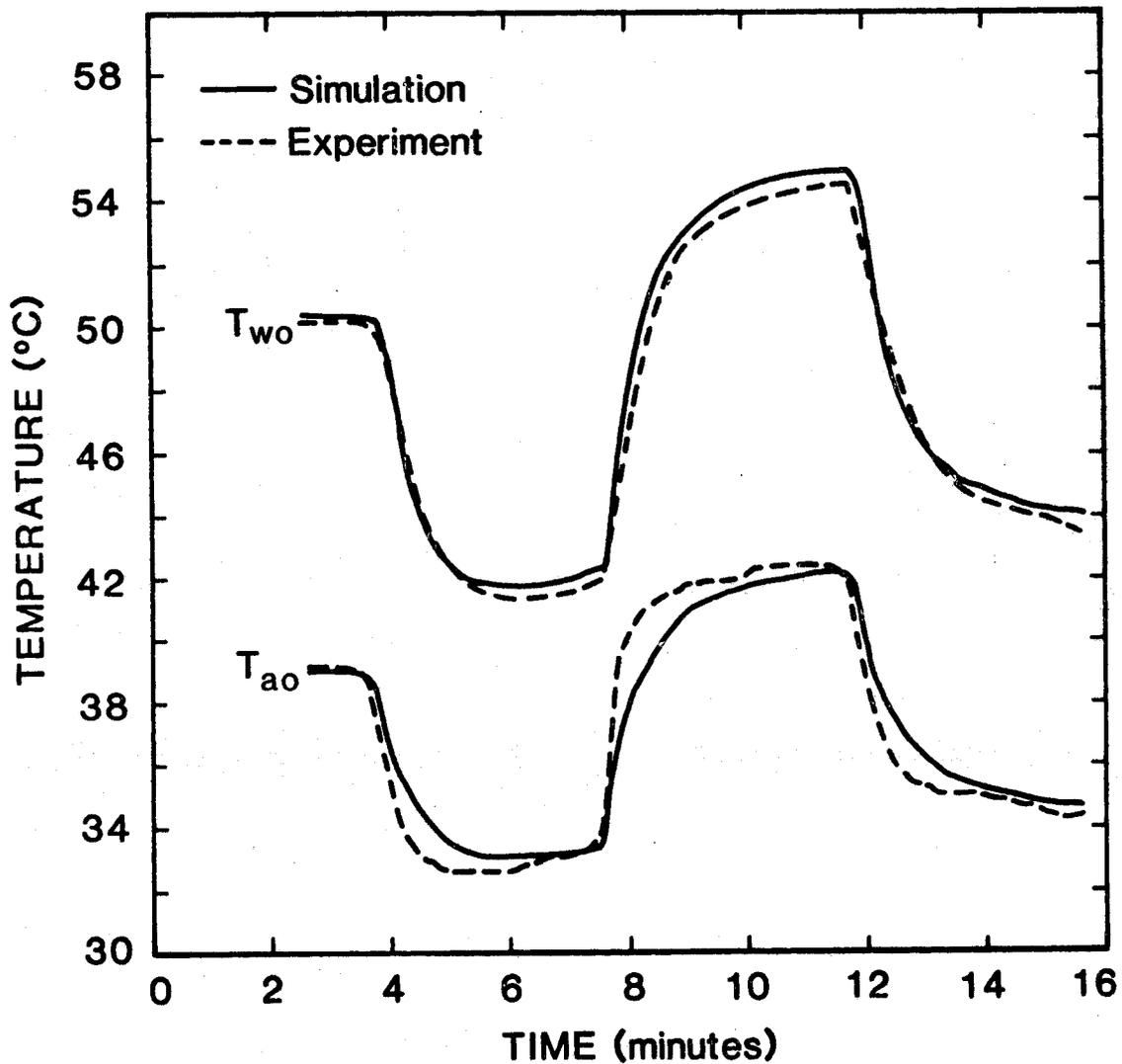
Figure 4.11.1   Heating coil response to rapid changes in the water inlet temperature.
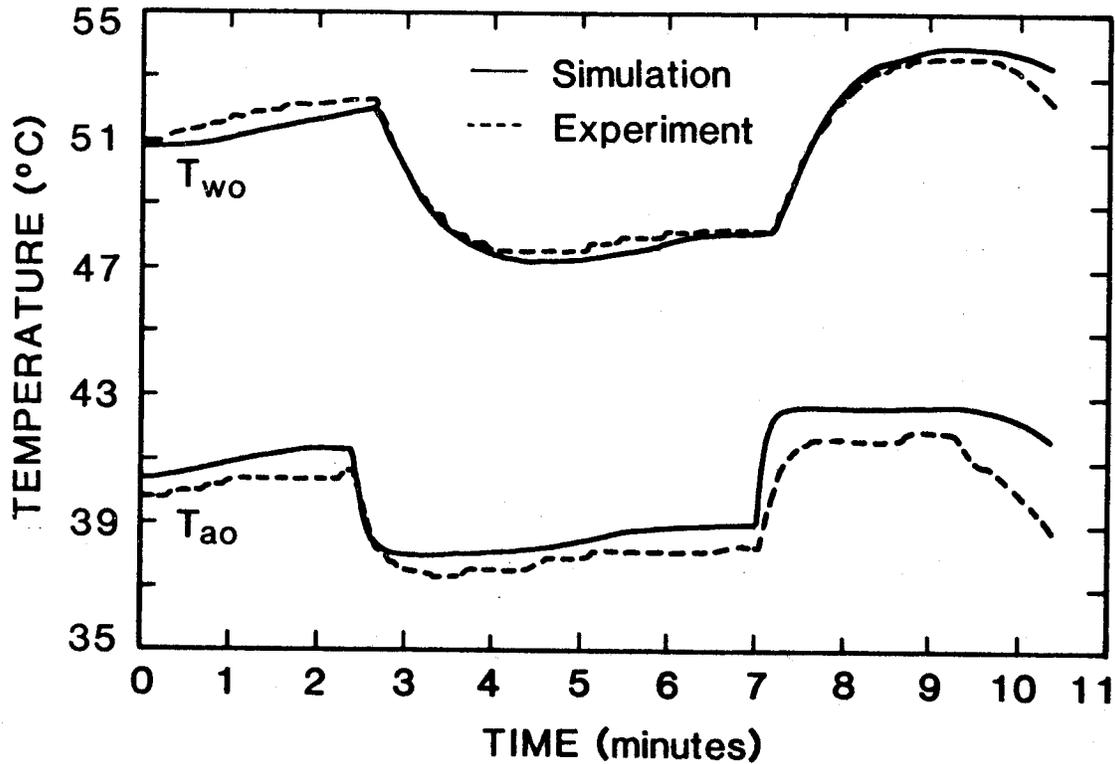
Figure 4.11.2 Heating coil response to rapid changes in the water flow rate.

TYPE 11 Component Configuration

Inputs		Description
1	P_{ao}	– outlet air pressure
2	w_a	– air mass flow rate
3	T_{ai}	– inlet air temperature
4	T_{ao}	– outlet air temperature (from first output)
5	P_{wo}	– outlet water pressure
6	w_w	– water mass flow rate
7	T_{wi}	– inlet water temperature
8	T_{wo}	– outlet water temperature (from second output)
9	T_d	– dummy temperature used to calculate air temperature dynamics (from third output)

Outputs   **Description**

1   T_{ao} – outlet air temperature

2   T_{wo} – outlet water temperature

3   T_d – dummy temperature used internally to calculate air temperature dynamics

4   P_{ai} – inlet air pressure

5   P_{wi} – inlet water pressure

Parameters   **Description**

1   a   – coefficient for the heat transfer correlation $(\eta hA)_a = aw_a^b$

2   b   – coefficient for the heat transfer correlation $(\eta hA)_a = aw_a^b$

3   c   – coefficient for the heat transfer correlation $(hA)_w = cw_w^d$

4   d   – coefficient for the heat transfer correlation $(hA)_w = cw_w^d$

5   K_a – flow resistance parameter, air side

6   K_w – flow resistance parameter, water side

7   Vol   – water side volume of coil

8   C_m – mass times specific heat of coil material

4.12 TYPE 12: COOLING OR DEHUMIDIFYING COIL

General Description

This model represents a circular finned or continuous finned serpentine heat exchanger with four or more rows in counterflow crossflow configuration. The subroutine is based on a program written by Elmahdy and Mitalas [8], with minor modifications to convert it into a TYPE subroutine and to add dynamics to their steady state model.

The subroutine uses equations for log mean temperature difference (LMTD) and log mean enthalpy difference (LMHD) which are strictly correct only for counterflow heat exchangers. When four or more rows are arranged in crossflow counterflow configuration, treating the coil as a counterflow heat exchanger is a good approximation. Use of the model to represent coils with fewer than four rows is not recommended.

The model accounts for condensation on the outside surface of the coil. There are three possible conditions for the coil: all wet, partially wet, or all dry. The subroutine determines which of these conditions applies, and treats each case separately.

TYPE 12 requires a relatively large number of parameters describing the physical characteristics of the coil. Fin efficiencies, heat transfer coefficients, and time constants are calculated within the subroutine from this information. The units given in the nomenclature section must be used.

Nomenclature

A_m — minimum area for air flow (m^2)

A_o — external surface area (m^2)

b — slope of enthalpy vs. temperature line at saturation, $\Delta H/\Delta T$ ($kJ\ kg^{-1}C^{-2}$)

C_f — correction factor (dimensionless)

C_p — specific heat ($kJ\ kg^{-1}C^{-1}$)

C_m — total thermal capacitance of coil material (kJ/C)

D_i — tube inside diameter (m)

G — maximum air mass flux, w_a/A_m ($kg\ m^{-2}s^{-1}$)

h — heat transfer coefficient ($kJ\ kg^{-1}C^{-1}$)

H — enthalpy ($kJ\ kg^{-1}$)

J — Colburn heat transfer J-factor (dimensionless)

Pr — Prandtl number (dimensionless)

Q — heat transfer rate (kW)

Re — Reynolds number (dimensionless)

T_w — bulk water temperature

U — overall heat transfer coefficient, temperature basis ($kW\ m^{-2}\ C^{-1}$)

U_c — overall heat transfer coefficient, enthalpy basis ($kg\ m^{-2}\ s^{-1}$)

V — velocity (m/s)

w — mass flow rate (kg/s)

η — fin efficiency (dimensionless)

τ — time constant (s)

ω — humidity ratio (kg water / kg dry air)

Subscripts

a   - air
d   - dry
i   - inside, inlet
o   - outside, outlet
w   - water, wet

Mathematical Description

The mathematical model has been described in detail elsewhere [8,25]. A brief summary is provided below.

The average heat transfer coefficient on the water side, h_w, is evaluated using the turbulent flow relation [26]:

$$h_w = 1.429 \ [1+0.0146 \ T_w] \ V_w^{0.8} D_i^{-0.2}$$

with all quantities in the units indicated in the nomenclature section. Under dry coil conditions, the average heat transfer coefficient on the air side, h_{ad}, is found from a correlation of the following form:

$$J = C_1 \ Re^{C_2}$$

$$h_{ad} = J \ G \ C_{p,a} \ Pr^{-2/3}$$

The quantities C_1 and C_2 are constants for a particular coil, and are calculated as functions of the coil geometry [27]. Under wet coil conditions, the dry coil heat transfer coefficient is modified as follows:

$$h_{aw} = C_f h_{ad}$$

where the correction factor C_f is a function of the Reynolds number [28].

Separate correlations are used to determine the dry fin efficiency, η_d [9], and the wet fin efficiency, η_w [29]. In addition to the inside and outside heat transfer coefficients and the fin efficiency, the overall heat transfer coefficient includes terms representing the thermal resistance of the tubes, and a constant fouling factor of 5×10^{-5} °C·m^2/W (3×10^{-4} °F·ft^2/Btu).

Under dry coil conditions, the steady state air and water outlet temperatures are found by simultaneous solution of the following three equations:

$$Q = w_a C_{p,a} \ (T_{ai} - T_{ao})$$

$$Q = w_w C_{p,w} \ (T_{wo} - T_{wi})$$

$$Q = UA_o \ (LMTD)$$

where

$$LMTD = \frac{(T_{ai} - T_{wo}) - (T_{ao} - T_{wi})}{\ln(T_{ai} - T_{wo}) - \ln(T_{ao} - T_{wi})}$$

When the coil is wet, enthalpies are used in place of temperatures:

$$Q = w_a (H_{ai} - H_{ao})$$

$$Q = w_w \frac{C_{p,w}}{b} (H_{wo} - H_{wi})$$

$$Q = U_c A_o (LMHD)$$

where

$$LMHD = \frac{(H_{ai} - H_{wo}) - (H_{ao} - H_{wi})}{\ln(H_{ai} - H_{wo}) - \ln(H_{ao} - H_{wi})}$$

H_{wi} and H_{wo} are saturated <u>air</u> enthalpies calculated at the water temperatures T_{wi} and T_{wo}, respectively, according to the following relation:

$$H_w = H_o + b\, T_w$$

The quantities H_o and b are determined iteratively.

Initially the entire air side surface of the coil is assumed to be wet. Based on this assumption, the outlet air and water temperatures and the air side surface temperatures at inlet and outlet are calculated. If the surface temperature at the inlet is lower than the inlet air dew point temperature, the coil is in fact all wet. If the surface temperature at the air outlet is higher than the inlet air dew point temperature, the coil surface is all dry. If neither of these conditions is met, the surface is partly wet and partly dry. In this case an iterative procedure is used to find the position in the air flow direction at which the surface temperature is equal to the dew point temperature. This position is the boundary between the dry and wet sections of the coil.

Dynamics have been added to the steady state model in a very simple and somewhat artificial manner:

$$\frac{dT'_{ao}}{dt} = \frac{T_{ao} - T'_{ao}}{\tau}$$

$$\frac{dT'_{wo}}{dt} = \frac{T_{wo} - T'_{wo}}{\tau}$$

$$\frac{d\omega'_{ao}}{dt} = \frac{\omega_{ao} - \omega'_{ao}}{\tau}$$

where

$$\tau = \frac{C_m}{UA_o}$$

Note that the dynamic variables T'_{ao}, T'_{wo}, and ω'_{ao} are used only in the three

65

differential equations given above. Steady state variables are used in determining whether the coil is wet or dry.

Model Validation

The steady state cooling coil model has been tested extensively by its authors. Comparisons between the steady state model and experimental data for a variety of coils and operating conditions can be found elsewhere [30,31].

The dynamics of the coil model have been investigated experimentally for changes in the water flow rate and for changes in the water inlet temperature. In both experiments, measured air and water inlet temperatures and a measured water flow rate were used as inputs to the model. The air flow rate was calculated from the measurements under steady state conditions. In both cases the outside surface of the coil remained dry. Measured outlet temperatures are compared with calculated temperatures in figures 4.12.1 and 4.12.2. Agreement between the model and the experiments is excellent.
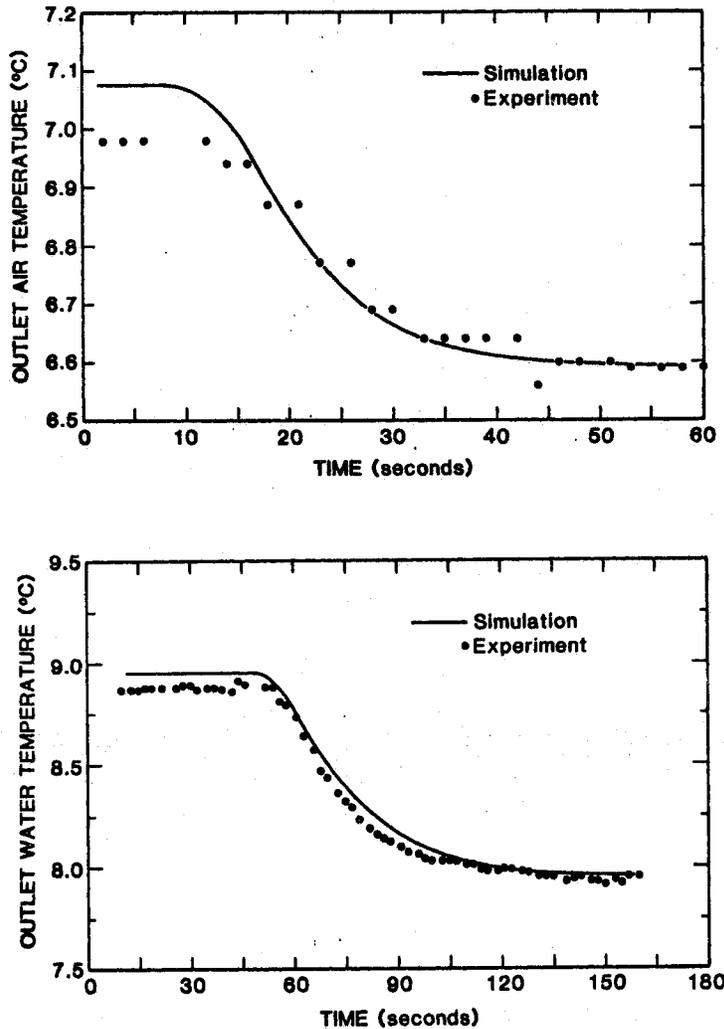
Figure 4.12.1 Cooling coil response to changes in the water flow rate.
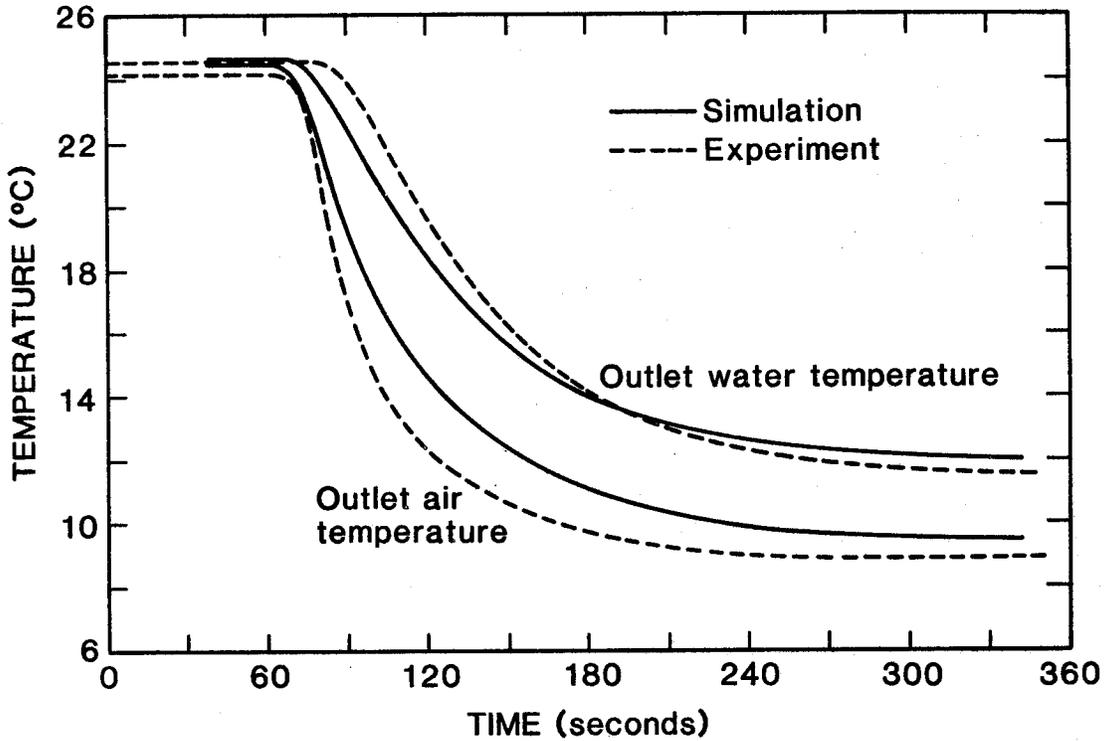
Figure 4.12.2   Cooling coil response to changes in the water inlet temperature

TYPE 12 Component Configuration

Inputs		Description
1	w_w	− water mass flow rate
2	T_{wi}	− inlet water temperature
3	w_a	− dry air mass flow rate
4	T_{ai}	− inlet air dry bulb temperature
5	ω_{ai}	− inlet air humidity ratio
6	T'_{wo}	− dynamic outlet water temperature
7	T'_{ao}	− dynamic outlet air temperature
8	ω'_{ao}	− dynamic outlet air humidity ratio

67

Outputs		Description
1	T'_{wo}	– dynamic outlet water temperature [C]
2	T'_{ao}	– dynamic outlet air temperature [C]
3	ω'_{ao}	– dynamic outlet air humidity ratio [–]
4	Q_t	– total cooling load at steady state [kW]
5	Q_s	– sensible cooling load at steady state [kW]
6	f_w	– wet fraction of total surface area [–]

Parameters		Description
1	I_c	– coil type: 0 = flat continuous fins, 1 = circular fins
2	A_p	– primary (tube exterior) surface area [m^2]
3	A_s	– secondary (fin) surface area [m^2]
4	A_i	– internal surface area [m^2]
5	A_m/A_f	– ratio of minimum flow area to face area [–]
6	k_f	– thermal conductivity of fin material [kW $m^{-1}C^{-1}$]
7	A_f	– coil face area [m^2]
8	N_f	– number of fins per centimeter
9	N_p	– number of tubes per row
10	N_r	– number of rows
11	D_o	– tube outside diameter [m]
12	D_i	– tube inside diameter [m]
13	δ	– fin thickness [m]
14	C_m	– mass times specific heat of coil material [kJ/C]
15	S_1	– tube longitudinal spacing (in air flow direction) [m]
16	D_f	– fin diameter (if I_c = 1) or fin length (if I_c = 0) [m]
17	D_c	– coil depth in air flow direction [m]
18	k_t	– thermal conductivity of tube material [kW $m^{-1}C^{-1}$]

68

4.13 TYPE 13: THREE-WAY VALVE WITH ACTUATOR

General Description

This component represents a valve with two inlet ports and one outlet port. The position of the valve is determined by an input control signal, C. Port 1 of the valve is closed when C=0 and open when C=1. The sixth parameter, MODE, determines the inherent characteristics of the two inlet ports. If MODE=0, both ports are linear. If MODE=1, port 1 is exponential (equal percentage) and port 2 is linear. If MODE=2, both ports are exponential. As discussed in Section 4.0, the actual behavior of the valve model in a system simulation will depend on the valve's authority over the system.

The valve model includes a pneumatic actuator model. Actuator hysteresis is represented using the utility subroutine HYSTER, described in chapter 3. The dynamic response of the actuator is modeled by a first order differential equation. The actuator can be effectively removed by setting the time constant and the hysteresis parameter to zero.

When the flow rate through the valve goes to zero, the pressure drop across the valve becomes indeterminate. To prevent this problem, a non-zero leakage parameter is required by the model.

Nomenclature

C — requested relative valve position $(0 < C < 1)$
C_a — relative actuator position $(0 \leq C_a < 1)$
C_{h1} — relative opening of port 1 $(0 \leq C_h \leq 1)$
C_{h2} — relative opening of port 2 $(0 \leq C_h < 1)$
hys — fraction of actuator's range over which C_h remains constant when actuator's direction of travel reverses
K_1 — flow resistance parameter for inlet port 1, when port 1 is open $(C_h = 1)$
K_2 — flow resistance parameter for inlet port 2, when port 2 is open $(C_h = 0)$
MODE — determines port 1 / port 2 inherent characteristics:
      0 => linear/linear
      1 => exponential/linear
      2 => exponential/exponential
w_1 — mass flow rate at inlet port 1
w_2 — mass flow rate at inlet port 2
w_3 — mass flow rate at outlet port 3
λ — leakage parameter: fractional leakage when $\Delta P = 1$.
τ — actuator time constant

Mathematical Description

The relationship between the input control signal, C, and the actuator position, C_a, is defined by:

$$\frac{dC_a}{dt} = \frac{C - C_a}{\tau}$$

This differential equation is solved by MODSIM unless the time constant, τ, is

69

less than one second, in which case the following solution is used:

IF $(\tau/\Delta t) < 0.05$ OR $|C_{ss} - C^-| < 10^{-10}$ THEN

$$C_a = C$$

ELSE

$$C_a = C - (C - C_a^-)\exp(-\Delta t/\tau)$$

where C_a^- is the value of C_a one time step ago.

The actuator position differs from the valve position due to hysteresis effects, which are determined by the utility function HYSTER described in section 3.2.

$$C_{h1} = \text{HYSTER}(C_a, \text{hys})$$

$$C_{h2} = 1. - C_{h1}$$

The outlet flow rate is the sum of the inlet flow rates, and the outlet temperature is a weighted average of the inlet temperatures:

$$w_3 = w_1 + w_2$$

$$T_3 = (w_1 T_1 + w_2 T_2) / w_3$$

Calculation of the inlet pressures depends upon the mode.

Mode = 0:

$$P_1 = P_3 + \text{sign}(w_1)K_1 w_1^2 [(1-\lambda)C_{h1} + \lambda]^{-2}$$

$$P_2 = P_3 + \text{sign}(w_2)K_2 w_2^2 [(1-\lambda)C_{h2} + \lambda]^{-2}$$

Mode = 1:

$$P_1 = P_3 + \text{sign}(w_1)K_1 w_1^2 \lambda^{-2C_{h2}}$$

$$P_2 = P_3 + \text{sign}(w_2)K_2 w_2^2 [(1-\lambda)C_{h2} + \lambda]^{-2}$$

Mode = 2:

$$P_1 = P_3 + \text{sign}(w_1)K_1 w_1^2 \lambda^{-2C_{h2}}$$

$$P_2 = P_3 + \text{sign}(w_2)K_2 w_2^2 \lambda^{-2C_{h1}}$$

70

TYPE 13 Component Configuration

Inputs		Description
1	w_1	– mass flow rate, inlet port 1
2	w_2	– mass flow rate, inlet port 2
3	P_3	– pressure at outlet
4	T_1	– temperature, inlet port 1
5	T_2	– temperature, inlet port 2
6	C	– input control signal ($0 \leq C \leq 1$)
7	C_a	– actuator relative position

Outputs		Description
1	C_a	– actuator relative position
2	w_3	– outlet mass flow rate
3	P_1	– pressure at inlet port 1
4	P_2	– pressure at inlet port 2
5	T_3	– temperature at outlet
6	C	– valve stem relative position

Parameters		Description
1	K_1	– port 1 flow resistance when port 1 is open
2	K_2	– port 2 flow resistance when port 2 is open
3	λ	– leakage parameter
4	τ	– actuator time constant
5	hys	– hysteresis parameter
6	Mode	– determines characteristics of ports 1 and 2:

$$0 \Rightarrow \text{linear/linear}$$
$$1 \Rightarrow \text{exponential/linear}$$
$$2 \Rightarrow \text{exponential/exponential}$$

71

4.14 TYPE 14: EVAPORATIVE HUMIDIFIER

General Description

This component, which is based on a model developed by Chi [32], represents an air stream passing over a water pan or through a porous pad, increasing the humidity and decreasing the temperature of the air. The model assumes that the process is adiabatic, which means that the water temperature is assumed to be equal to the entering air wet bulb temperature. In addition, the process is assumed to occur at constant pressure. Any pressure drop through the humidifier may be modeled by increasing the flow resistance of an adjacent component.

The evaporative humidifier model requires the inlet air temperature, humidity ratio, and mass flow rate as inputs. It computes the exit humidity ratio, temperature, and mass flow rate. The wet bulb temperature and total enthalpy of the air remain constant while the moisture content rises and the exit dry bulb temperature decreases. The mass flow rate increases slightly due to the addition of moisture to the air.

Nomenclature

C_p — specific heat at constant pressure
H — enthalpy
hA — water to air heat transfer coefficient times area
T — temperature
w — mass flow rate
ω — humidity ratio

Subscripts

a   — air
fg   — latent heat of vaporization
i   — inlet
o   — outlet
ref   — reference point at which H = 0 (−20°C)
s   — saturation
v   — water vapor
wb   — wet bulb

Mathematical Description

The total enthalpy at the inlet is determined by the following equation :

$$H_{ai} = \frac{\omega_{ai}[C_{pv}(T_{ai} - T_{ref}) + \Delta H_{fg}] + C_{pa}(T_{ai} - T_{ref})}{1 + \omega_{ai}}$$

Next, the inlet wet bulb temperature is calculated, using a correlation from reference [8]:

$$T_{wb} = 30.9185 - 39.682 \, C + 20.5841 \, C^2 - 1.758 \, C^3$$

where $C = \log_e(H_{ai})$

The outlet saturation humidity ratio is calculated next, using the subprogram PSATS described in Chapter 3 to find the saturation pressure at the inlet wet bulb temperature. The constant in the following equation is the water to air molecular weights ratio.

$$\omega_{aos} = \frac{0.62198\ PSATS(T_{wb})}{P_{atm} - PSATS(T_{wb})}$$

where P_{atm} is 101.325 kPa. The outlet air humidity ratio, temperature, and mass flow rate can now be determined.

$$\omega_{ao} = \omega_{ai} + (\omega_{aos} - \omega_{ai})(1 - e^{-R})$$

where $R = \dfrac{hA}{C_{pa}w_{ai}}$

The outlet air temperature and mass flow rate are given by

$$T_{ao} = T_{ai} - \frac{\omega_{ao} - \omega_{ai}}{C_{pa}}\Delta H_{fg}$$

$$w_{ao} = \frac{1 + \omega_{ao}}{1 + \omega_{ai}}\ w_{ai}$$

TYPE 14 Component Configuration

Inputs		Description
1	T_{ai}	– inlet air temperature [C]
2	ω_{ai}	– inlet air humidity ratio [–]
3	w_{ai}	– inlet moist air mass flow rate [kg/s]

Outputs		Description
1	ω_{ao}	– outlet air humidity ratio [–]
2	T_{ao}	– outlet air temperature [C]
3	w_{ao}	– outlet moist air mass flow rate [kg/s]

Parameters		Description
1	hA	– water to air heat transfer coefficient times area [KW/°C]

73

4.15 TYPE 15: ROOM MODEL

General Description

The room model has been described in reference [33], and is loosely based on reference [34]. The model considers the air mass, an interior mass, and a wall mass. No modeling of the building shell is attempted, and heat flows are treated as inputs to the model.

The air mass is artificially divided into two parts: a fully mixed part near the ventilation supply, and a piston flow part near the ventilation exhaust. The fraction of the air mass in these two parts is adjustable. Each part of the air mass exchanges heat with two kinds of room mass: an interior mass and a wall mass. The temperature of each room mass is assumed to be uniform. Two heat flows are also defined: a heat flow into the wall mass, representing conduction through the walls, and a direct heat flow into the fully mixed air mass, representing internal gains.

Nomenclature

C_a — specific heat of air
C_I — specific heat of interior mass
C_w — specific heat of wall mass
f — fraction of air mass which is fully mixed
H_I — heat transfer coefficient times area of interior mass
H_w — heat transfer coefficient times area of wall mass
M_I — interior mass
M_w — wall mass
Q_I — heat flow due to internal gains
Q_w — conduction heat flow into wall mass
T_{ai} — ventilation air inlet temperature
T_{ao} — exhaust air temperature
T_I — interior mass temperature
T_m — temperature of fully mixed portion of air mass
T_p — temperature of piston flow portion of air mass
T_r — average room air temperature
T_w — wall mass temperature
V — volume of room air mass
w — mass flow rate of ventilation air
ρ — density of air

Mathematical Description

The fully mixed air mass temperature is given by the following equation:

$$(f\rho V C_a)\, \frac{dT_m}{dt} = Q_I + wC_a(T_{ai} - T_m) + fH_w(T_w - T_m) + fH_I(T_I - T_m)$$

The piston flow part of the air mass exchanges heat with the interior mass and the wall mass as it moves through the room according to the following equation:

74

$$wLC_a \frac{dT_p(x)}{dx} = (1 - f)[H_W(T_W - T_p(x)) + H_I(T_I - T_p(x))]$$

where L is the total distance moved by the air mass. The solution to this equation is:

$$T_p(x) = T_s - (T_s - T_m)\exp(-ax/L)$$

where

$$T_s = \frac{H_W T_W + H_I T_I}{H_W + H_I}$$

$$a = \frac{H_W + H_I}{wC_a}(1 - f)$$

and $T_p(x)$ is the temperature of the air as a function of the distance traveled. The spatial average air mass temperature is approximately given by:

$$\frac{d\overline{T}_p}{dt} = \frac{w(\overline{T}_s - \overline{T}_p)}{\rho V(1 - f)}$$

where

$$\overline{T}_s = T_s - (T_s - T_m)\frac{(1 - \exp(-a))}{a}$$

The room mass temperatures are given by the following equations:

$$M_W C_W \frac{dT_W}{dt} = Q_W + H_W(T_r - T_W)$$

$$M_I C_I \frac{dT_I}{dt} = H_I(T_r - T_I)$$

$$T_r = fT_m + (1-f)\overline{T}_p$$

The exhaust air temperature is equal to the temperature of the piston flow portion of the air mass at x=L:

$$T_{ao} = T_s - (T_s - T_m)\exp(-a)$$

A transport delay is applied to the exhaust air temperature using the utility subroutine DELAY with a delay time equal to the flush time for the piston flow fraction of the room volume.

75

TYPE 15 Component Configuration

Inputs		Description
1	w	– mass flow rate of ventilation air
2	T_{ai}	– ventilation air inlet temperature
3	T_m	– temperature of fully mixed portion of air mass
4	T_w	– wall mass temperature
5	T_I	– interior mass temperature
6	$\bar{T}_p$	– spatial average temperature of the piston flow portion of the air mass
7	Q_w	– conduction heat flow into wall mass
8	Q_I	– heat flow due to internal gains

Outputs		Description
1	T_m	– temperature of fully mixed portion of air mass
2	T_w	– wall mass temperature
3	T_I	– interior mass temperature
4	$\bar{T}_p$	– spatial average temperature of the piston flow portion of the air mass
5	T_r	– average room air temperature
6	T_{ao}	– exhaust air temperature

Parameters		Description
1	V	– volume of room air mass
2	$M_w C_w$	– thermal capacitance of walls
3	$M_I C_I$	– thermal capacitance of interior mass
4	H_w	– heat transfer coefficient times area for the wall mass
5	H_I	– heat transfer coefficient times area for the interior mass
6	f	– fraction of the air mass which is fully mixed

4.16 TYPE 16: 'STICKY' PROPORTIONAL CONTROLLER

General Description

This component represents a proportional controller with upper and lower limits on the rate of change of the output control signal. The limits are determined by the second parameter, B. If the error signal changes by an amount less than B, the change is ignored and the output control signal remains constant until the cumulative changes in error equal or exceed B. A change in the error signal with a magnitude greater than 2B is treated as a change in error of magnitude B. Thus for purposes of calculating the output control signal, all changes in error lie between B and 2B. The measured input signal, generally representing a temperature, must be a control variable between zero and one, which can be obtained by passing a temperature through the TYPE 7 temperature sensor model.

NOTE: this component is to be used _only_ to pass control signals between superblocks. The variable defined by its output should _not_ be solved simultaneously.

Nomenclature

B – error band
C – output control signal $(0 \leq C \leq 1)$
C_{Tm} – measured temperature input control signal $(0 \leq C_{Tm} < 1)$
C_{Ts} – temperature setpoint control signal $(0 \leq C_{Ts} \leq 1)$
E – effective error at current simulation time
E_o – effective error at previous simulation time
K_o – offset
K_p – proportional gain

Mathematical Description

$$\Delta E = C_{Ts} - C_{Tm} - E_o$$

If $(|\Delta E| < B)$ $E = E_o$

If $(B \leq |\Delta E| \leq 2B)$ $E = C_{Ts} - C_{Tm}$

If $(2B < |\Delta E|)$ $E = E_o + (B)\,\text{sign}(\Delta E)$

$$C = K_o + K_p E$$

If $(C < 0)$ $C = 0$

If $(C > 1)$ $C = 1$

TYPE 16 Component Configuration

Inputs		Description
1	C_{Tm}	– measured temperature input control signal
2	C_{Ts}	– temperature setpoint control signal

Outputs		Description
1	C	– controller output

Parameters		Description
1	K_p	– proportional gain
2	B	– error band
3	K_o	– offset

4.17 TYPE 17: MIXING DAMPERS AND MERGE

General Description

TYPE 17 represents a pair of mixing dampers, mechanically linked so that one damper opens as the other closes. The first inlet is closed when the input control signal, C, is zero, and open when C is one. The reverse is true for the second inlet. Hysteresis effects and an actuator time constant are not included in the model.

Flow resistances are calculated in the same manner as in the TYPE 5 damper or valve model. For a discussion of damper characteristics and recommended parameter values, see section 4.5.

The TYPE 17 component takes three pressures as inputs and calculates three flow rates. In this respect it is similar to the TYPE 21 flow split and the TYPE 3 inlet conduit, both of which also compute an inlet flow rate rather than an inlet pressure.

Nomenclature

C   – control signal $(0 \leq C \leq 1)$
K_o – flow resistance coefficient when damper is open
P   – pressure
R   – flow resistance as a function of damper position
T   – temperature
w   – mass flow rate
W_f – weighting factor for linear term of flow resistance

Subscripts

1   – first inlet
2   – second inlet
3   – outlet

Mathematical Description

R_1 and R_2 are the flow resistances of the two inlets:

$$R_1 = \frac{W_f K_o}{[(1 - \lambda)C + \lambda]^2} + (1 - W_f)K_o \lambda^{-2(1-C)}$$

$$R_2 = \frac{W_f K_o}{[(1 - \lambda)(1 - C) + \lambda]^2} + (1 - W_f)K_o \lambda^{-2C}$$

These equations model linear damper characteristics when W_f is one, exponential characteristics when W_f is zero, and intermediate characteristics for intermediate values of W_f. These flow resistances are used to find two of the three flow rates:

$$w_1 = \text{sign}(P_1 - P_3) \left[\frac{|P_1 - P_3|}{R_1} \right]^{0.5}$$

$$w_2 = \text{sign}(P_2 - P_3) \left[\frac{|P_2 - P_3|}{R_2} \right]^{0.5}$$

The third flow rate is the sum of the first two:

$$w_3 = w_1 + w_2$$

The outlet temperature is a weighted average of the two inlet temperatures:

$$T_3 = \frac{w_1 T_1 + w_2 T_2}{w_3}$$

TYPE 17 Component Configuration

Inputs		Description
1	P_1	— pressure at first inlet
2	P_2	— pressure at second inlet
3	P_3	— pressure at outlet
4	T_1	— temperature at first inlet
5	T_2	— temperature at second inlet
6	C	— input control signal

Outputs		Description
1	w_1	— flow rate at first inlet
2	w_2	— flow rate at second inlet
3	w_3	— flow rate at outlet
4	T_3	— temperature at outlet

Parameters		Description
1	K_o	— full open resistance [.001/(kg-m)]
2	λ	— leakage parameter [-]
3	W_f	— weighting factor for linear term of flow resistance [-]

4.18 TYPE 18: PLENUM

General Description

The output of this component is simply the sum of its inputs. The single parameter, n, specifies the number of input flow rates to be summed. Up to ten flow rates may be added together. Ten input flow rates are required by the model, but only the first n are summed. The remaining inputs are ignored.

Nomenclature

> n – number of input mass flow rates ($1 \leq n \leq 10$)
> w_i – i^{th} input mass flow rate (kg/s)
> w_{out} – output mass flow rate

Mathematical Description

$$w_{out} = w_1 + w_2 + \ldots + w_n$$

NOTE: Inputs $(n + 1)$ through 10 must be specified in the simulation, but are ignored. Only the first n inputs are used.

TYPE 18 Component Configuration

Inputs		Description
1	w_1	– first mass flow rate
2	w_2	– second mass flow rate
:		
n	w_n	– nth mass flow rate
:		
10	w_{10}	– tenth mass flow rate

Outputs		Description
1	w_{out}	– outlet mass flow rate

Parameters		Description
1	n	– number of input mass flow rates

4.19 TYPE 19: FLOW BALANCE CONTROL

<u>General Description</u>

This subroutine does not represent any physical component. It serves instead to minimize mathematical problems which arise when a closed loop flow stream is divided between two superblocks. It functions by providing one superblock with an estimate of the net flow resistance encountered by the flow stream in another superblock. The purpose and use of TYPE 19 is most easily explained by way of an example.

Figure 4.19.1 represents a very simple flow circuit consisting of two ducts, a fan, and a damper. Figure 4.19.2 shows one possible way of modeling the system, with half of the circuit in one block and half in another. The arrows indicate the flow of information among units and blocks. For the purpose of this example, only pressures and flow rates are shown. Pressure P_1 is a boundary condition, providing a baseline from which pressure changes can be calculated. Pressure P_2 and flow rate w_1 are determined simultaneously within block 1. Pressure P_4 is solved within block 2. Pressure P_3 is determined by simultaneous solution of blocks 1 and 2, provided that the two blocks are in the same superblock. However, MODSIM assumes that couplings between superblocks are weak enough so that each superblock can be treated as an independent subsystem. No simultaneous equations are defined between superblocks. Thus if blocks 1 and 2 are in different superblocks, they will be solved sequentially rather than simultaneously and a self-consistent solution to the full set of equations defining pressures and flow rates is unlikely to be achieved.
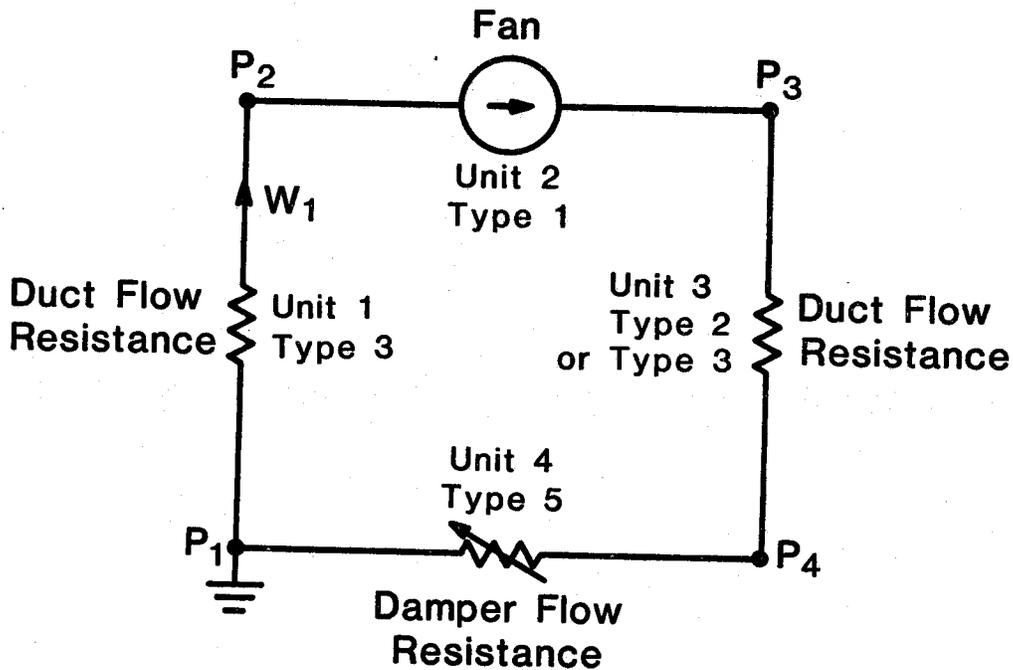

Figure 4.19.1 Schematic diagram of a simple closed-loop system.
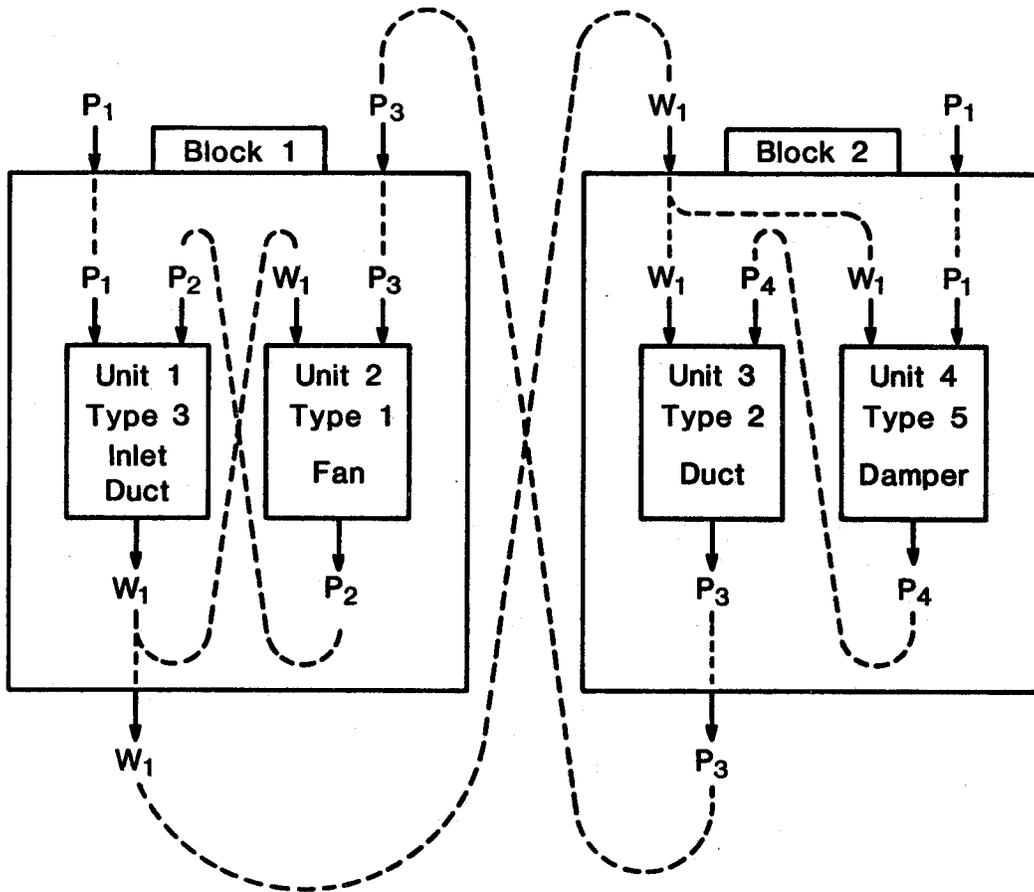
Figure 4.19.2.   Information flow diagram for a simulation of the system shown in figure 4.19.1.

In Figure 4.19.3, the same system is modeled using the Type 19 flow balance component. Flow rates w_1 and w_2 are calculated essentially independently in the two blocks, although both represent the same flow stream. Type 19 maintains internally a variable flow resistance representing the net flow resistance in block 2. At each time step it examines the difference between w_1 and w_2, and modifies its internal resistance in such a way as to reduce the difference to zero. It then uses this resistance to calculate P_3, which is ultimately involved in determining both w_1 and w_2. Note that in both Figure 4.19.2 and Figure 4.19.3, the same·amount of information passes between super-blocks: a flow rate is exchanged for a pressure. The important difference is that in the latter case, the Type 19 component actively takes responsibility for bringing the two superblocks to convergence.
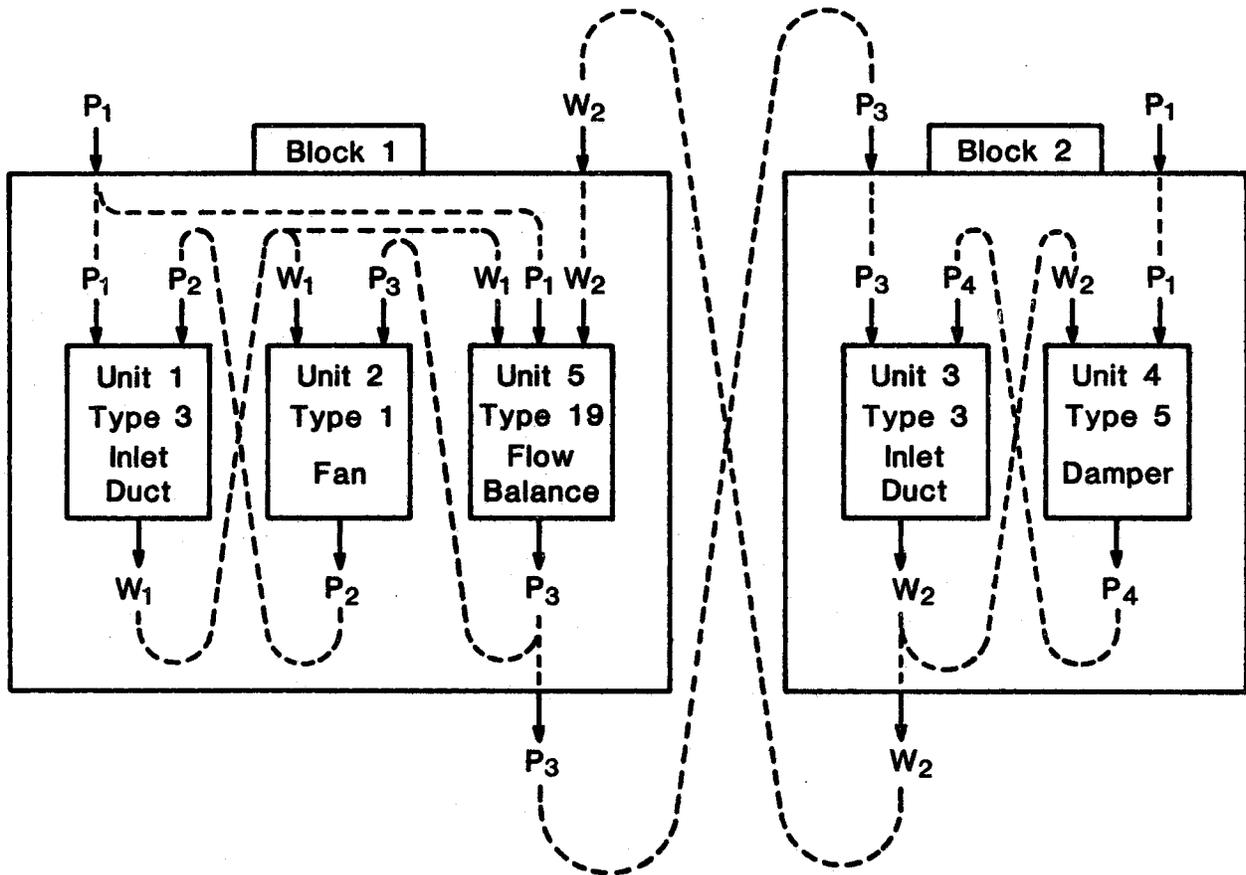
Figure 4.19.3. Information flow diagram for a simulation of the same system in two superblocks, using the TYPE 19 component.

Nomenclature

K — flow resistance parameter at current simulation time

K_p — flow resistance parameter at previous simulation time

P_i — upstream pressure, where flow stream exits this SB

P_o — downstream pressure, where flow stream enters this SB

w — mass flow rate determined in this SB

w' — mass flow rate determined in another SB

ΔP — upstream pressure minus downstream pressure

Δw — w' minus w

SB — superblock

84

Mathematical Description

By definition,

$$K = \frac{\Delta P}{w^2}$$

from which

$$\frac{dK}{dw} = \frac{-2\Delta P}{w^3} = \frac{-2K}{w}$$

The change in K which will eliminate the difference between w' and w is approximated by

$$\Delta K = \frac{-2K\Delta w}{w}$$

or

$$\Delta K = 2K(1 - \frac{w'}{w})$$

and the new value of K is

$$K = K_p + \Delta K$$

The upstream pressure can then be calculated:

$$P_i = P_o + Kw^2$$

TYPE 19 Component Configuration

Inputs		Description
1	w	– mass flow rate determined in this superblock
2	P_o	– downstream pressure, where flow stream enters this superblock
3	w'	– mass flow rate determined in another superblock

Outputs		Description
1	P_i	– upstream pressure, where flow stream exits this superblock

Parameters		Description
1	K_0	– flow resistance parameter at time = 0.

4.20 TYPE 20: HIGH LIMIT OR LOW LIMIT CONTROLLER

General Description

This component represents a proportional controller which exerts a control action only when the measured input signal is on one side of a fixed setpoint. For a high limit controller, no control action is exerted unless the measured input signal exceeds the setpoint. For a low limit controller, control action is exerted only when the controlled input falls below the setpoint. The setpoint is determined by the absolute value of the second parameter. The component represents a high limit controller when the second parameter is positive, and a low limit controller when the second parameter is negative. The measured input signal, generally representing a temperature, must be a control variable between zero and one, which can be obtained by passing a temperature through the TYPE 7 temperature sensor model. The setpoint parameter must also be a normalized control variable with an absolute value between zero and one.

Nomenclature

C – output control signal $(0 \le C \le 1)$

C_{Tm} – measured temperature input control signal $(0 \le C_{Tm} < 1)$

C_{Ts} – temperature setpoint parameter $(0 \le |C_{Ts}| \le 1)$

K_p – proportional gain

Mathematical Description

$$E = |C_{Ts}| - C_{Tm}$$

High limit controller $(C_{Ts} \ge 0)$:

If $(C_{Tm} > C_{Ts})$ $C = C + K_p E$

Low Limit Controller $(C_{Ts} < 0)$:

If $(C_{Tm} < |C_{Ts}|)$ $C = C + K_p E$

TYPE 20 Component Configuration

Inputs		Description
1	C_{Tm}	– measured temperature input control signal
2	C	– controller output

Outputs		Description
1	C	– controller output

Parameters		Description
1	K_p	– proportional gain
2	C_{Ts}	– setpoint times i, where i = -1 for low limit controller, or i = 1 for high limit controller

4.21   TYPE 21: 'GROUNDED' WATER OR AIR SPLIT

General Description

This component models the division of a flow stream into two flowstreams. It is referred to as 'grounded' by analogy to a grounded electrical junction: the pressure at the center of the split is a constant, entered as a parameter. Using as inputs the pressures at the two outlets, the subroutine calculates the pressure and mass flow rate at the single inlet and the mass flow rates at the two outlets. The model assumes that the flow resistance parameter is the same for all three branches. Care has been taken to ensure that correct results are obtained even when the flow direction of one or more flow streams is reversed.

Nomenclature

K – flow resistance coefficient
P – pressure
w – mass flow rate

Subscripts

0   – center of split
1   – inlet
2   – first outlet
3   – second outlet

Mathematical Description

$$w_2 = \text{sign}(P_0 - P_2) \left[\frac{P_0 - P_2}{K} \right]^{0.5}$$

$$w_3 = \text{sign}(P_0 - P_3) \left[\frac{P_0 - P_3}{K} \right]^{0.5}$$

$$w_1 = w_2 + w_3$$

$$P_1 = P_0 + \text{sign}(w_1) \frac{K}{2} w_1^2$$

TYPE 21 Component Configuration

Inputs		Description
1	P_2	– pressure at first outlet
2	P_3	– pressure at second outlet

Outputs		Description
1	P_1	– pressure at inlet
2	w_1	– flow rate at inlet
3	w_2	– flow rate at first outlet
4	w_3	– flow rate at second outlet

Parameters		Description
1	P_0	– pressure at center of split
2	K	– flow resistance coefficient

4.22 TYPE 22 : STEAM SPRAY HUMIDIFIER

General Description

The steam spray humidifier model calculates an outlet air temperature, flow rate, and humidity ratio for a constant pressure process in which steam is injected into an air flow stream to increase the humidity of the air. The outlet air humidity ratio is limited either by the steam flow rate or by the saturation humidity ratio (at atmospheric pressure) times a saturation efficiency. The mass flow rate of the air increases slightly due to the addition of moisture.

Nomenclature

C_{pa} — specific heat of dry air [kJ/(kg K)]
C_{pai} — specific heat of moist inlet air [kJ/(kg K)]
C_{ps} — specific heat of steam or water vapor [kJ/(kg K)]
P_{atm} — atmospheric pressure (101.325 kPa)
T_{ai} — temperature of inlet air (C)
T_{ao} — temperature of outlet air (C)
T_s — temperature of inlet steam (C)
w_{ai} — mass flow rate of moist inlet air (kg/s)
w_{ao} — mass flow rate of moist outlet air (kg/s)
w_s — mass flow rate of inlet steam (kg/s)
η — saturation efficiency (–)
ω_{ai} — inlet air humidity ratio (kg water / kg dry air)
ω_{ao} — outlet air humidity ratio (kg water / kg dry air)
ω_{aos} — saturation humidity ratio (kg water / kg dry air)

Mathematical Description

The inlet air specific heat depends on the inlet air humidity ratio:

$$C_{pai} = \frac{C_{pa} + \omega_{ai} C_{ps}}{1 + \omega_{ai}}$$

where C_{pa} is the specific heat of dry air and C_{ps} is the specific heat of steam or water vapor. The air outlet temperature is a weighted average of the inlet air and steam temperatures:

$$T_{ao} = \frac{w_s C_{ps} T_s + w_{ai} C_{pai} T_{ai}}{w_s C_{ps} + w_{ai} C_{pai}}$$

Assuming that all of the entering steam is absorbed by the air, the outlet humidity ratio is

$$\omega_{ao} = \omega_{ai} + \frac{w_s (1 + \omega_{ai})}{w_{ai}}$$

The saturation humidity ratio at the outlet temperature is calculated next, using the subprogram PSATS described in Chapter 3 to find the saturation

pressure at the outlet temperature. The constant in the following equation is the ratio of water to air molecular weights.

$$\omega_{aos} = \frac{0.62198 \ PSATS(T_{ao})}{P_{atm} - PSATS(T_{ao})}$$

where P_{atm} is 101.325 kPa. The outlet humidity ratio, ω_{ao}, cannot exceed the saturation efficiency times the outlet humidity ratio at saturation, $\eta\omega_{aos}$:

$$\omega_{ao} = min(\ \omega_{ao}, \ \eta\omega_{aos})$$

TYPE 22 Component Configuration

Inputs   Description

1   T_s – steam temperature [C]

2   T_{ai} – inlet air temperature [C]

3   ω_{ai} – inlet air humidity ratio [–]

4   w_{ai} – inlet air mass flow rate [kg/s]

5   w_s – steam mass flow rate [kg/s]

Outputs   Description

1   T_{ao} – outlet air temperature [C]

2   w_{ao} – outlet air mass flow rate [kg/s]

3   ω_{ao} – outlet air humidity ratio [–]

Parameters

1   η – saturation efficiency

90

4.23 TYPE 23 : STEAM NOZZLE

General Description

The steam nozzle calculates a mass flow rate and exit steam temperature, given the downstream pressure and the upstream steam stagnation temperature and pressure. The model assumes an adiabatic, isentropic process, takes condensation into account and uses real gas correlations to determine the steam properties.

Nomenclature

A   - nozzle exit cross sectional area
H   - enthalpy
P   - pressure
S   - entropy
T   - temperature
V   - specific volume
w   - mass flow rate
x   - quality

Subscripts

e   - exit
f   - saturated liquid
g   - saturated vapor
o   - stagnation (upstream)
r   - reference
sat - saturation temperature

Mathematical Description

The input pressures P_o and P_e are first converted from gage pressures to absolute pressures by adding the second parameter, P_r, which represents the absolute pressure corresponding to zero gage pressure. The entropy, inlet enthalpy, saturation temperature, saturation entropy and outlet temperature are then determined by calling the steam property function subprograms described in chapter 3.

$$S = S_o = S_e = SS \ (\ P_o , T_o \ )$$

$$H_o = HS \ (\ P_o , T_o \ )$$

$$T_{sat} = TSATS \ (\ P_e \ )$$

$$S_g = SSATS \ (\ T_{sat} \ )$$

$$T_e = TPSS \ (\ P_e , S \ )$$

Condensation occurs if S_g is greater than S_o. If condensation does occur, then the concept of quality is used to determine the exit properties.

$$S_f = SSATW \ (\ T_{sat} \ )$$

$$V_f = VSATW \ (\ T_{sat} \ )$$

$$V_g = VSATS \ (\ P_e, T_{sat} \ )$$

$$H_f = HSATW \ (\ T_{sat} \ )$$

$$H_g = HSATS \ (\ T_{sat} \ )$$

$$x = \frac{S - S_f}{S_g - S_f}$$

$$V_e = V_f + x(\ V_g - V_f \ )$$

$$H_e = H_f + x(\ H_g - H_f \ )$$

If condensation does not occur, then the outlet properties are determined as follows :

$$V_e = VS \ (\ P_e, T_e \ )$$

$$H_e = HS \ (\ P_e, T_e \ )$$

The steam mass flow rate can now be calculated. The factor of 1000 in the following equation converts the enthalpy from kJ/kg to J/kg.

$$w = \frac{A}{V_e} \ [\ 2(\ H_o - H_e \ )1000 \ ]^{0.5}$$

TYPE 23 Component Configuration

Inputs		Description
1	T_o	– upstream stagnation temperature [C]
2	P_o	– upstream stagnation pressure [KPa, gage]
3	P_e	– exit pressure [KPa, gage]

Outputs		Description
1	T_e	– exit temperature [C]
2	w	– steam mass flow rate [kg/s]

Parameters

1	A	– nozzle exit cross sectional area [m^2]
2	P_r	– absolute pressure corresponding to zero gage pressure

4.24 TYPE 24 : IDEAL GAS NOZZLE

General Description

The Type 24 subroutine models a convergent nozzle for any ideal gas and assumes the nozzle to be isentropic, adiabatic and without condensation. The model calculates a mass flow rate and exit temperature given the upstream stagnation pressure and temperature and the downstream pressure. The model uses absolute pressures internally, but allows the input and output pressures to be gage pressures. The second parameter is the absolute pressure corresponding to zero gage pressure. The third and fourth parameters specify properties of the ideal gas.

Nomenclature

A — nozzle exit cross sectional area
P — pressure
T — temperature
w — mass flow rate
γ — specific heat ratio, Cp/Cv

Subscripts

c — critical
e — exit
o — stagnation

Mathematical Description

The two input pressures are first converted from gage to absolute pressures by addition of the second parameter, which represents the absolute pressure corresponding to zero gage pressure. The critical pressure ratio is then defined as:

$$P_c = C1^{-C2}$$

where $\quad C1 = \dfrac{2}{1 + \gamma} \quad$ and $\quad C2 = \dfrac{\gamma}{1 - \gamma}$

The method used to determine the flow rate depends on the critical pressure ratio and on another pressure ratio, Ra:

$$Ra = \frac{P_e}{P_o}$$

The maximum flow rate is obtained when the pressure ratio, Ra, is less than or equal to the critical pressure ratio:

$$w = 1000 \ A \ P_o \left[\frac{\gamma C1^{-C2 \cdot C3}}{RT_o} \right]^{.5}$$

93

where $C3 = \dfrac{1 + \gamma}{\gamma}$

Otherwise, the flow rate is defined by

$$w = 1000 \ A \ P_o \left[\frac{2C2(\ Ra^{C1 \cdot C3} - Ra^{C3})}{RT_o} \right]^{.5}$$

Lastly, the exit temperature is defined by the following equation:

$$T_e = T_o \ Ra^{-(1/C2)}$$

TYPE 24 Component Configuration

Inputs		Description
1	P_o	– upstream stagnation pressure [KPa, gage]
2	P_e	– downstream pressure [KPa, gage]
3	T_o	– stagnation temperature [C]

Outputs		Description
1	w	– mass flow rate [kg/s]
2	T_e	– exit temperature [C]

Parameters		Description
1	A	– nozzle exit cross sectional area [m^2]
2	P	– absolute pressure corresponding to zero gage pressure [KPa]
3	γ	– ratio of specific heats [–]
4	R	– gas constant per unit mass [J/(kg-K)]

4.25 TYPE 25: STEAM TO AIR HEAT EXCHANGER

General Description

This component represents a cross flow heat exchanger consisting of horizontal tubes with circular fins. Air flowing across the finned tubes is heated, while steam inside the tubes is condensed. The steam side is assumed to be equipped with a trap which allows condensate but not vapor to leave the tubes. No attempt is made to compute the condensate temperature at the exit. Instead, the condensate outlet temperature is assumed to be a fixed number of degrees below the saturation temperature. In addition, the average temperature of steam and condensate is assumed equal to the saturation temperature, and condensation is assumed to occur at constant pressure. The method used to represent air temperature dynamics assumes that the dynamics are due to changes in the steam pressure and thus in the saturation temperature, rather than to changes in the air inlet temperature or flow rate.

Inputs to the model are the steam pressure and inlet temperature, the air inlet temperature and flow rate, and the air outlet temperature, which is obtained from the first output of the model. Other outputs are the condensate outlet temperature, the mass flow rate of the steam (which is determined by the condensation rate), and the rate of heat transfer to the air.

Nomenclature

A_a — minimum air flow area (m^2)

A_i — tube inside surface area (m^2)

A_p — primary (tube) outside surface area (m^2)

A_s — secondary (fin) surface area (m^2)

C_m — thermal capacitance of heat exchanger (kJ/C)

C_{pa} — specific heat of air [kJ/(kg C)]

C_{pw} — specific heat of liquid water [kJ/(kg C)]

D_f — fin diameter (m)

D_i — tube inside diameter (m)

D_o — tube outside diameter (m)

F_m — number of fins per meter

g — acceleration of gravity (m/s^2)

h_i — inside surface heat transfer coefficient [kW/(m^2 C)]

h_o — outside surface heat transfer coefficient [kW/(m^2 C)]

H_{fg} — latent heat of vaporization of water (kJ/kg)

H_{si} — steam enthalpy at inlet (kJ/kg)

H_{wo} — condensate enthalpy at outlet (kJ/kg)

K_a — thermal conductivity of air [kW/(m C)]

K_f — fin thermal conductivity [kW/(m C)]

K_t — tube thermal conductivity [kW/(m C)]

K_w — thermal conductivity of liquid water [kW/(m C)]

N_{tu} — number of transfer units (—)

Nu — Nusselt number, based on tube outside diameter (—)

Pr — Prandtl number (—)

P_s — steam pressure (kPa, absolute)

Q — dynamic rate of heat transfer to air (kW)

Q_{ss} — steady state rate of heat transfer (kW)

Re — Reynolds number, based on tube outside diameter (—)

R_f — ratio of finned tube surface area to unfinned tube surface area

R_i – inside surface heat transfer resistance (C/kW)
R_o – outside surface heat transfer resistance (C/kW)
R_r – ratio of inside surface heat transfer resistance to total heat transfer resistance (–)
R_t – tube heat transfer resistance (C/kW)
T_{ai} – air inlet temperature (C)
T_{ao} – air outlet temperature (C)
T_{aos} – steady state air outlet temperature (C)
T_{sc} – difference between saturation temperature and condensate outlet temperature (C)
T_{ss} – saturation temperature of steam (C)
T_{si} – steam inlet temperature (C)
T_t – tube inside surface temperature (C)
T_{wo} – condensed water outlet temperature (C)
w_a – air mass flow rate (kg/s)
w_s – steam mass flow rate (kg/s)
δ – fin thickness (m)
η – fin efficiency (–)
μ_a – air viscosity [kg/(m s)]
μ_w – liquid water viscosity [kg/(m s)]
ρ_w – density of water (kg/m^3)
τ – heat exchanger time constant (s)

Mathematical Description

The model first determines the inlet steam enthalpy, the saturation temperature, and the outlet temperature and enthalpy, using property functions described in section 3.5:

$$H_{si} = HS(P_s, T_{si})$$

$$T_{ss} = TSATS(P_s)$$

$$T_{wo} = T_{ss} - T_{sc}$$

$$H_{wo} = HSATW(T_{ss}) - C_{pw}T_{sc}$$

Next the outside heat transfer coefficient is computed, using a correlation from Hausen [35] for air flow over finned tubes:

$$Nu = C\, Re^{0.625}\, R_f^{-0.375}\, Pr^{1/3}$$

In this equation, the constant C is 0.30 for in-line tube rows or 0.45 for staggered tube rows. A value of 0.30 is used in the present model. The Prandtl number, Pr, is evaluated at the bulk air temperature, using air property functions described in section 3.6. R_f, the ratio of outside surface area to the area of unfinned tubes with the same outside diameter, is calculated from the parameters. The characteristic length used in the Reynolds number, Re, and in the Nusselt number, Nu, is the tube outside diameter. Thus,

$$Re = \frac{w_a D_o}{\mu_a A_a}$$

and

$$h_o = \frac{Nu\ K_a}{D_o}$$

The fin efficiency, η, is determined by the use of subroutine SUFED described in section 3.3. The outside heat transfer resistance is then given by

$$R_o = [\ h_o(\eta A_s + A_p)\ ]^{-1}$$

The heat transfer resistance of the heat exchanger tubes is

$$R_t = \frac{(D_o - D_i)}{(2K_t A_i)}$$

The inside surface temperature of the pipe, T_s, and the inside heat transfer coefficient, h_i, are determined iteratively from the following equations:

$$h_i = 0.612\ \left[\ \frac{K_w^3 \rho_w^2 g H_{fg}}{\mu_w D_i (T_{ss} - T_t)}\ \right]^{0.25}$$

$$T_t = T_{ss} - R_r(T_{ss} - T_a)$$

where

$$R_r = \frac{R_i}{R_i + R_t + R_o}$$

$$R_i = (h_i A_i)^{-1}$$

and the correlation for h_i is from Kern [36].

The steady state rate of heat transfer is used to find the steady state air outlet temperature and the rate of condensation, which is equal to the entering steam mass flow rate:

$$Q_{ss} = h_i A_i(T_{ss} - T_t)$$

$$T_{aos} = T_{ai} + \frac{Q_{ss}}{(w_a C_{pa})}$$

$$w_s = \frac{Q_{ss}}{(H_{si} - H_{wo})}$$

Air temperature dynamics are computed by the approximate method of Myers et al. [37], assuming that the thermal capacitance of the heat exchanger walls is much greater than that of the air.

$$\frac{dT_{ao}}{dt} = \frac{T_{aos} - T_{ao}}{\tau}$$

where

$$\tau = \frac{C_m}{\alpha w_a C_{pa}}$$

$$\alpha = \frac{2 N_{tu} \exp(-z) \sinh(z)}{R_r\{[\sinh(z) + \cosh(z)]\exp(-z) - \exp(-N_{tu})\}}$$

$$z = \frac{N_{tu}}{2(1 - R_r)}$$

$$N_{tu} = [w_a C_{pa}(R_i + R_t + R_o)]^{-1}$$

Finally, the dynamic rate of heat transfer to the air is calculated:

$$Q = w_a C_{pa}(T_{ao} - T_{ai})$$

TYPE 25 Component Configuration

Inputs		Description
1	P_s	— steam pressure [KPa, absolute]
2	T_{si}	— inlet steam temperature [C]
3	T_{ai}	— inlet air temperature [C]
4	T_{ao}	— outlet air temperature [C]
5	w_a	— mass flow rate of air [kg/s]

Outputs		Description
1	T_{ao}	— outlet air temperature
2	T_{wo}	— outlet temperature of condensate [C]
3	w_s	— mass flow rate of steam [kg/s]
4	Q	— rate of heat transfer to air [kW]

Parameters	Description
1	T_{sc} – subcooling of condensate below saturation [C]
2	A_a – minimum air flow area [m^2]
3	A_i – inside heat transfer area [m^2]
4	A_p – primary (tube) outside surface area [m^2]
5	A_s – secondary (fin) surface area [m^2]
6	D_i – tube inside diameter [m]
7	D_o – tube outside diameter [m]
8	K_t – tube thermal conductivity [W/m K]
9	D_f – fin diameter [m]
10	K_f – fin thermal conductivity [W/m K]
11	δ – fin thickness [m]
12	F_m – number of fins per meter
13	C_m – thermal capacitance of metal (fins and tubes) [kJ/K]

4.26 TYPE 26: CONTROL SIGNAL INVERTER

General Description

This component is a very simple representation of an inverting relay. Its single input is a control signal between zero and one, and its output is one minus the input.

In a typical simulation, a single control variable might represent both the output of a controller and the relative position of a valve or damper. The TYPE 8 and TYPE 16 controller models operate on the assumption that increasing the output control signal will increase the controlled variable. Depending on the system, it may be necessary to use a TYPE 26 inverter between a controller and a valve or damper in order to maintain the validity of this assumption.

Nomenclature

C_i – input control signal
C_o – output control signal

Mathematical Description

$$C_o = 1. - C_i$$

$$0. \leq C_o \leq 1.$$

TYPE 26 Component Configuration

Input		Description
1	C_i	– input control signal

Output		Description
1	C_o	– output control signal

Parameters:

1   Multiplier

REFERENCES

1. May, W.B., and Clark, D.R. HVACSIM(+) Building Systems and Equipment Simulation Program: User's Manual. National Bureau of Standards, NBSIR, 1985.

2. Hill, C.R. "Simulation techniques for building systems." Proceedings of the Workshop on HVAC Controls Modeling and Simulation, Georgia Institute of Technology, February 1984.

3. Klein, S.A., et al. "TRNSYS: A Transient System Simulation Program." Report 38-11, Solar Energy Laboratory, University of Wisconsin - Madison, April 1981.

4. Hiebert, K.L. "SNSQ." Source code for subroutine SNSQ was obtained from the CMLIB software library, National Bureau of Standards, Gaithersburg, Maryland.

5. Powell, M.J.D. "A hybrid method for nonlinear equations." Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz (ed.), Gordon and Breach, 1970.

6. Chua, L.O., and Lin, P.M. Computer Aided Analysis of Electronic Circuits, Prentice-Hall, 1975.

7. Brayton, R.K., et al. "A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas." Proc. IEEE, Vol. 60, No. 1, Jan. 1972.

8. Elmahdy, A.H., and Mitalas, G.P. "FORTRAN IV program to simulate cooling and dehumidifying finned-tube multi-row heat exchangers." Computer Program No. 43, Division of Building Research, National Research Council of Canada, Ottawa (1977).

9. Gardner, K.A. "Efficiency of extended surfaces." Trans. ASME, Vol. 67, pp. 621-631 (1945).

10. Downing, R.C. "Refrigerant Equations." ASHRAE Transactions, Vol. 80, part 2, pp. 158-169 (1974).

11. Irvine, T.F. and Liley, P.E. Steam and Gas Tables with Computer Equations, Academic Press, Inc., 1984.

12. Van Wylen, G.J., and Sonntag, R.E. Fundamentals of Classical Thermodynamics. SI Version. New York: John Wiley and Sons, Inc. (1973).

13. Chapman, Alan J. Heat Transfer, 3rd edition. New York: Macmillan Publishing Co., Inc. (1974).

14. Chi, J. "Building Environmental Control System (BECS) Analysis Computer Program, Volume 2: Component Subroutines." HCP Systems, Inc., final report to the National Bureau of Standards under Contract No. NB82SBCA1577, June 1983.

15. Karlekar, B.V., and Desmond, R.M. _Engineering Heat Transfer._ St. Paul: West Publishing Company (1977).

16. _CRC Handbook of Chemistry and Physics_, 61st Edition (1980-1981).

17. Clark, D.R., Hill, C.R., and Hurley, C.W., "Dynamic models for HVAC system components." _ASHRAE Transactions,_ Vol. 91 (1985).

18. Tobias, J.R., "Simplified transfer function for temperature response of fluids flowing through coils, pipes or ducts." _ASHRAE Transactions,_ Vol. 79, pp. 19-22, 1973.

19. Croome, J.D. and Roberts, B.M., _Airconditioning and Ventilation of Buildings._ Pergamon Press Inc., p. 310, 1981.

20. Haines, Roger W., _Control Systems for Heating, Ventilating and Air Conditioning._ Van Nostrand Reinhold Company Inc., pp. 67-68, 1983.

21. McQuiston, F.C., and Parker, J.D. _Heating, Ventilating, and Air Conditioning Analysis and Design,_ 2nd edition. New York: John Wiley and Sons, Inc. (1982).

22. Pearson, J.T., Leonard, R.G., and McCutchan, R.D. "Gain and time constant for finned serpentine crossflow heat exchangers." _ASHRAE Transactions_ Vol. 80, part 2, pp. 255-267 (1974).

23. Gartner, J.R., and Harrison, H.L. "Dynamic characteristics of water-to-air cross-flow heat exchangers." _ASHRAE Transactions_ Vol. 71, part 1 (1965).

24. Bhargava, S.C., McQuiston, F.C., and Zirkle, L.D. "Transfer functions for crossflow multirow heat exchangers." _ASHRAE Transactions_ Vol. 81, part 2, pp. 294-314 (1975).

25. Elmahdy, A.H., and Mitalas, G.P. "A simple model for cooling and dehumidifying coils for use in calculating energy requirements for buildings." _ASHRAE Transactions_ Vol. 83, part 2, pp. 103-117 (1977).

26. McAdams, W.H. "Review and summary of developments in heat transfer by conduction and convection." _Trans. A.I.Ch.E.,_ Vol. 36, p.1 (1940).

27. Elmahdy, A.H., and Biggs, R.C. "Finned tube heat exchanger: correlation of dry surface heat transfer data." _ASHRAE Transactions_ Vol. 85, part 2, pp. 262-273 (1979).

28. Elmahdy, A.H. "Analytical and experimental multi-row finned tube heat exchanger performance during cooling and dehumidifying processes." Ph.D. Thesis, Carleton University, Ottawa, Canada (1975).

29. Elmahdy, A.H., and Biggs, R.C. "Efficiency of extended surfaces with simultaneous heat and mass transfer." _ASHRAE Transactions,_ Vol. 89, part 1 (1983).

30. Elmahdy, A.H., and Mitalas, G.P. "Facility for testing cooling and dehumidifying coils: description, procedures and test results." DBR Paper No. 756, Division of Building Research, National Research Council of Canada, Ottawa (1978).

31. Elmahdy, A.H., and Biggs, R.C. "Performance simulation of multi-row dry (and/or wet) heat exchangers." DBR Paper No. 797, Division of Building Research, National Research Council of Canada, Ottawa (1978).

32. Chi, J. "Building Environmental Control System Analysis (BECSA) Computer Program, Volume 1: Component Mathematical Models." HCP Systems, Inc., final report to the National Bureau of Standards under Contract No. NB82SBCA1577, Sept. 1982.

33. Hill, C.R. "Simulation of a multizone air handler." ASHRAE Transactions, Vol. 91 (1985).

34. Borresen, B.A. "Thermal room models for control analysis." ASHRAE Transactions, Vol. 87, part 2, 1981.

35. Hausen, H. Heat Transfer in Counterflow, Parallel Flow and Cross Flow. New York: McGraw-Hill Book Company, 1983.

36. Kern, D.Q. Process Heat Transfer. New York: McGraw-Hill Book Company, 1950.

37. Myers, G.E., Mitchell, J.W., and Lindeman, C.F. Jr. "The transient response of heat exchangers having an infinite capacitance rate fluid." Transactions of the ASME: Journal of Heat Transfer, pp. 269-275, May 1970.

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA** **SHEET** *(See instructions)*	1. PUBLICATION OR REPORT NO. NBSIR 84-2996	2. Performing Organ. Report No.	3. Publication Date January 1985

4. TITLE AND SUBTITLE

HVACSIM(+) Building Systems and Equipment Simulation Program Reference Manual

5. AUTHOR(S)

Daniel R. Clark

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)* **NATIONAL BUREAU OF STANDARDS** **DEPARTMENT OF COMMERCE** **WASHINGTON, D.C. 20234**	7. Contract/Grant No.
	8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Office of Buildings & Community Systems   U.S. Naval Civil Engineering Laboratory
U.S. Department of Energy   U.S. Department of Defense
1000 Independence Ave., NW   Port Hueneme, CA 93043
Washington, DC 20585

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

HVACSIM(+) is a modular, non-proprietary computer simulation package developed at the National Bureau of Standards, designed to allow detailed simulation of entire building energy systems: the heating, ventilating, and air conditioning (HVAC) system, the equipment control system, the building shell, the physical plant, and the dynamic interactions among these subsystems.

The HVACSIM(+) package consists of a main simulation program, a library of subroutines containing mathematical models of building energy system components, and two programs used in preparing a description of the system to be simulated. Models representing the components of a physical plant, such as boilers and chillers, and a model representing a multizone building, are under development and will be added to the HVACSIM(+) package as they become available.

This document describes the operation, capabilities, and limitations of the main simulation program, presents descriptions of the existing library of component models for simulating HVAC systems and their controls, includes theoretical or experimental validation of the models wherever possible, and explains the procedure by which new component models may be written and added to the library.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

building dynamics; building simulation; building system modeling; computer simulation techniques; control dynamics; dynamic modeling of building systems; dynamic performance of building system; dynamic simulation; HVAC system simulation; HVACSIM+

13. AVAILABILITY

☒ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

110

15. Price

$13.00