A QUASI-DYNAMIC HVAC AND BUILDING SIMULATION METHODOLOGY

A Thesis

by

CLINTON PAUL DAVIS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2012

Major Subject: Electrical Engineering

A QUASI-DYNAMIC HVAC AND BUILDING SIMULATION METHODOLOGY

A Thesis

by

CLINTON PAUL DAVIS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee, Charles Culp
Prasad Enjeti
Committee Members, Don Russell
Pierce Cantrell
Head of Department, Costas Georghiades

May 2012

Major Subject: Electrical Engineering

ABSTRACT

A Quasi-Dynamic HVAC and Building Simulation Methodology. (May 2012)

Clinton Paul Davis, B.S., Texas A&M University

Co-Chairs of Advisory Committee: Dr. Charles Culp
Dr. Prasad Enjeti

This thesis introduces a quasi-dynamic building simulation methodology which complements existing building simulators by allowing transient models of HVAC (heating, ventilating and air-conditioning) systems to be created in an analogous way to their design and simulated in a computationally efficient manner. The methodology represents a system as interconnected, object-oriented sub-models known as components. Fluids and their local properties are modeled using discrete, incompressible objects known as packets. System wide pressure and flow rates are modeled similar to electrical circuit models. Transferring packets between components emulates fluid flow, while the system wide fluid circuit formed by the components' interconnections determines system wide pressures and flow rates.

A tool named PAQS, after the PAacketized Quasi-dynamic Simulation methodology, was built to demonstrate the described methodology. Validation tests of PAQS found that its steady state energy use predictions differed less than 3% from a comparable steady state model. PAQS was also able to correctly model the transient behavior of a dynamic linear analytical system.

To my mother Melanie.

ACKNOWLEDGEMENTS

I would like to acknowledge Dr. Culp for all of his help, guidance and insight. I would also like to thank Christina Chea for her help with editing. Finally, I would like to thank my family for their support over the years.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Page

LIST OF TABLES

1. INTRODUCTION

1.1 Background and Problem Statement

Designing successful controllers for HVAC systems that positively impact building energy consumption and indoor air quality depends on the availability of dynamic models that describe their key behaviors. Complexities in HVAC systems, such as distributed parameters and nonlinearities, make it difficult to obtain exact mathematical models (Tashtoush et al., 2005).

The complexities of coupled interactions in buildings require computer simulations to predict energy consumption and system performance (CERL, 1999). Programs such as SPARK (LBNL, 2003), TRNSYS (Klein et al., 1976), and HVACSIM$^+$ (Park et al., 1985) perform dynamic simulations of buildings using a modular equation solver approach. Modules generally consist of linear or nonlinear algebraic or differential equations with the equation input and output variables serving as the module's inputs and outputs. A module's connections with other modules determine the overall system design. When performing a simulation, a computational equation solver simplifies a system's equations and determines a solution scheme.

While these methodologies can accurately model a building's dynamic performance, improvements can be made. Iterative calculations involved in nonlinear equation solving take up considerable computational effort. Eliminating or

---

This thesis follows the style of *ASHRAE Transactions.*

reducing these iterations would allow simulations to be performed more quickly. HVAC designers also view systems as linked objects that carry fluid and control signals, not as coupled sets of equations. Presenting simulations in this manner would make them more intuitive to industrial users. Overall, developing a dynamic HVAC simulation tool that focuses on computational efficiency and usability would benefit HVAC control system designers and engineers to better understand the dynamics that are involved with their systems.

1.2 Objectives

The concept of a new HVAC and building simulation methodology was developed and prototyped as part of the research involved with this thesis. This methodology simulates dynamic behavior over time using a series of steady state simulations. Each steady state simulation covers a discrete period of time. The methodology's quasi-dynamic nature varies operating conditions between time steps while modeling a system as steady state on a particular time step.

The design of this methodology focuses on three major areas. First, its structure must be intuitive so those people familiar with HVAC control system design can create and simulate dynamic HVAC system models. Second, coupling between system components needs to be defined on the user's level so that novel HVAC system models can be created from a library of standard components. Finally, it must be computationally efficient so that it is practical for use in the work flow of HVAC control system designers.

After developing the simulation methodology, a prototype graphical user interface and component library were designed and implemented in order to test the methodology and demonstrate its flexibility. The performances of overall simulations were compared against previously validated steady state and analytical dynamic models, to ensure that the simulations worked as designed. Other tests were also performed to ensure that components could be linked in general configurations and simulated correctly.

2. LITERATURE REVIEW

2.1 HVAC Control Methodologies

2.1.1 Complete HVAC Systems

HVAC systems typically contain multiple interacting control loops that perform several functions, such as maintaining the water level in a boiler, controlling the output water temperature of a chiller, and throttling a fan. As a result of the numerous system configurations available, the method of control used in a particular control system depends heavily on the engineers involved in the design (Levenhagen, 1999).

2.1.2 Feedback Control Methods

Current control methods include analog electromechanical (pneumatic), analog electronic, and digital. Pneumatic HVAC controls use compressed air to perform the control logic and mechanical actuation of HVAC systems. Sensors output a pressure differential, and fluid logic circuits control pneumatic actuators (Wilson et al., 1965). Analog electronic control systems operate similarly to pneumatic systems using analog electronic sensors, analog computers, and electromechanical actuators (Gupton, 2002). Direct digital control, or DDC, makes all control decisions and performs all communication with digital circuits (Newman, 1994).

2.1.3 HVAC Control Algorithms

HVAC systems make extensive use of two term proportional-integral, or PI controllers, and occasionally use three term proportional-integral-derivative, or PID, controllers. The output of these types of controllers consists of the weighted sum of the input error, the time integral of the input error, and the time derivative of the input error. Terms that stabilize a system may be set using empirical methods such as the Ziegler-Nichols method or by individual hand tuning (Letherman, 1981).

Self-tuning controllers in HVAC systems come in two types: autotuning and adaptive. Autotuning software automates the controller tuning procedure by exciting a response from a control loop, and calculating the control parameters from this response. Adaptive controls change their control strategy based on a well-known, but slowly varying operating condition (CIBSE, 2000).

2.2 HVAC Control System Design

In the current literature regarding application of modern control techniques to HVAC, most use state space analytical representations of HVAC systems for control system synthesis and analyses. For example, Argüello-Serrano and Vélez-Reyes (1995) developed a state observer for estimating a building's thermal loads. Also, Moshiri and Rashidi (2004) created an adaptive proportional-integral-derivative controller for HVAC systems using fuzzy logic.

State space HVAC system models allow control system designers to analytically analyze the effects of different changes in a system. However, they neglect many parameters. To test and validate control system designs made using a simplified model, Anderson et al. (2007) constructed a physical HVAC system. The dynamic simulation tool Simulink (MathWorks, 2011) was used to make all feedback control decisions that allowed them to configure and test new control methods entirely within software. Hepworth and Dexter (1994) tested and validated their control system using an actual building. They combined a traditional proportional-integral controller with a neural network for modeling nonlinear HVAC plant characteristics.

2.3 Building Simulation Methodologies and Programs

2.3.1 Noncomputerized Building Simulations

Degree-day (DD) methods assume that a building's daily heating or cooling usage varies proportionally to the difference between the daily mean outdoor temperature and a certain balance point temperature, typically 65°F (Knebel, 1983). Although DD methods provide rough estimates of a building's energy usage, they lack the capability to perform a detailed analysis of a building's physics (ASHRAE, 2001).

Bin methods, such as the Modified Bin method (Knebel, 1983), perform steady state building simulations under a variety of weather conditions, and estimate energy usage for a given climate based on how often these weather conditions occur. They can account for individual system characteristics such as the part load performance of HVAC equipment and heat pump systems.

2.3.2 Whole Building Energy Analysis

In 1976, a collaboration of several national laboratories resulted in a building energy analysis program known as DOE-2 (Birdsall et al., 1990). DOE-2 consists of several subprograms that calculate different aspects of a simulation such as loads, HVAC system performance, plant performance, and costs. It simulates an entire year on an hourly basis, and includes dynamic building behavior that occurs on hourly time scales (York and Cappiello, 1981).

In one analysis of DOE-2's accuracy, the results came within 10% to 26% of measured data values, within 1% to 30% of other energy simulation programs, and within 5% of analytical calculations (Haberl and Cho, 2004). In another analysis using DOE-2, discrepancies of over 50% were found between simulated and measured data values. Differences in this case were found to be caused by the uncertainties inherent in building energy simulation (Ahmad and Culp, 2006).

Like DOE-2, BLAST consists of multiple programs that simulate a building on an hourly basis (CERL, 1999). However, BLAST uses a more advanced method based on calculating zone loads (Strand et al., 2001). More computational effort is utilized for a more physically accurate calculation (Strand et al., 1999).

EnergyPlus, released in 2001, serves as the successor to BLAST and DOE-2 by integrating their most popular features and capabilities. It uses a modular architecture that facilitates adding new features and links to other programs (Crawley et al., 1999). It integrates the calculation methodologies of previous programs, such as the thermal load calculations from a research version of BLAST, the daylighting calculation method of DOE-2, the window modeling methods of WINDOW, and the interzonal airflow calculation methods of COMIS (Strand et al., 1999). EnergyPlus structures simulations around components such as pipes, ducts, and chillers placed on abstract representations of duct or pipe systems. During a simulation, all aspects of a building are simulated at once, running an iterative algorithm until the entire system has converged on each time step (Fischer et al., 1999).

2.3.3 Equation Based Building Simulations

SPARK, (LBNL, 2003), represents buildings as interconnected modules containing nonlinear algebraic and differential equations. During a simulation, it solves an entire decomposed equation set as a whole (Buhl et al., 1993). Libraries of components allow it to perform different tasks, such as hourly steady state energy analyses and automated fault detection and diagnostics (Sowell and Moshier, 1995).

TRNSYS, (McDowell et al., 2004), represents systems in the same way as SPARK. However, TRNSYS allows "black box" modules based on empirical data tables. Like SPARK, TRNSYS automatically generates calculation procedures for a system (Klein et al., 1976). HVACSIM^{+} (Park et al., 1985) performs dynamic simulations of buildings and HVAC systems using a hierarchical, modular approach based on TRNSYS.

2.3.4 Interzonal Airflow Analysis

COMIS, (Feustel, 1999), performs quasi-steady state pressure, airflow, and pollutant transport simulations of buildings. It represents buildings as connected modules that represent physical components such as zones, ducts, openings between zones, and fans. Users provide schedules that determine fan speeds and other system inputs for a simulation.

CONTAM, (Walton and Dols, 2005), performs the same type of airflow modeling as COMIS. However, it uses a transient airflow model whenever non-flow processes like humidity removal exist. For modeling 1D convection at high air flow rates, CONTAM divides the air within a duct or zone into constant volume cells.

Adding and removing cells from opposite ends of a duct or zone on each time step models 1D convection exactly.

2.3.5 Computational Fluid Dynamics

Computational fluid dynamics, or CFD, numerically solves the Navier-Stokes equations for fluid flow. Models that approximate turbulent flow must be used under turbulent flow conditions. Using CFD in HVAC requires an in-depth knowledge of fluid flow and the approximations used in the CFD modeling (Chen, 1997), due to the complexities of the numerical algorithms.

2.3.6 Coupling Different Building Analysis Methodologies

Different methodologies used in the analysis of buildings can be coupled together by (1) expanding the capabilities of existing software or by (2) facilitating communication between software tools (Djunaedy et al., 2005). Coupling building simulation tools that focus on different domains, such as fluid flow and heat transfer, combines the sophisticated methods used in each tool. Expanding current software eliminates the need for communication between two different programs, but may require rewriting major portions of a program's source code.

2.4 Fluid Network Modeling

Fluid networks, such as pipe and duct systems in HVAC, model fluid flow as a circuit with fluid traveling between sets of nodes. This section reviews fluid network modeling methodologies outside the area of HVAC that can be applied to it.

2.4.1 Blood Flow Simulation

Other areas of science develop fluid network models in ways that apply to other fields. To model the human circulatory system, Migliavacca et al. and Westerhof et al. (2000; 1968) used passive electrical components as analogies to blood flow and blood vessel behavior. This method derives from the Navier-Stokes equations and Hooke's law. By comparing to electrical analogies, voltage serves as an analog to pressure, current serves as an analog to flow, resistors serve as an analog to arterial flow resistance, inductors serve as an analog to fluid inertia, and capacitors serve as analogs to arterial elasticity.

2.4.2 Municipal Water Distribution

The numerous branches and segments found in HVAC ductwork and pipework resemble the structures found in municipal water distribution systems. In such networks, nonlinear equations relate the flows through each network element, and the head drop between the network nodes. When treated as steady state systems, the Newton-Raphson method can be used to calculate all head losses and flow rates throughout the network for a single instant in time. Linearizations of these equations can be used in a dynamic simulation, provided that changes in operating conditions remain small (Coulbeck, 1980).

Filion and Karney (2003) found that simulations of water distribution networks have many potential sources of error. Generating a system model from measured data has the greatest error potential, due to the difficulty of testing and modeling each section of a network. Other fundamental sources of error exist as well. For instance, the estimates of the Darcy-Weisbach and Hazen-Williams friction models become unreliable under transient conditions.

2.5 Modeling HVAC Components

HVAC system models combine the models of various subcomponents to create an entire system. This section reviews the modeling methodologies currently used on these components. This ensures that any system-wide methodologies developed permit accurate models of all parts of a system.

2.5.1 Boilers and Chillers

While boilers often have complex internal controls, a single value representing their overall efficiency can represent the effects numerous model-specific parameters. Coil models can be used to simulate their dynamic performance (Garcia-Borras, 1983). Creating dynamic models of vapor compression chillers requires modeling all internal components and control algorithms in order to model effects such as start-up, feedback control, and shut-down (Bendapudi et al., 2005).

2.5.2 Cooling and Heating Coils

Three main types of dynamic coil models exist: lumped capacitance models that use transfer functions to model heat conduction, dynamic models considering spatial variation, and empirical models based on heat exchanger effectiveness (Yu et al., 2005). In order to develop an optimized control for cooling coils, Wang et al. (2004) developed a simplified steady state model with empirically chosen coefficients. Also, Xu et al. (2006) developed a dynamic cooling coil model for use in SPARK that divides a coil into twenty independent and discrete tube sections. This allows the modeling of partially wet coil conditions.

2.5.3 Cooling Towers

Cooling tower analysis applies energy and mass balance equations to the evaporation, heat flow, and mass flow that occurs within them. Commonly used analysis methods make several simplifying assumptions, such as neglecting heat transfer within the circulating water (Webb, 1984).

2.5.4 Dampers and Valves

One dimensional (fluid circuit) damper and valve flow modes use empirically derived function of a device's stroke position and the ratio of its pressure drop versus the system's total pressure drop. Manufacturers typically publish graphs of these flow characteristics. Their actuation method limits their dynamic response (Ward-Smith, 1980).

2.5.5 Ducts and Pipes

Air in ducts loses energy through heat and air losses. As conditioned air passes through them, it can lose 10% to 40% of its cooling capacity and 0.5°C to 6°C through heat losses with the surrounding air (Fisk et al., 2000). R-values can be used to quantify the magnitude of such losses (Griffiths and Zuluaga, 2004). Duct air leakage typically accounts for approximately 25% of the flow through fans. In commercial duct systems, most of this leakage typically occurs at joints versus the seams. Leakage rates typically vary as an exponential function of the static pressure difference at the duct's surface (Aydin and Ozerdem, 2006).

Compared to ducts, pipes typically lose more energy through radiation. Also, pressure losses due to valves and fittings usually exceed that of straight runs. Pipes can also experience water hammer when water abruptly stops flowing. Over time, cavitation and other factors cause pipe system corrosion, changing the relationships between pressure and flow (Vedavarz et al., 2007).

Modeling flows in pipes and ducts in multiple dimensions account for fluid changes along the cross section of a flow (Ward-Smith, 1980). In a one-dimensional analysis, empirical formulas relate system pressure losses to its physical properties, internal fluids, and internal volume flow rate under ideal flow conditions (Vedavarz et al., 2007). One-dimensional analysis of pipe or duct networks can use numerical methods such as the Newton-Raphson method to find all nodal pressures and branch volume flow rates (Majumdar et al., 1998). To represent axial temperature variations along a duct or pipe branch, dynamic HVAC simulation programs typically use polynomials, fit to the axial temperature variation or tables that give temperatures along the length of the flow (Clark et al., 1985).

2.5.6 Fans and Pumps

All fans and pumps that operate in closed fluid loops obey relations known as affinity laws. These laws relate volume flow rate, rotational speed, head, fluid density, and power usage (Ward-Smith, 1980). However, they are only applicable under ideal no-loss operating conditions. For instance, nonideal inlet or outlet conditions affect the operation of fans and pumps (Vedavarz et al., 2007). In addition, pumps decrease their performance due to clogged or broken hardware, and gasses trapped in pipes (ASHRAE, 1996).

2.5.7 Sensors

Modeling sensor performance requires several factors to be considered, including the range, sensitivity, linearity, response time, accuracy, repeatability, and long-term stability (Newman, 1994).

2.5.8 Zones

Overall, three types of zone models exist. Well-mixed models assume that homogeneous air exists within a zone; zonal models divide zones into several well-mixed subvolume; CFD models analyze the airflow within zones in detail (Riederer et al., 2002). All zone models must take into account the exterior heat conduction, occupancy, lighting, appliances, solar radiation, and air leakage occurring within zones (Langley, 2000). R-values specify a steady state thermal conductance, while transient models take thermal mass into account. Performing solar load calculations requires extensive information about the orientation and properties of buildings. Use of a calculated value, known as a sol-air temperature (ASHRAE, 2001), simplifies solar load calculations. Infiltration and exfiltration from outside air causes both sensible and latent heat loss by replacing as much as 10% to 60% of the air within a zone with outside air. Building simulation programs typically model the rate of infiltration through a crack as a nonlinear function of the crack's differential pressure, times proportionality constant. Internal heat gains from people, plug loads, and lighting make a significant contribution to a zone's cooling load. These heat gains vary with time and cannot be easily measured

(Claridge et al., 2003). Building simulators typically handle this variation and uncertainty by classifying each day of a simulation as a particular daytype, such as weekday or weekend, and using a separate heat gain schedule for each day type.

2.6 Simulation Methodology Validation

Validation of a building simulation methodology establishes the accuracy of the numerical solutions produced, and the range in which the model has validity. Three well-established methods of validation exist: analytical verification, intermodel comparisons, and empirical validation (Irving, 1988). Of all validation methods, empirical validation provides the most direct measure of a simulation's authenticity.

3. HIGH LEVEL ALGORITHM DESIGN

The quasi-dynamic building simulation methodology emulates the dynamic response of buildings by dividing the time response of a building into fixed-length time steps and treating each time step as a steady state calculation. This methodology was built into a program and named PAQS, after the concept of packets it uses to model fluids within a system. This chapter introduces the high level design of PAQS and its calculation algorithm.

3.1 Program Division

A PAQS simulation consists of two main parts: (1) modular system models known as components, and (2) a system-wide control module known as the simulation manager. The components implement the physical behaviors of a system while the simulation manager coordinates components so that they interact correctly to produce a valid simulation.

Figure 3-1 illustrates the layers of abstraction between a user and a PAQS simulation. Users interact with a user interface that sets up, runs, and displays the outputs of a simulation. Within PAQS, the simulation manager coordinates the actions performed by components that model the physical system itself.

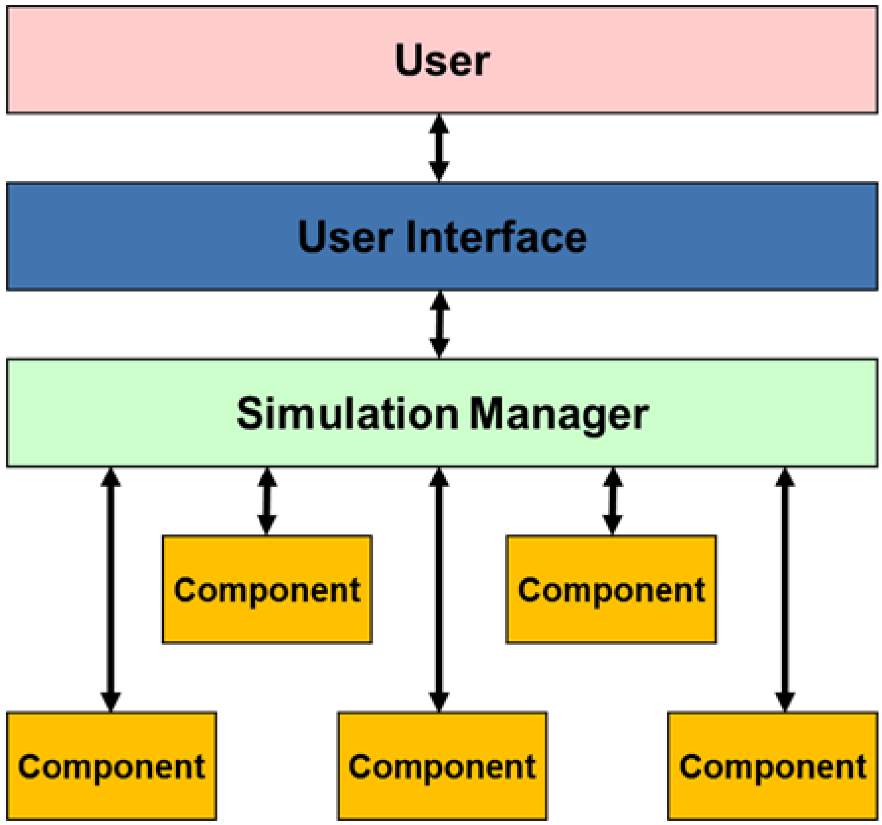

Figure 3-1 – The Abstraction Layers of PAQS

3.1.1 Components

Components model all physical equipment and control logic in PAQS. Internally, components store the state of their modeled equipment at a single instant in time. The simulation manager performs a simulation by instructing components to perform various actions that change their internal conditions over time.

For example, suppose component representing a room has two states that vary over time: (1) its internal temperature, and (2) its internal relative humidity. The room component itself contains these two states along with the logic of how to change its conditions when adding or removing heat or humidity. However, the simulation manager controls when these actions would occur.

A set of components and the connections between them define a modeled system. Fundamental HVAC components such as ducts, dampers, or control valves can be used to control the structure of as a system at a low level. Likewise, coarser components that incorporate entire air handling units (AHUs) or subsystems into a single component can be used to reduce the design complexity of a system.

Figure 3-2 shows a simple HVAC system. This system consists of a single room with a fan that blows air over a heating and cooling coil. Figure 3-3 shows one way of representing this system using components. In this figure, solid lines represent airflow connections while dotted lines represent control system data flow. In addition to the physical components shown in Figure 3-2, Figure 3-3 contains two components that only contain control logic: (1) a PID controller for the fan and (2) a controller for the dampers that regulate outside airflow.

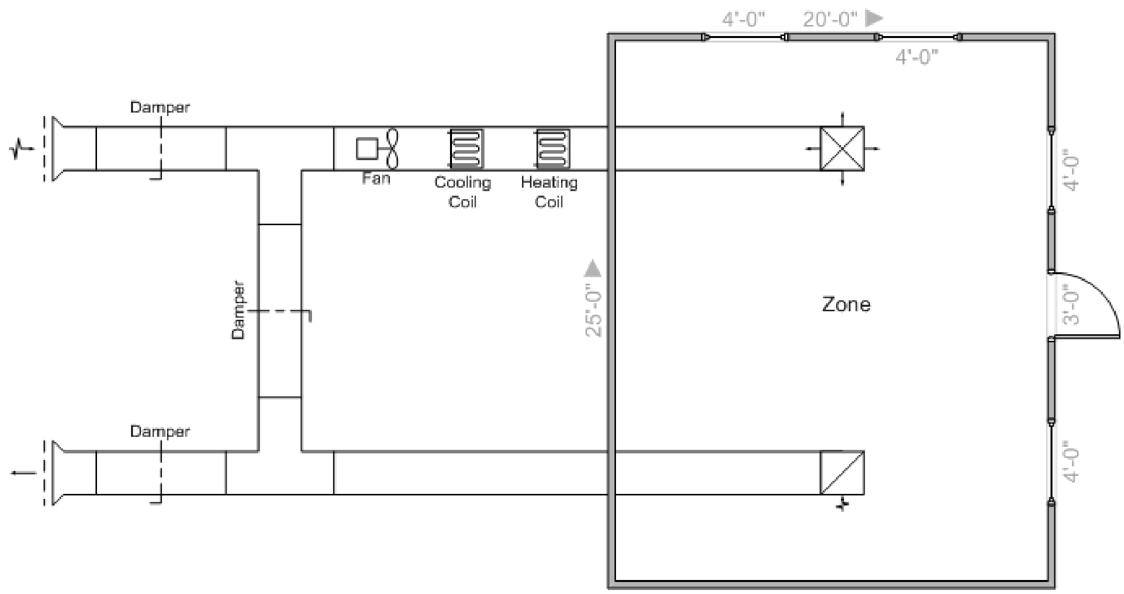

Figure 3-2 – A Simple HVAC System

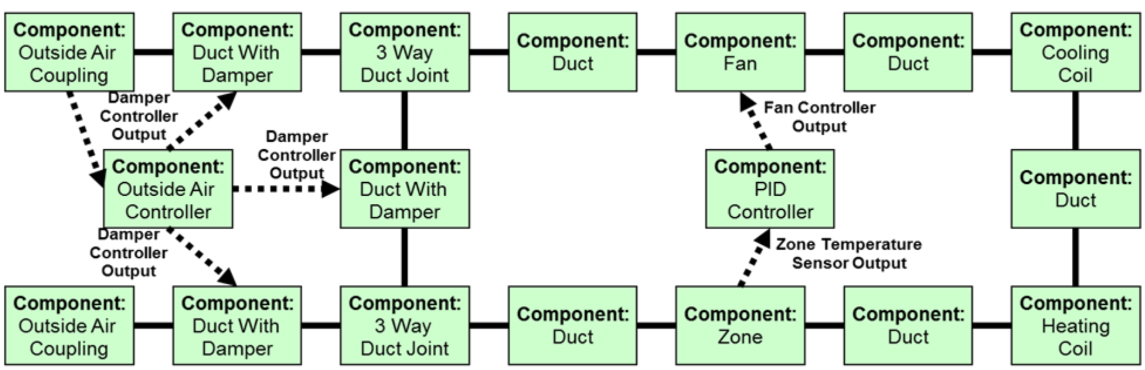

Figure 3-3 – Component Interconnections of a Simple HVAC System

3.1.2 The Simulation Manager

The simulation manager controls the overall execution of a PAQS simulation. It ensures that components modeling individual pieces of a system interact with each other to produce a correct system-wide simulation. It also acts as an interface to the user for setting up, running, and viewing the results of a simulation.

Data wise, the simulation manager stores five items: (1) components used in a simulation, (2) the interconnections between components, (3) time-varying input data such as weather data, (4) time-varying output data, and (5) the current date and time of an ongoing simulation. The simulation manager uses and manipulates this data using the main simulation algorithm discussed in section 3.2.3. This algorithm performs system-wide pressure and flow calculations and other tasks needed to ensure a valid simulation.

3.2 Simulations

3.2.1 Time Step Periods

A simulation in PAQS covers a continuous time interval between user defined start and stop dates. PAQS divides time into discrete time steps during a simulation. Users define a simulation's time step period, and it remains constant throughout a simulation.

Ideal time step periods vary between different systems but should be below the response time of any high frequency dynamics. When performance is a factor, a longer time step period can be selected that produces the same results as a small time step period while still avoiding high frequency errors. For example, a building with a temperature response time constant of fifteen minutes can start with a time step period of one second. This can be increased until an ideal tradeoff between the time required to run a simulation and the potential of incorrectly modeling high frequency dynamics is reached.

3.2.2 Makeup of a Simulation

Adding components, simulation timing and duration information, input data, and output data collection settings to the simulation manager prepares the system to be simulated. Once initialized, a command given to the simulation manager begins the process. After execution, the simulation manager writes all output data to an Excel file for presentation to the user.

3.2.3 The Main Simulation Algorithm

The main simulation algorithm, or MSA, simulates a system by controlling each component's internal operations and external interactions in a specific way. Figure 3-4 contains pseudocode for the main simulation algorithm. Overall, the MSA consists of two phases, known as settle and progress. In the settle phase, a component reacts to new inputs or physical conditions. The progress phase moves fluid between components.

The MSA runs the settle phase once at the beginning of a simulation to initialize each of the components. After this, the MSA iterates through a loop that changes the system’s state over time. Each iteration of this loop runs the progress phase to move fluid around the system and then the settle phase to update sensor outputs to their new values.

```
FUNCTION MainSimulationAlgorithm()
{
	FOR timeStep = 1 to numberOfTimeSteps
	{
		Settle()
		Progress()
	}
	Settle()
}
```

Figure 3-4 - Pseudocode for the Main Simulation Algorithm

Figure 3-5 contains the pseudocode for the settle phase of the main simulation algorithm. In the first step of this phase, the simulation manager sends current time varying input data, such as weather data, to the components. Then, the simulation manager calculates the system wide pressure and flow rates. Next, each component object calculates outputs used for control logic. Finally, the simulation manager logs all output data.

```
FUNCTION Settle()
{
	UpdateInputsForTimeStep()
	CalculateCurrentPressureAndFlowRates()
	CalculateControlOutputs()
	LogOutputData()
}
```

Figure 3-5 – Pseudocode for the Settle Phase of the Main Simulation Algorithm

Figure 3-6 contains the pseudocode for the progress phase of the main simulation algorithm. The progress phase involves the movement of fluid and thermal energy between components. In the first step of this phase, components model any internal transient responses. For instance, a PID controller updates its integral and derivative values at this step. In the second step of this phase, the simulation manager controls the transfer of discrete packets of air and water between components. This emulates fluid flow for a system.

```
FUNCTION Progress()
{
    CalculateInternalDynamicResponses()
    MoveFluidBetweenComponents()
}
```

Figure 3-6 – Pseudocode for the Progress Phase of the Main Simulation Algorithm

4. LOW LEVEL ALGORITHM DESIGN

This chapter presents the HVAC system modeling, fluid modeling, and the calculation of system pressures and fluid volume flow rates used in PAQS. Combined with the overall framework presented in Chapter 3, these algorithms enable PAQS to perform full quasi-dynamic building simulations.

4.1 Component Outputs and Inputs

A component's external interface allows double precision floating point data to be read from and shared. Two mechanisms allow this to happen: 1) a component's data value can be shared through a commonly referenced memory location, allowing multiple components to access the same data while only requiring a single write operation to change it; 2) a component can be queried by the simulation manager for a particular value, such as its current flow rate or pressure drop.

4.1.1 Time Varying Inputs

Simulations often contain input data that changes with time. Weather data and building conditions that vary by date fall into this category. In PAQS, the simulation manager maintains the current value for components to use for this type of data. Components reference the memory location that the simulation manager writes to when using the data.

4.1.2 Component Outputs

Components have an interface that allows them to output double precision floating point numbers. The interface facilitates output data collection and control output reading from components. The simulation manager passes a parameter to a component that indicates what value should be returned. This value can be an internal parameter, such as a duct’s static pressure or a zone’s temperature. Outputs can also be more general, such as the temperature at a given number of feet in a duct. While a component in a simulation has a fixed number of output types, all outputs used for data collection can also be used as control outputs.

4.1.3 Control Inputs and Outputs

When instructed by the simulation manager, components can pass floating point values from one another in the form of control signals. A control connection updates itself once on each time step through interfaces known as control input ports and control output ports. Figure 4-1 shows a simple component loop with a PID controller operating the fan. The zone has a control output port that sends its temperature to the PID controller's control input port. The PID controller also has a control output port that sends its output to the fan's control input port.

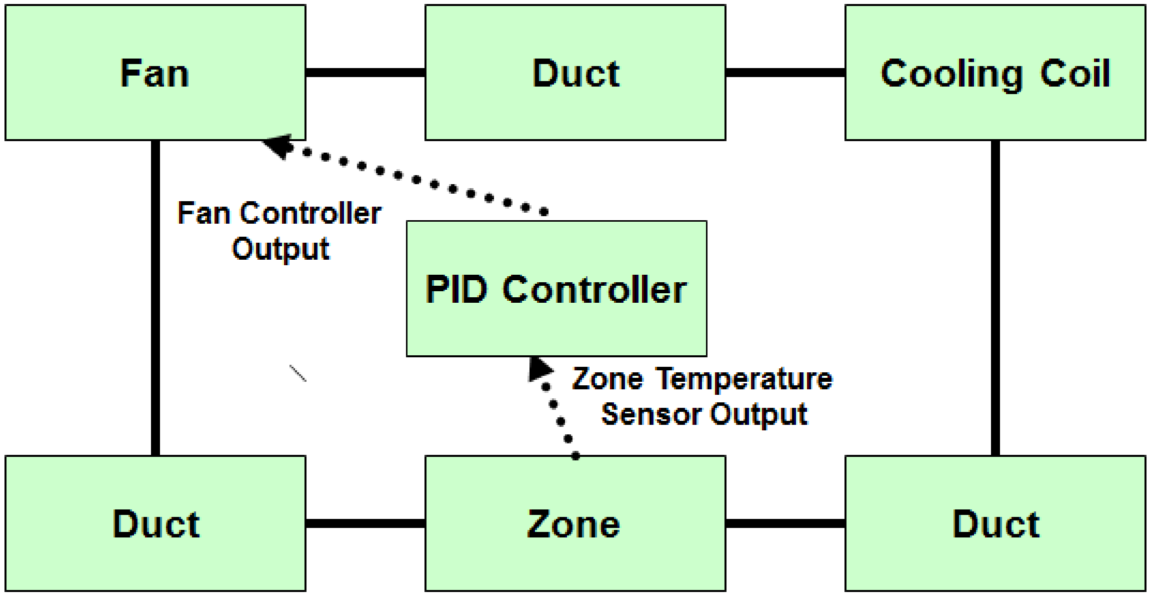

Figure 4-1 – Control Connections for a Simple System

A component's control outputs can depend on its control inputs. To ensure all control outputs can be resolved, the structure representing the control data flow must be directed and acyclic. The control system of Figure 4-1 obeys this constraint since the control data flows in one direction from the zone to the PID controller to the fan. Allowing the PID controller's control output to feed into input could create a situation that requires multiple iterations on a single time step to resolve.

4.1.4 Output Data Collecting

In PAQS, the simulation manager collects and stores all output data for a simulation. At the beginning of a simulation, each component contains information on what data to collect from it and the time steps or intervals to do so. The simulation manager extracts this information and collects data from components using each component's output data interface.

4.2 Modeling Fluids

Packets represent discrete, incompressible volumes of fluid. They operate in the same manner as cells do in the 1D convection modeling of CONTAM (Walton and Dols, 2005). Every packet has a type (i.e. air or water), a volume, a temperature, and other properties. For instance, air packets have a humidity ratio property. Packets do not have a pressure property, since system wide pressures use a separate model. Packets can be split and merged together, with the total fluid volume remaining the same.

Components use packets to model fluid internally and to send fluid between them. A system's packets model the distribution of its temperature within its fluids. Sending packets between components emulates fluid flow for a system. At the appropriate point during a time step, the simulation manager controls the exchange of packets between components. Packet propagation refers to this process.

Figure 4-2 shows how a pipe component in PAQS can represent its lengthwise temperature distribution using packets. This situation could arise when a pump changes its volume flow rate from time step to time step. This pipe, which has 6.0 ft^3 of total volume, has its internal fluid partitioned into three packets. The packet at the left end of the pipe has a volume of 2.25 ft^3 and a temperature of 41°F. The center packet has a volume of 2.0 ft^3 and has a temperature of 43°F. Finally, the packet at the right end of the pipe has a volume of 1.75 ft^3 and a temperature of 44°F.

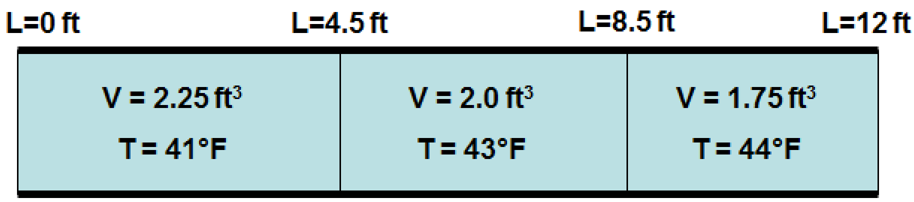

Figure 4-2 – A Section of Pipe with its Internal Packets

Packets can be split into multiples or merged into one. When splitting a packet, its volume is divided into multiple packets with the same temperature as the original packet. Figure 4-3 illustrates this.

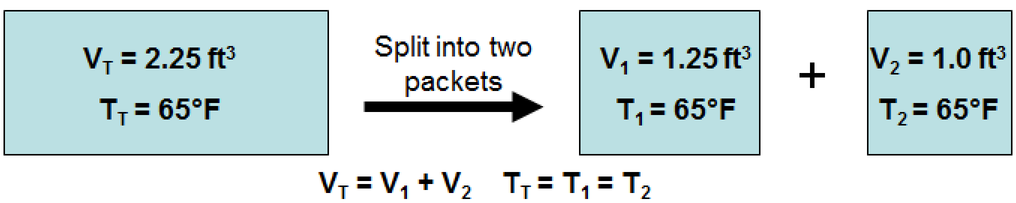

Figure 4-3 – A Packet Being Split Into Two

When merging multiple packets, the result has a volume equal to the sum of the volumes of the merged packets, and a temperature equal to the volume weighted average of the merged packet's temperatures. Figure 4-4 illustrates this.

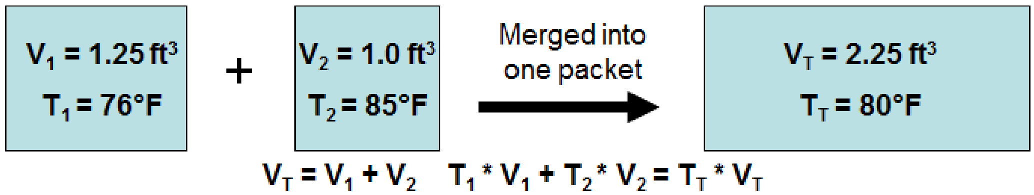

Figure 4-4 – Two Packets Being Merged Into One

Apart from the splitting and merging operations, all properties of packets can be changed minus the packet's volume. For example, a component can add heat to a packet to raise its temperature, but it cannot change the packet's volume in response to this rise in temperature. The next section discusses ramifications of such a change.

4.2.1 The Effects of Modeling Fluids as Incompressible

Packets make no reference to pressures. During a simulation, a packet changes temperature and its volume remains the same. Incompressible fluids, such as water, behave this way, but in contrast, compressible fluids like air change density as pressure changes. However, as PAQS primarily concerns itself with energy flow, this assumption holds true. COMIS and CONTAM, two building air flow modeling programs, also make use of this assumption to simplify their calculations (Feustel, 1999; Walton and Dols, 2005).

Figure 4-5 represents a worst-case scenario that may be seen when cooling air in an HVAC system. 100 ft^3 of air, initially at 110°F and at 100% relative humidity, is cooled to 50°F with the pressure remaining constant. After cooling and removing excess humidity, the air would then be 82.75 ft^3 in volume.

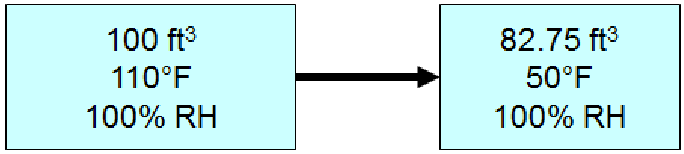

Figure 4-5 – Worst Case Cooling Density Change of Air

Figure 4-6 represents a worst-case scenario that may be seen when heating air in an HVAC system. 100 ft^3 of air initially at -10°F and at 0% relative humidity is heated to 110°F with the pressure remaining constant. After heating, the volume of air would be 126.69 ft^3 in volume. In these worst-case scenarios, the air's volume changes approximately 17% and 27%. However, in actual HVAC systems these extreme conditions rarely occur. Also, reuse of inside air lowers the temperature difference of any air cooled or heated. These two factors result in lower actual worst case air density variations; typically around 10%.

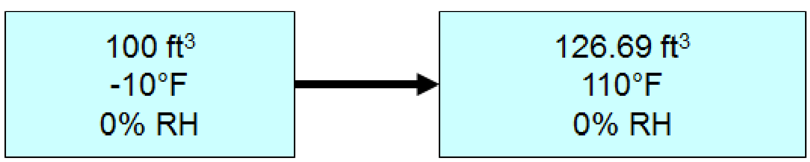

Figure 4-6 – Worst Case Heating Density Change of Air

In the actual operation of a system, changes in the density of air cause volume flow rate changes. In Figure 4-7, air with a temperature of 50°F and a density of 0.075 lbm/ft^3 flows through a heating coil at 1.0 ft^3/s. After being heated to 85°F, the air then has a density of 0.07 lbm/ft^3 and a flow rate of 1.07 ft^3/s.

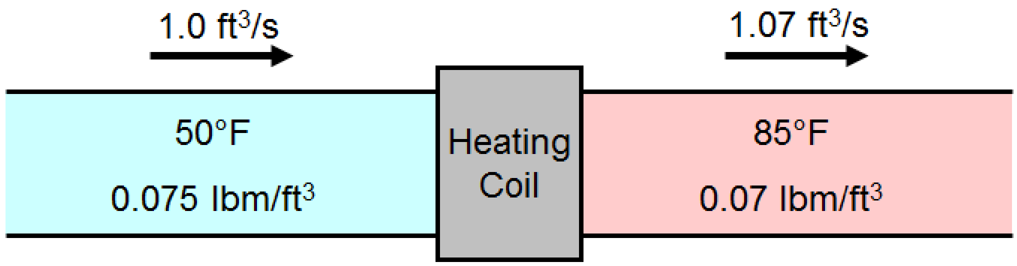

Figure 4-7 – Air Density Changes Past a Heating Coil

In contrast, the packet based system representation assumes constant density and volume flow rates along a duct, as shown in Figure 4-8. Regardless of the method used, the energy flow out of each duct stays constant since mass flows remain the same. Errors introduced using the packet based system representation for air includes time delay errors in the flow between components and variations in air's thermodynamic properties not accounted for. However, these errors remain small and are a function of the air's pressure.

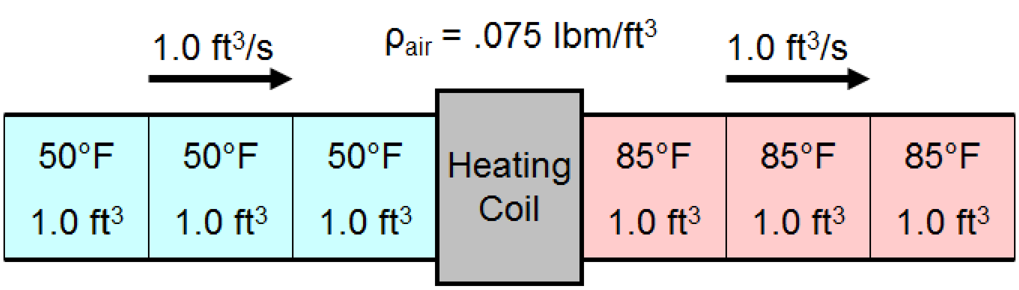

Figure 4-8 – Heating Coil Density Changes Using Packets

4.2.2 Fluid Ports

Components have ports where fluid packets enter or leave during a simulation. Each port has a given type depending on the type of packets it handles. A port's packet type remains the same during a simulation. For example, the cooling coil of Figure 4-9 has four ports: two air ports and two water ports. During a simulation, all ports connect to a single, different port of the same type. These connections determine where fluid is sent when it leaves a component.

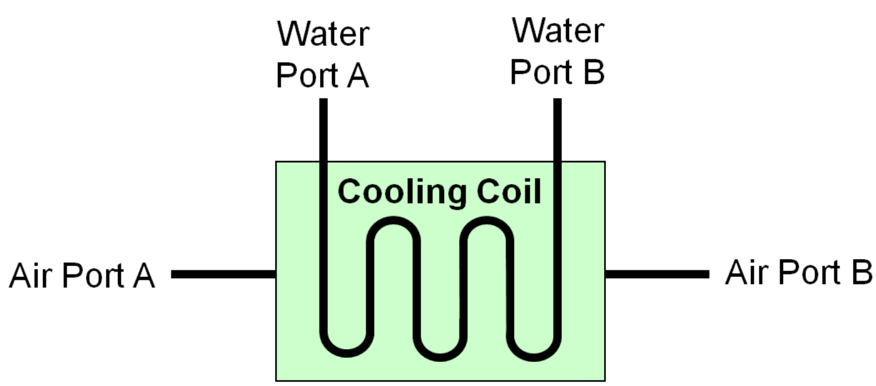

Figure 4-9 – The Ports of a Cooling Coil

In PAQS HVAC system components, ports can be implemented so that flow occurs in one direction or in either direction depending on system wide pressures. Packets enter or leave from port following the direction of flow at a port. This flow direction can change over time as a simulation progresses but remains fixed on a given time step. At a given time step, an input port refers to a port that takes in a packet, and an output port refers to a port that sends out a packet. Ports can also have no flow; in which case it no packets enter or leave it. Output ports and input ports always come in pairs since flow between components travels in a single direction between two ports.

4.2.3 Direct Triggering

On each time step of a PAQS simulation, the simulation manager emulates fluid flow by controlling the exchange of packets between the various components. This process, known as packet propagation, begins when the simulation manager instructs the components of select output ports to send a packet out of them. Direct triggering refers to this process.

When directly triggered, an output port releases a packet made from the component's internal fluid. This temporarily creates a void inside the component that persists until the component receives a volume of packets to fill it. Figure 4-10, Figure 4-11, and Figure 4-12 illustrate this.

The amount of fluid that a component internally stores limits the size of packets that can be sent out through direct triggering. For instance, a duct with a volume of 10 ft^3 can only send out a packet less than or equal to that size. Figure 4-13 shows two ducts with a volume of 20 ft^3 and two air packets of different sizes. The top air packet, with a volume of 30 ft^3, cannot be sent out of the duct that only holds 20 ft^3. However, the lower air packet with a volume of 15 ft^3 can be sent out of the duct.

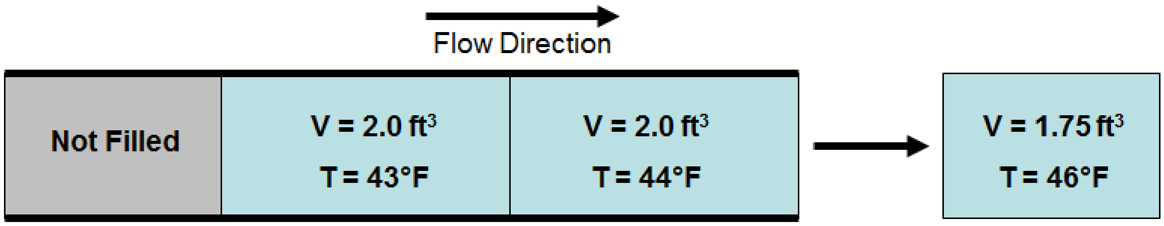

Figure 4-10 – A Directly Triggered Pipe

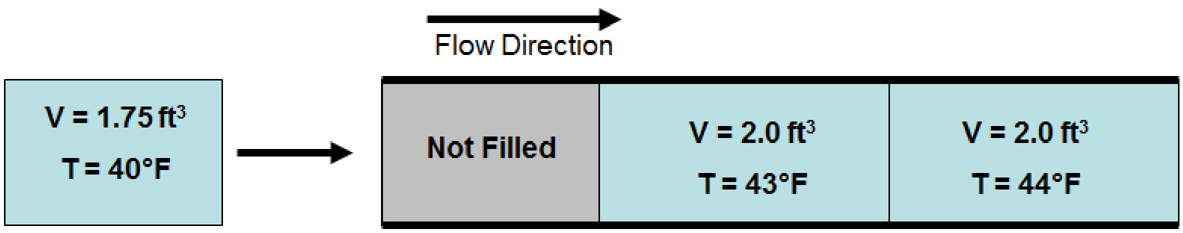

Figure 4-11 – A Packet Being Sent Into a Pipe

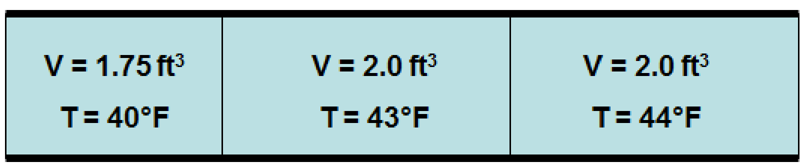

Figure 4-12 – A Pipe After Packet Propagation

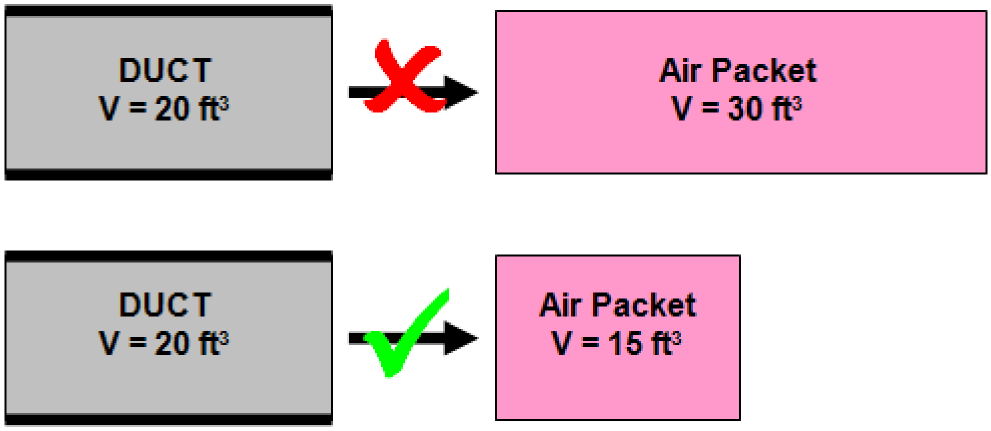

Figure 4-13 – Invalid and Valid Output Packet Sizes for a Duct

During a simulation, a system's volume flow rates and time step periods may be too large for components to be directly triggered just once for an entire time step. When this arises, PAQS splits the original time step period into equal subtime steps and performs packet propagation multiple times so that no component has to release more fluid than it can hold during direct triggering.

For example, suppose that the chilled water loop in Figure 4-14 has a volume flow rate of 3.0 ft^3/s in a simulation with a time step period of 10 seconds. Using a 10-second time step period, one of the components would need to be directly triggered so that it releases a packet with a volume of 30 ft^3. However, no component holds this amount of fluid. Therefore, PAQS automatically splits the 10 second time step into two 5 second subtime steps for packet propagation. This way, each directly triggered component only needs to send out 15 ft^3 at a time.

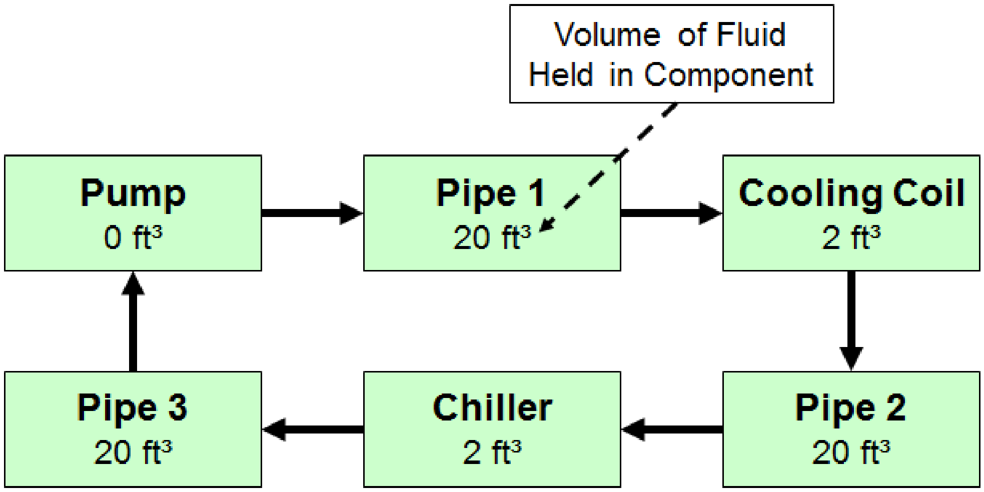

Figure 4-14 – A Chilled Water Loop with Internally Held Packets

4.2.4 Indirect Triggering

In a pipe filled with water, forcing water into one end pushes the same volume of water out the other end. Using packets to represent fluids, components in PAQS act in the same way. In the case of a pipe component, a packet receiving a packet in its input port will send out a packet of equal volume out of its output port. Indirect triggering refers to this process. Altogether, packet propagation occurs by directly triggering select output ports so that the rest of the system moves fluid through indirect triggering.

The following example takes place on a single time step of a simulation that uses one-second time step periods. On this time step, water flows through the pipe from Figure 4-15 at 2.25 ft^3/s. When packet propagation occurs, the component connected to the left end of the pipe sends a packet with 2.25 ft^3 of volume into the pipe's input port. When this occurs, the pipe moves its internal packets along its length, indirectly triggering its output port to release 2.25 ft^3 of water. In order to accomplish this, the pipe sends out its far right packet and 0.5 ft^3 of its middle packet, as shown in Figure 4-15. However, ports can only receive or release one packet during packet propagation, and as a result of this, the pipe combines the two packets sent out in Figure 4-16 into one packet, before passing it along to another component. Figure 4-17 illustrates this.

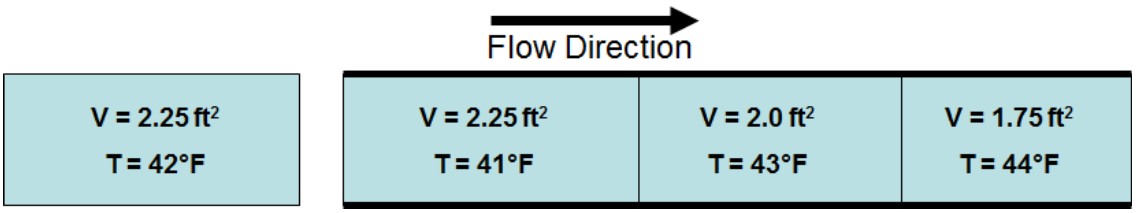

Figure 4-15 – A Pipe Before Accepting a Packet

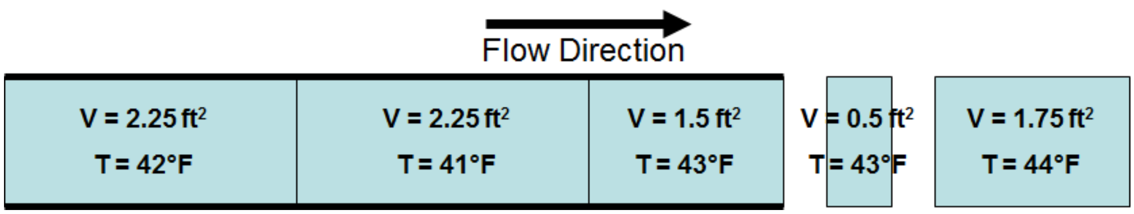

Figure 4-16 – Fluid Pushed Out of a Pipe

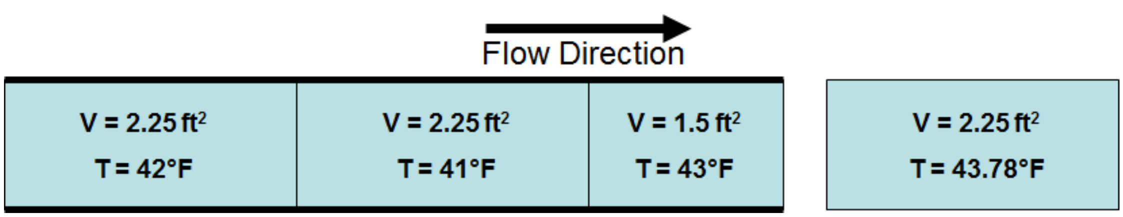

Figure 4-17 – A Pipe's Output Packet

Unlike in direct triggering, packets larger than the volume of fluid that an object holds can be sent out of an output port with indirect triggering. Figure 4-18 shows an air packet with a volume of 2.0 ft^3 being sent into a duct that holds 1.0 ft^3 of air. Figure 4-19 shows the fluid that is sent out the other end of the duct as a result of this. Finally, Figure 4-20 shows the fluid sent out combined into a packet to be sent to the duct's connected component.

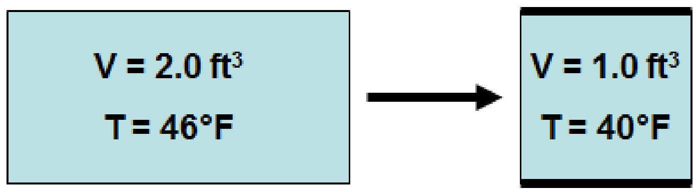

Figure 4-18 – A Large Air Packet Being Sent Into a Smaller Duct

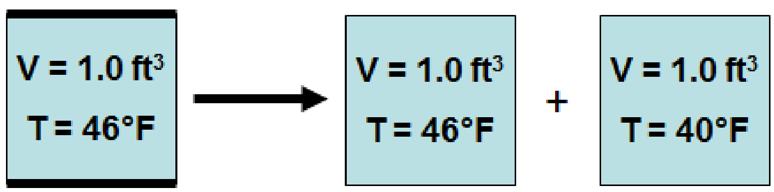

Figure 4-19 – The Fluid Output of a Duct

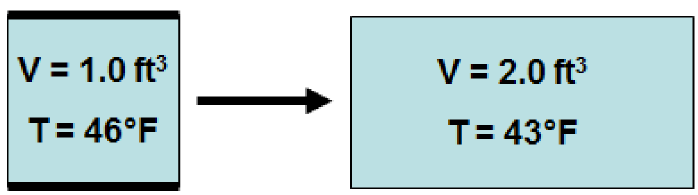

Figure 4-20 – The Packet Sent to Another Component

A component can have any number of ports. The inner workings of a component type and the direction of flow determine the behavior of its ports. Figure 4-21 shows an air-water heat exchanger. During packet propagation the two flows behave as an independent pipe and duct. A packet sent to its Water Port A indirectly triggers Water Port B, and a packet sent to Air Port A indirectly triggers Air Port B.

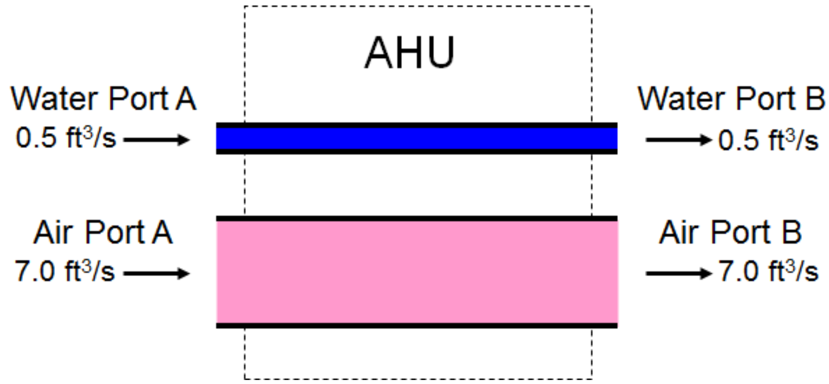

Figure 4-21 – The Port Connections of an Air-Water Heat Exchanger

These rules can change from time step to time step. For instance, if the flow direction of the duct in the AHU of Figure 4-21 changes, then Air Port A becomes an output port that gets indirectly triggered when Air Port B receives a packet of air. Figure 4-22 illustrates this.

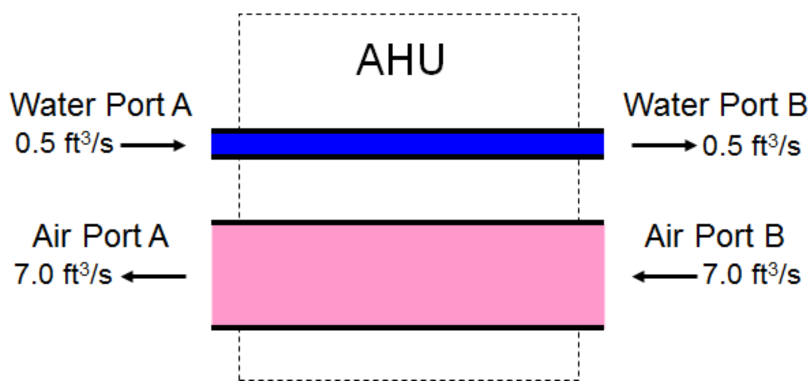

Figure 4-22 – An AHU Component with Different Flow Directions

4.2.5 Packet Propagation

Packet propagation refers to the process of emulating fluid flow, by moving packets between the components of a system. To control this process, the simulation manager ensures that all components move their packets correctly, while minimizing the number of subtime steps used.

Because several components in a system hold fluid internally, multiple directly triggerable ports may exist on each given time step. To perform packet propagation, the simulation manager first selects a set of ports to directly trigger.

In Figure 4-23, suppose that the water side only system has 5 directly triggerable ports for the given flow directions: the output ports of Pipes 1, 3, and 4 and AHUs 1 and 2. The simulation manager has four sets of ports it can directly trigger to ensure that packet propagation completes: the output ports of 1) Pipe 1 alone, 2) Pipe 3 alone, 3) Pipe 4 and AHU 2, or 4) AHU 1 and AHU 2.

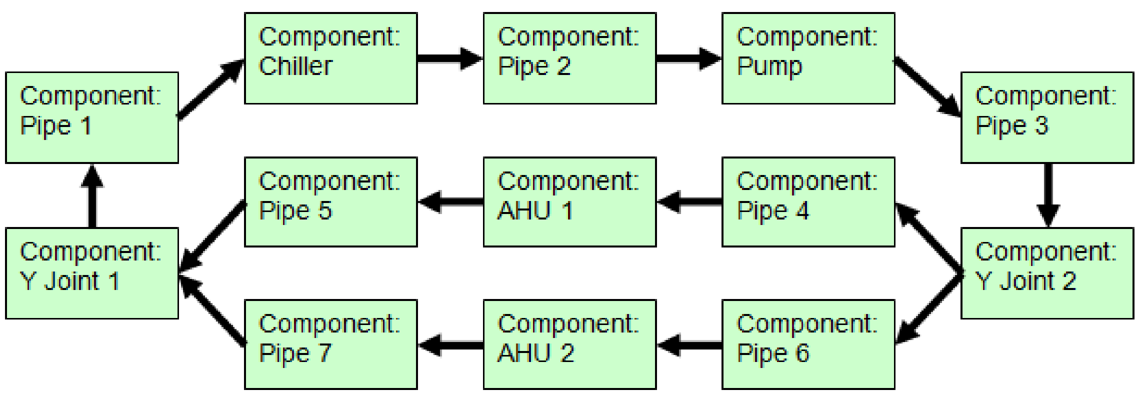

Figure 4-23 – Flow Directions in an HVAC System

To minimize the number of subtime steps required, the simulation manager first needs to know how many subtime steps each directly triggerable output port requires for the given flow rates and directions. Suppose that Table 4-1 lists this data for a given time step.

Table 4-1 – Subtime Steps Required

Component	Sub time steps Required by Component's Output Port
Pipe 1	4
Pipe 3	3
Pipe 4	2
AHU 1	3
AHU 2	2

The simulation manager then selects the combination of directly triggerable ports that optimally completes packet propagation. When directly triggering a single output port, PAQS simply uses the number of subtime steps that it requires. However, when directly triggering multiple output ports, PAQS uses enough subtime steps so that no object tries to move out more fluid than it contains.

For example, to directly trigger an output port that requires three subtime steps along with one that requires eight would require eight subtime steps. Table 4-2 lists the number of subtime steps required for each directly triggerable output port combination. In this case, the simulation manager would directly trigger the output ports of Pipe 4 and AHU 2 since they require the fewest number of subtime steps.

Table 4-2 – Subtime Steps for Triggering Combinations

Components Whose Output Ports to Directly Trigger	Sub time steps Required by This Combination
A) Pipe 1 alone	4
B) Pipe 3 alone	3
C) Pipe 4 and AHU 2	2
D) AHU 1 and AHU 2	3

4.3 Fluid Circuit Based Pressure and Flow Calculations

A fluid circuit uses electrical circuit analogies to solve fluid flow problems in the same way as electric circuit problems. PAQS models the pressure and flow of HVAC systems using a resistive fluid circuit. In this method, pressure behaves the same way as electrical voltage, flow behaves the same way as electrical current, and fluid resistance behaves the same way as electrical resistance.

In a fluid circuit representation of an HVAC system, objects that create a differential pressure, such as fans and pumps, behave the same way as voltage or current sources in an electric circuit. In addition, objects that impede fluid flow, such as ducts and dampers, act as fluid resistors. Finally, grounds present in fluid circuits provide a vacuum or reference pressure for the circuit. Figure 4-24 shows a fluid circuit representation for a single-zoned building.

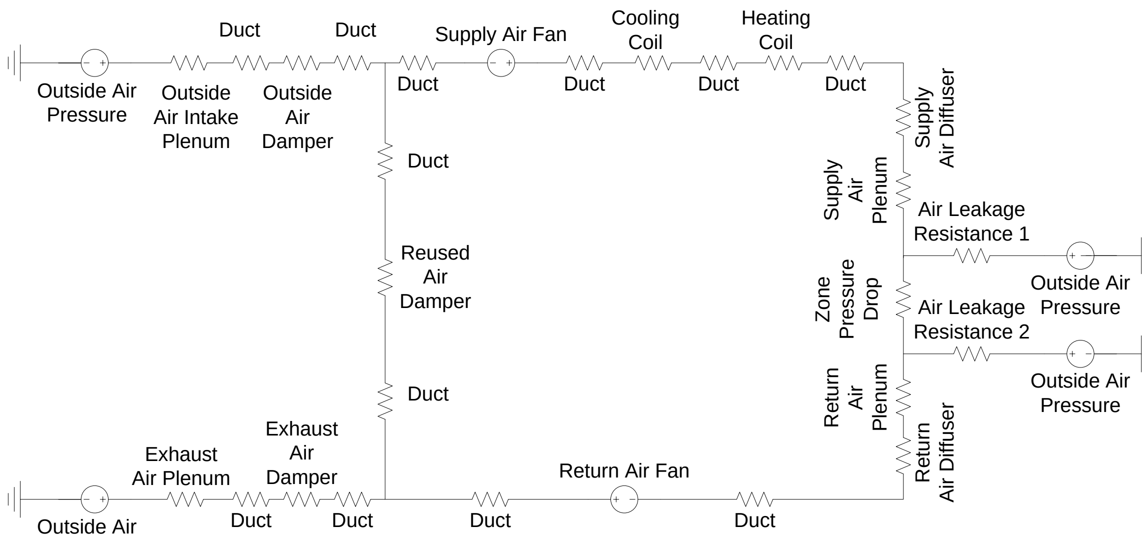

Figure 4-24 – A Single Zoned Building Represented as a Fluid Circuit

PAQS uses a fluid circuit representation of a system to determine its system wide pressures and volume flow rates. On each time step, the simulation manager recalculates system wide pressures for each port on every component. The components then use these port pressures to determine the system's volume flow rates, and their own internal pressures. Once all necessary components have their calculated volume flow rates for the new time step, PAQS can move fluid within the system using the packet based system representation.

Each component in a simulation contains an internal resistive fluid circuit model of itself. This fluid circuit model uses the same ports as its packet based system representation. Figure 4-25 gives the fluid circuit diagram for a dual duct mixing box, used for mixing streams of hot and cold air. The simulation manager constructs a system wide fluid circuit model from the port connections of the components and each component's internal fluid circuit model.

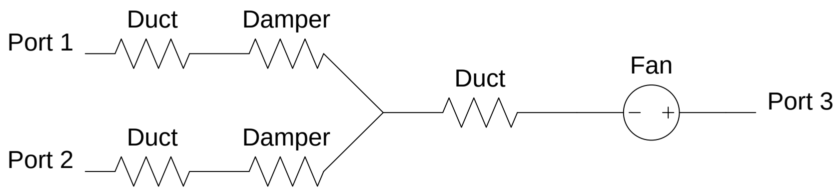

Figure 4-25 – A Dual Duct Mixing Box's Fluid Circuit

During a simulation, components can change the values of their internal resistances and differential pressures, once per time step. This causes volume flow rates and pressure values to vary throughout an entire system.

When viewing a system as a fluid circuit, a discrete number of nodes exist with each node having an associated pressure. While this representation gives pressures for where components join together, it says nothing about pressures inside components. A component can calculate pressures within itself using internal models and port pressures. For example, suppose a component representing a uniform 60 ft. long duct has port pressures of 1.02 atm and 0.99 atm and that the duct has a pressure sensor placed midway between each end. Pressure into the duct decreases constantly at 0.0005 atm/ft because of the duct's uniform nature. Therefore, the pressure sensor at the midpoint would read 1.005 atm. Figure 4-26 and Figure 4-27 illustrate this.

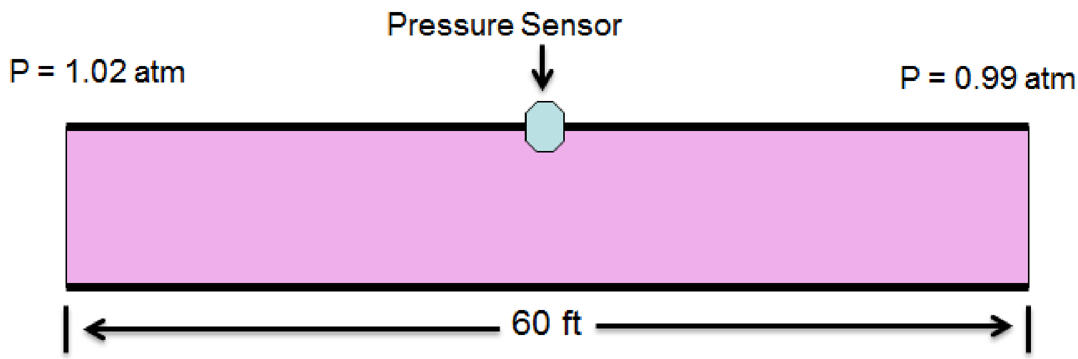

Figure 4-26 – A 60 ft Long Duct with Known Port Pressures

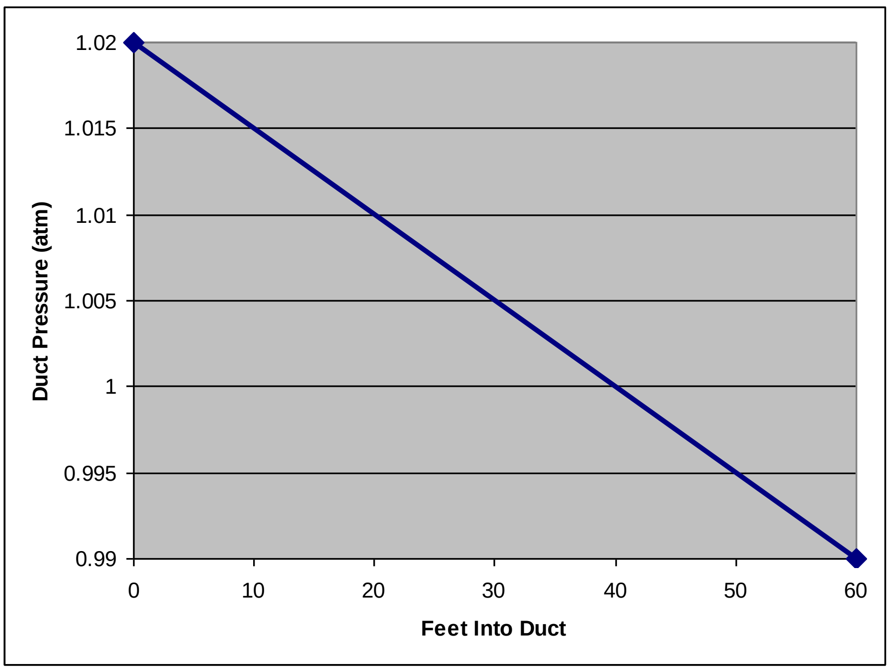

Figure 4-27 – Pressure Along a Duct's Length

4.3.1 Calculating Port Pressures

A directly triggered output port's component requires access to its internal volume flow rates, in order to know how much fluid to send out of the port. Sometimes, a component needs to calculate an internal pressure, such as when a duct has a pressure sensor attached to it. In both cases, a component's internal fluid circuit model, along with its port pressures, provides enough information to perform the necessary calculations.

PAQS calculates the port pressures of every component on a system wide level, on each time step. It then sends these pressures to the various components, which use them to calculate their internal volume flow rates and pressures. Components can then be directly triggered to start packet propagation.

In the following example, Figure 4-28 shows the fluid circuit model of a closed water loop. If the elements of this circuit hold the values shown in Figure 4-29 during a time step of a simulation, when PAQS calculates the port pressures of the system, it produces the results seen in Figure 4-30. PAQS then sends each component their newly calculated port pressures, enabling the components to calculate their internal volume flow rates and pressures. For instance, once the valve receives its port pressures of 6 psi and 2 psi, it can then use the value of its internal resistance, 2 psi/cfm from its internal fluid circuit model, to calculate its volume flow rate of 2 cfm, as illustrated in Figure 4-31.

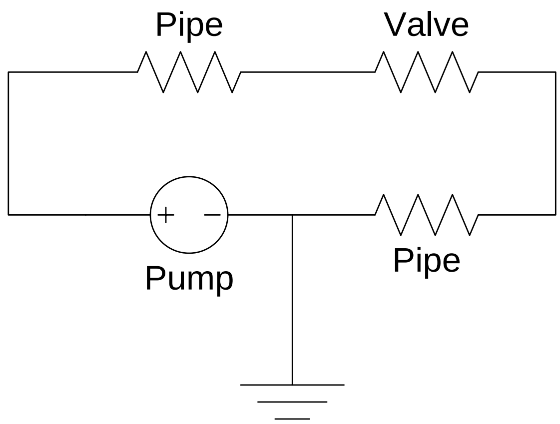

Figure 4-28 – A Fluid Circuit Representation of a System

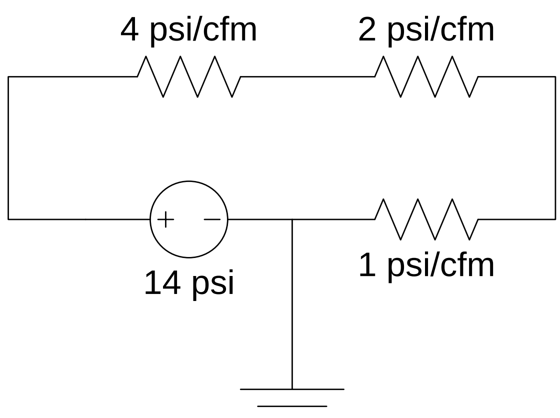

Figure 4-29 – Fluid Circuit Values on a Time Step

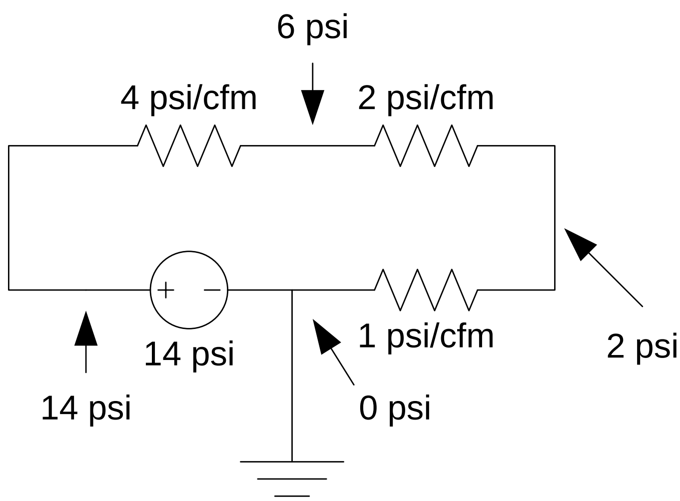

Figure 4-30 – Port Pressures on the Time Step

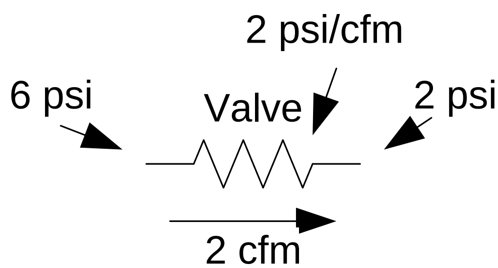

Figure 4-31 – A Valve's Flow Rate

4.3.2 Modified Nodal Analysis

In a linear fluid circuit, the values of all resistors, pressure sources, and flow sources provide enough information to calculate total pressures and volume flow rates throughout the system. PAQS uses a numerical technique known as modified nodal analysis, or MNA, to perform pressure and flow calculations on a system wide level (Ho et al., 1975).

MNA has benefits over other circuit-solving methods in that it works for nonplanar circuits, and calculates all volume flow rates and nodal pressures with respect to a system's ground. Also, MNA can be set up by inspection, and its results produce all nodal pressures and volume flow rates of a system without requiring any further calculations.

4.3.2.1 General Method

MNA transforms the system of equations of a fluid circuit problem into a matrix problem in the form **Ax = z**. The **A** matrix holds information on all fluid resistances in as system and how they connect with one another. The **z** vector contains the values of all pressure and flow sources in the system. From these two, the solution **x** vector holding all nodal pressures and pressure source flow rates can be calculated using the formula $\mathbf{x} = \mathbf{A}^{-1}\mathbf{z}$.

The number of nongrounded nodes where two or more circuit elements meet, n, and the number of pressure sources, m, in a fluid circuit determine the dimensions of the matrices and vectors. In the beginning of the MNA procedure, one first numbers all nodes from 1 to n excluding the ground node. Then, one numbers the pressure sources from 1 to m. Finally, one constructs the needed matrices.

4.3.2.2 The $\boldsymbol{A}$ Matrix

The $\boldsymbol{A}$ matrix of MNA is a $(m+n) \times (m+n)$ matrix that consists of four sub-matrices. The $\boldsymbol{G}$ matrix has $n \times n$ elements whose values get determined by the interconnections of resistances. The $\boldsymbol{B}$ matrix has $n \times m$ elements whose values get determined by the connection of the voltage sources. The $\boldsymbol{C}$ matrix is the transpose of the $\boldsymbol{B}$ matrix; i.e. $\boldsymbol{C} = \boldsymbol{B}^T$. Finally the $\boldsymbol{D}$ matrix consists of a $m \times m$ matrix containing only zero values.

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}$$

In the $\boldsymbol{G}$ matrix, each diagonal element equals the sum of the conductances touching that corresponding node number. Each diagonal element equals the negative conductance of the fluid resistor connected between the two node numbers.

The $\boldsymbol{B}$ matrix's elements consist of the values 0, 1, and -1. Rows of this matrix correspond to node numbers, while columns correspond to a particular pressure source number. An element in this matrix has a value of 1 if the positive terminal of its corresponding voltage source connects to the corresponding node. Likewise, an element has a value of -1, if the negative terminal of its corresponding pressure source connects to the corresponding node. Otherwise, an element has a value of zero.

4.3.2.3 The $\boldsymbol{z}$ Vector

The $\boldsymbol{z}$ vector consists of two subvectors; $\boldsymbol{i}$ and $\boldsymbol{e}$. The $\boldsymbol{i}$ vector consists of $n \times 1$ elements made up of the sum of the flow sources entering a given node. The $\boldsymbol{e}$ vector consists of $m \times 1$ elements whose values equal their corresponding pressure source's value.

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{e} \end{bmatrix}$$

4.3.2.4 The $\boldsymbol{x}$ Vector

The results matrix $\boldsymbol{x}$ consists of $(m+n) \times 1$ elements that can be divided into two submatrices; $\boldsymbol{v}$ and $\boldsymbol{j}$. The $\boldsymbol{v}$ vector consists of $n \times 1$ elements that hold the pressures at each node. The $\boldsymbol{j}$ vector consists of $m \times 1$ elements, and it holds the volume flow rates through each pressure source.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{j} \end{bmatrix}$$

4.3.3 Branches

When using MNA in PAQS, each node or pressure source in a system adds a row and a column to the problem's **A** matrix. In large systems, this presents a problem since the number of operations required to invert a $n \times n$ matrix typically increases with n^3. This means if a 4×4 matrix required 64 operations, a 5×5 matrix would need around 125. This can make matrix inversion a performance bottleneck for large systems. However, the structure of HVAC systems provides a way to minimize this problem.

In typical systems, such as the system from Figure 4-24, resistors, pressure sources, and flow sources often appear in series. Knowing the overall differential pressure and flow of a series branch allows the pressure drops of the circuit elements within the branch to be calculated without using matrices. Calculating the pressures and flows of a system by consolidating its elements into branches first reduces the dimension of the MNA matrices used, resulting in a more efficient calculation.

Each branch of a fluid circuit contains a number of resistors, pressure sources, and at most one flow source. To compress a branch into larger elements, PAQS combines all of its resistors into a single element, known as a branch resistor. It also combines all pressure sources on a branch into a single differential pressure element called a branch pressure source. Using the laws of series circuits, a branch resistor's value equals the sum of its individual resistances, and a branch pressure source's value equals the sum of its individual pressures. Figure 4-32 shows a typical fluid circuit branch, and Figure 4-33 shows how the multiple resistors and pressure sources on the branch combine to form a series branch resistor and a branch pressure source.

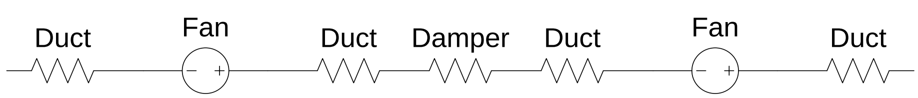

Figure 4-32 – A Typical Fluid Circuit Branch

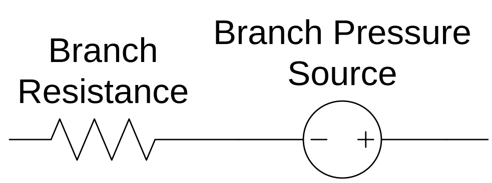

Figure 4-33 – Combined Components of a Branch

Figure 4-34 shows the system of Figure 4-24 represented as branch resistors and branch pressure sources. PAQS uses this representation, along with modified nodal analysis, to calculate the nodal pressures throughout systems.

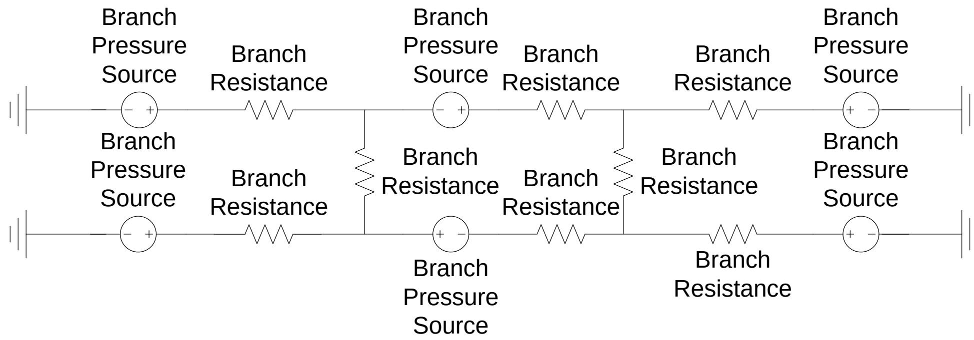

Figure 4-34 – A System Represented as Circuit Branches

4.3.4 Overall Pressure Calculation Example

Figure 4-35 shows a schematic of a cold water loop from an HVAC system. In this cold water loop, a chiller supplies chilled water to two cooling coils, while a pump and a valve regulate the water flow rate. Figure 4-36 shows this same cold water loop represented as a fluid circuit. The ground in the circuit serves as a reference pressure for the system.

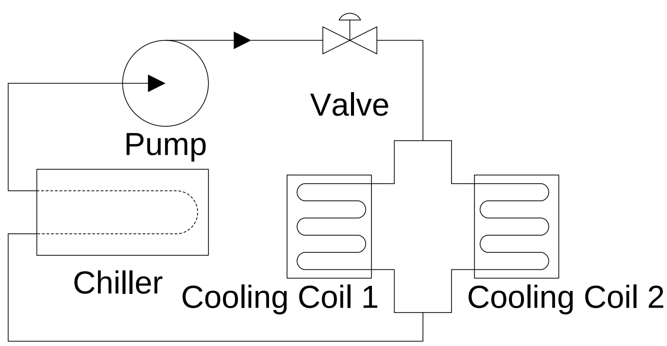

Figure 4-35 – An HVAC Cold Water Loop

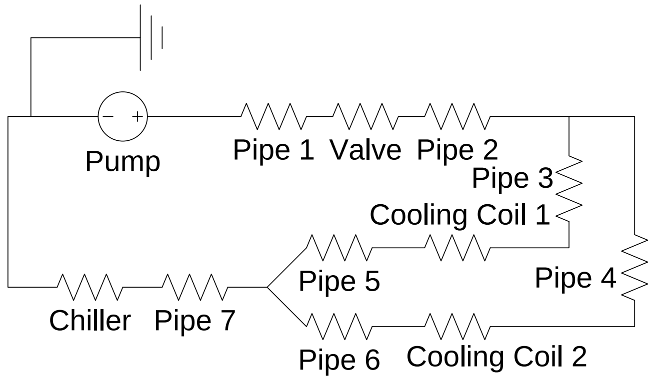

Figure 4-36 – A Fluid Circuit of a Cold Water Loop

In order to reduce the order of the matrix, which must be inverted, PAQS compresses the elements of the circuit from Figure 4-36 into branches, resulting in the system shown in Figure 4-37.

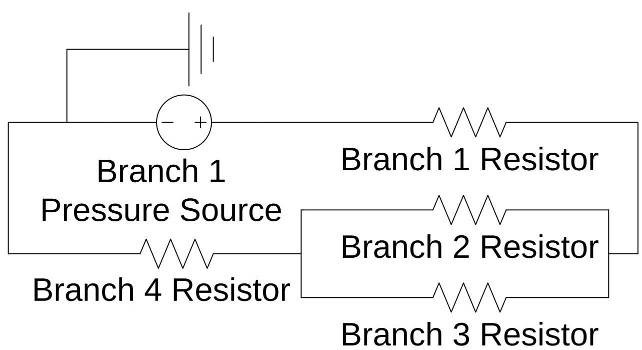

Figure 4-37 – A Branch Circuit of a Cold Water Loop

With the circuit elements of Figure 4-36 having the values shown in Table 4-3, the resulting branch circuit elements of Figure 4-37 have the values given in Table 4-4.

Table 4-3 – Values of Circuit Elements

Circuit Element	Value	Unit
Pipe 1	5.0	psi/cfm
Pipe 2	5.0	psi/cfm
Pipe 3	5.0	psi/cfm
Pipe 4	5.0	psi/cfm
Pipe 5	5.0	psi/cfm
Pipe 6	5.0	psi/cfm
Pipe 7	5.0	psi/cfm
Cooling Coil 1	7.0	psi/cfm
Cooling Coil 2	7.0	psi/cfm
Chiller	10.0	psi/cfm
Valve	16.5	psi/cfm
Pump	44.0	psi

Table 4-4 – Values of Branch Circuit Elements

Circuit Element	Value	Unit
Branch 1 Resistor	26.5	psi/cfm
Branch 2 Resistor	17.0	psi/cfm
Branch 3 Resistor	17.0	psi/cfm
Branch 4 Resistor	15.0	psi/cfm
Branch 1 Pressure Source	44.0	psi

Next, using the branch resistor and branch pressure source values from Table 4-4, PAQS calculates the nodal pressures of Figure 4-37 using Modified Nodal Analysis. This results in the nodal pressure values seen in Figure 4-38.

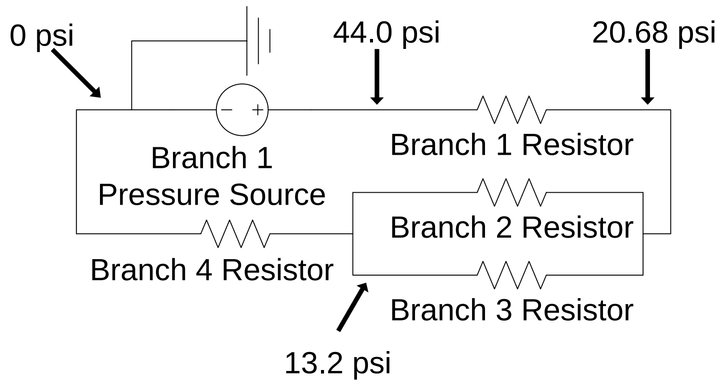

Figure 4-38 – Nodal Pressures of a Branch Circuit

PAQS then calculates the pressures from within each circuit branch, using the nodal pressures at each end of the branch, as shown in Figure 4-39. Every circuit element has its entering and leaving pressures calculated as a result, as shown in Figure 4-40.

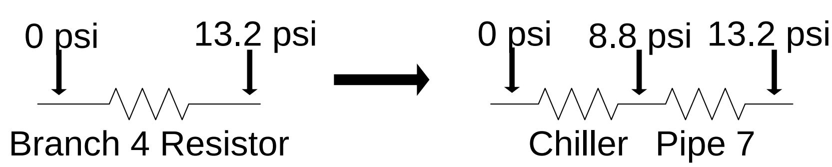

Figure 4-39 – Calculating Port Pressures of a Circuit Branch

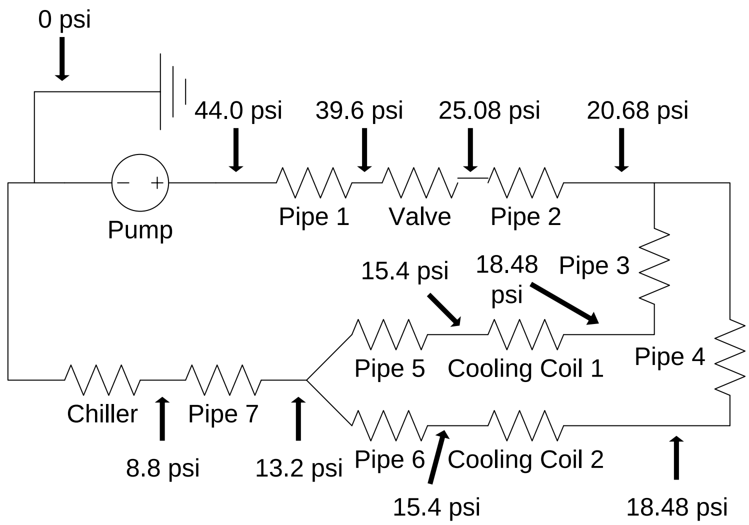

Figure 4-40 – Calculated Cold Water Loop Pressures

5. VALIDATION AND TESTING

5.1 Unit Testing

In small computer programs, a few manual tests performed after completing the code may be enough to ensure that the program works correctly. This type of testing does not scale up to large programs composed of many interacting modules. Performing extensive testing on one type of module cannot guarantee that small changes or bug fixes made later to that module won't affect its performance. Hard-coded tests known as unit tests help solve this problem.

Unit tests for a module of code run that module for different situations and check to see whether the actual results match the intended ones. During initial coding, unit tests check to see that each module works as intended. Later, during debugging, tests can be written for situations that cause errors so to test whether the errors get fixed. Throughout software development, unit tests for all written code can be periodically re-run to make sure bugs don’t appear in previously written code.

PAQS makes extensive use of unit testing. The unit tests written for PAQS cover the various code modules that work together to perform HVAC system calculations. When debugging PAQS as a whole, unit tests written for a module of code can detect errors within that module. Debugging efforts can then be focused on that particular area of the program. Without unit tests, the debugging process would be slowed by the need to manually review the inner workings of code modules to ensure that they work properly.

NUnit, a unit testing utility, executes the unit tests themselves (NUnit.org, 2011). ReSharper, a software development tool, integrates into the development environment and serves as a graphical interface for the unit tests (JetBrains, 2011). Visual Studio served as the development environment for writing all of PAQS's code (Microsoft, 2011).

5.2 Steady State System Test

Steady state HVAC simulations work by performing an energy balance on a building, assuming that all transients have died down. This test compares the steady state performance of PAQS against an independently validated methodology created by Dr. Jeffrey Haberl for the ASHRAE 865-RP project (Haberl et al., 2001). A typical duct constant air volume HVAC system was modeled using both methodologies for comparison.

Figure 5-1 shows the HVAC system layout utilized in this test. The system itself has two zones, each with a single thermostat. The system distributes air past a cooling coil, using a single duct with a constant airflow. If necessary, reheat coils heat up air entering a zone so overcooling is not performed. A return fan aids the flow of air from the zones. Some air is exhausted, and the remainder is mixed with makeup outside air and used as supply air.

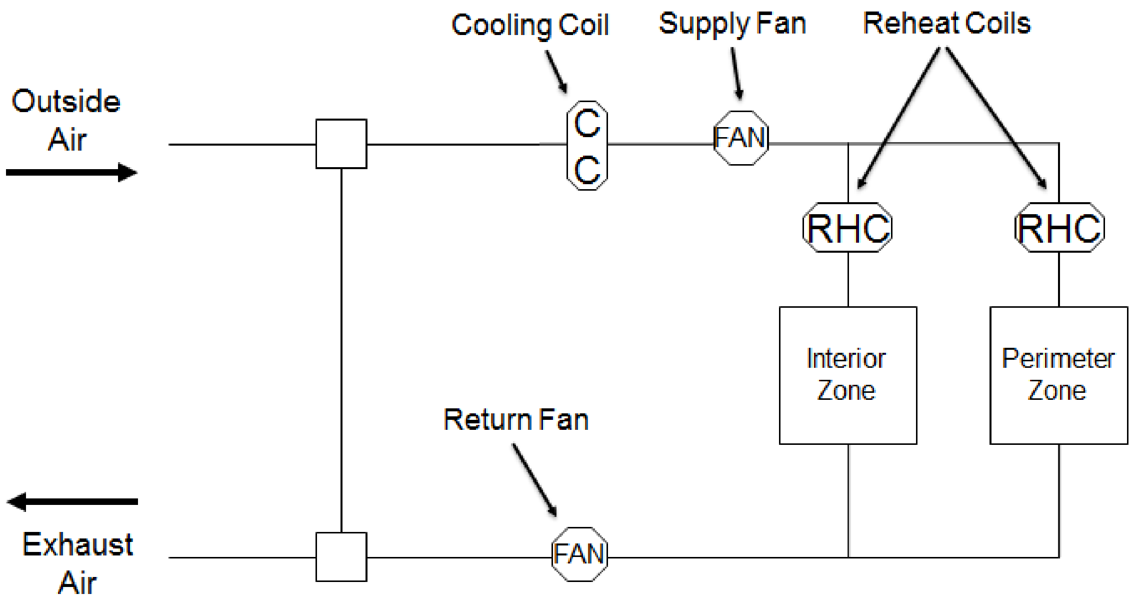

Figure 5-1 – A Single Duct Constant Air Volume System

Table 5-1 shows the numerical values used in this simulation. These values represent those of a typical single duct constant air volume system where the envelope thermal conductivity of the building has a large effect on the building's overall performance.

Table 5-1 – Single Duct Constant Air Volume System Parameters

Variable	Value	Description
Floor Area	1000.0 ft²	The total floor area
Interior Zone Percentage	30.0 %	The interior zone is 300.0 ft² while the perimeter zone is 700.0 ft²
Supply Airflow Rate	1.0 cfm/ft²	The amount of air flowing through the cooling coils and supply fan per ft² of floor area
Outside Airflow Rate	0.3 cfm/ft²	The amount of outside air entering the system per ft² of floor area
Zone Sensible Load	4.0 Btu/h• ft²	The sensible heat flow into each zone per ft² of floor area
Zone Latent Load	2.0 Btu/h• ft²	The latent heat flow into each zone per ft² of floor area
Zone Temperature Setpoint	75.0°F	The thermostat's temperature setpoint of each zone
Interior Zone Conductivity	0.02 Btu/h• ft²•°F	The interior zone's envelope thermal conductivity per ft² of floor area
Perimeter Zone Conductivity	0.1629 Btu/h• ft²•°F	The perimeter zone's envelope thermal conductivity per ft² of floor area
Supply Fan Delta T	2.0°F	The temperature increase across the supply fan
Return Fan Delta T	1.0°F	The temperature increase across the return fan
Cooling Coil Temperature Setpoint	55.0°F	The cooling coil cools any warmer air entering it to this temperature

The test simulated each system on a uniform grid of outside air conditions covering -10.0°F to 110.0°F and 0.0 % to 100.0 % relative humidity. Overall, this grid covers 24 different relative humidities and 365 outside air temperatures, for a total of 8,760 test outside air conditions.

Figure 5-2 shows the dynamic version of the single duct constant volume system as it appears in PAQS. In this model, a PID controller manages the reheat for each zone. Three fans exist, including the supply air fan, the return air fan, and the perimeter zone flow controller. This differs from the steady state layout, as the fan model in PAQS operates as analogs of current sources and multiple current sources cannot exist in series. However, they create identical functional results. The total heating power component serves as a data collector by summing up the reheat usage from each of the reheat coils.

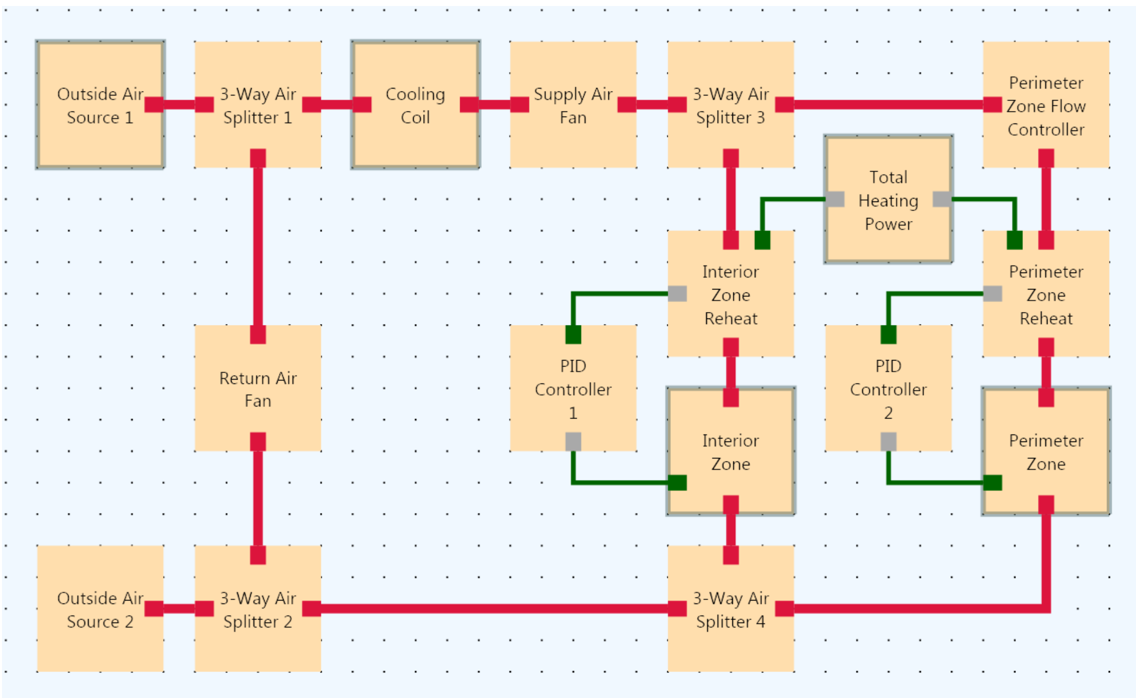

Figure 5-2 – A Steady State System in PAQS

Figure 5-3 shows PAQS's results as compared to Dr. Haberl's model. Overall, PAQS has a maximum cooling difference of 1.91% and a maximum heating difference of 2.96% using the error formulas below.

$$Cooling\ Percent\ Error = \frac{PAQS\ Cooling - Haberl\ Cooling}{Max\ Haberl\ Cooling} \times 100$$

$$Heating\ Percent\ Error = \frac{PAQS\ Heating - Haberl\ Heating}{Max\ Haberl\ Heaing} \times 100$$

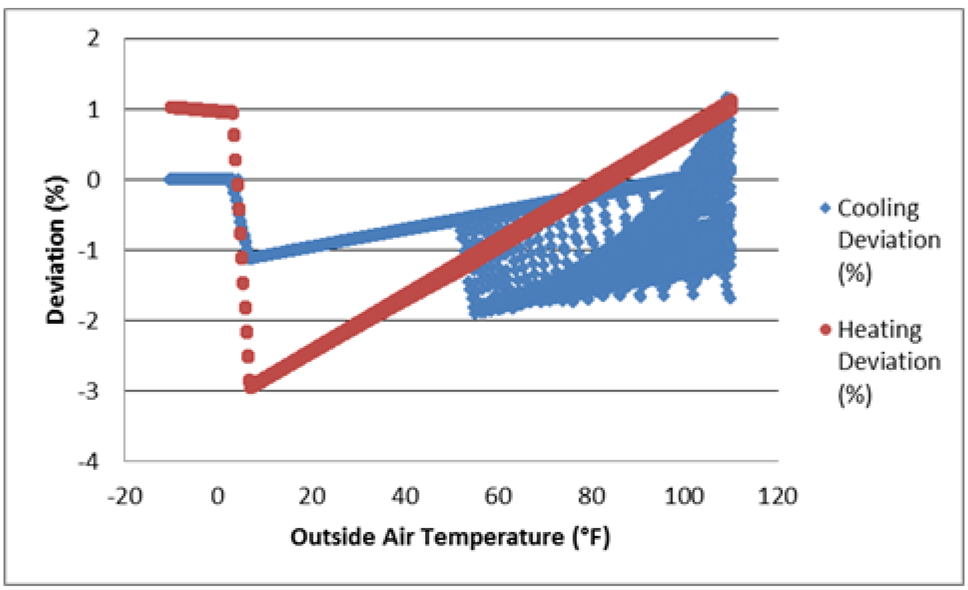

Figure 5-3 - Haberl vs. PAQS Cooling and Heating Usage Deviation

5.3 Dynamic System Test

This section compares the results of PAQS against an analytical dynamic model of an HVAC control system. This test uses a single-roomed building, served by a package unit type air conditioner as its test system. Figure 5-4 shows an overview of this type of building and its five modes of heat transfer. The room gains and loses heat through wall conduction, solar gains, heat generation from occupants and equipment, outside air leakage, and heat exchange through the air conditioner.

The air conditioning system itself has a fan that runs at a constant speed and adjusts the supply air temperature with a PI controller to manage the room temperature. In addition, outside air enters and leaves the building at a constant rate.

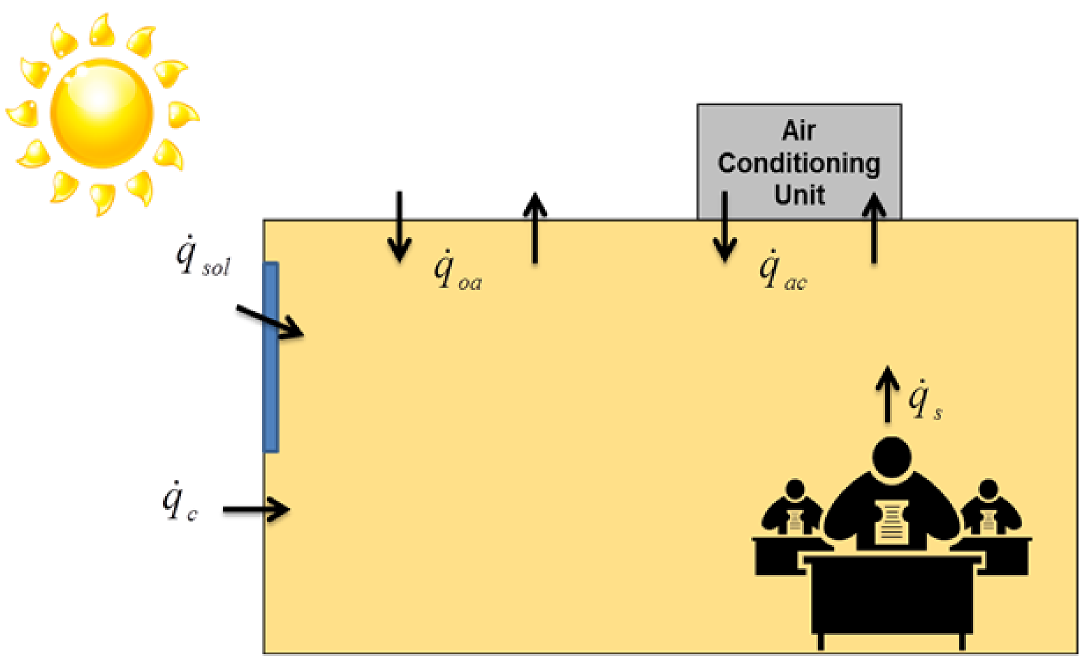

Figure 5-4 – Heat Transfer in a Package Unit AC System

5.3.1 Development of the Linear, Analytical HVAC System

Table 5-2 gives the nomenclature used in creating the linear, analytical system.

Table 5-2 – Nomenclature Used for the Analytical System

Variable	Unit	Description
$\dot{q}_r$	Btu/s	The total heat flow rate into the room
$\dot{q}_s$	Btu/s	The sensible heat generation within the room
$\dot{q}_c$	Btu/s	The sensible heat flow into the room from conduction through the perimeter
$\dot{q}_{sol}$	Btu/s	The sensible heat flow into the room from solar radiation
$\dot{q}_{oa}$	Btu/s	The sensible heat flow into the room from outside air exchange
$\dot{q}_{ac}$	Btu/s	The heat flow rate into the room due to the AC system
UA	Btu/s•°F	The total envelope heat conductivity of the room (A U-value times the wall area)
ρc_p	Btu/ft³•°F	The density times the specific heat capacity of air at constant pressure (assumed to be constant at 0.018072 Btu/ft³•°F
T_r	°F	The room's temperature
T_{oa}	°F	The outside air temperature
T_{ac}	°F	The temperature of the air supplied by the AC system
T_{sp}	°F	The room's temperature setpoint
T_e	°F	The error in the room's temperature $(T_e = T_r - T_{sp})$
T_{ei}	°F	The integrated room temperature error
V_r	ft³	The volume of air held within the room
$\dot{V}_{oa}$	ft³/s	The outside air exchange rate
$\dot{V}_{ac}$	ft³/s	The amount of air flowing through the AC system
K_P	dimensionless	A PI controller's proportional constant
K_I	dimensionless	A PI controller's integral constant
a, b, c	varies	Constants used to simplify the overall system equations

A heat balance of the entire building gives the following equation:

$$\dot{q}_r = \dot{q}_c + \dot{q}_{ac} + \dot{q}_{ac} + \dot{q}_s + \dot{q}_{sol}$$

Broken into input components, this equals:

$$\rho c_p V_r \dot{T}_r = UA(T_{oa} - T_r) + \rho c_p \dot{V}_{ac}(T_{ac} - T_r) + \rho c_p \dot{V}_{oa}(T_{oa} - T_r) + \dot{q}_s + \dot{q}_{sol}$$

This can then be rearranged and substituted to simplify the equations down:

$$\dot{T}_r = -\left(\frac{UA}{\rho c_p V_r} + \frac{\dot{V}_{oa}}{V_r} + \frac{\dot{V}_{ac}}{V_r}\right) T_r + \frac{\dot{V}_{ac}}{V_r} T_{ac} + \left(\frac{UA}{\rho c_p V_r} + \frac{\dot{V}_{oa}}{V_r}\right) T_{oa} + \frac{1}{\rho c_p V_r} \dot{q}_s + \frac{1}{\rho c_p V_r} \dot{q}_{sol}$$

$$a = \frac{UA}{\rho c_p V_r} + \frac{\dot{V}_{oa}}{V_r}, b = \frac{\dot{V}_{ac}}{V_r}, c = \frac{1}{\rho c_p V_r}$$

$$\dot{T}_r = -(a + b)T_r + bT_{ac} + aT_{oa} + c\dot{q}_s + c\dot{q}_{sol}$$

The AC uses a PI controller that adjusts its supply air temperature depending on the error in the room's temperature. To handle this, these equations are rewritten to acknowledge this error.

$$T_e = T_r - T_{sp} \rightarrow T_r = T_e + T_{sp},\ \dot{T}_e = \dot{T}_r$$

$$\dot{T}_e = -(a + b)T_e - (a + b)T_{sp} + bT_{ac} + aT_{oa} + c\dot{q}_s + c\dot{q}_{sol}$$

The PI controller also uses the integrated error, T_{ei}:

$$T_{ei} = \int_0^t T_e$$

Written in state space form, the open loop system appears as follows:

$$\begin{bmatrix} \dot{T}_e \\ \dot{T}_{ei} \end{bmatrix} = \begin{bmatrix} -a - b & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} T_e \\ T_{ei} \end{bmatrix} + \begin{bmatrix} b & -a - b & a & c & c \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{ac} \\ T_{sp} \\ T_{oa} \\ \dot{q}_s \\ \dot{q}_{sol} \end{bmatrix}$$

The closed loop controller sets its temperature setpoint as follows:

$$T_{ac} = K_P T_e + K_I T_{ei}$$

This results in the following closed loop system:

$$\dot{T}_e = -(a+b)T_e - (a+b)T_{sp} + bK_P T_e + bK_I T_{ei} + aT_{oa} + c\dot{q}_s + c\dot{q}_{sol}$$

$$\dot{T}_{ei} = T_e$$

The entire system in state space form, $\dot{x} = Ax + Bu, y = Cx + Du$, appears as:

$$\begin{bmatrix} \dot{T}_e \\ \dot{T}_{ei} \end{bmatrix} = \begin{bmatrix} -a-b+bK_P & bK_I \\ 1 & 0 \end{bmatrix} \begin{bmatrix} T_e \\ T_{ei} \end{bmatrix} + \begin{bmatrix} -a-b & a & c & c \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{sp} \\ T_{oa} \\ \dot{q}_s \\ \dot{q}_{sol} \end{bmatrix}$$

$$[T_r] = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} T_e \\ T_{ei} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{sp} \\ T_{oa} \\ \dot{q}_s \\ \dot{q}_{sol} \end{bmatrix}$$

The analytical form of the overall time domain solution of the can be found using the overall state space equations in the following equation:

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}(0) + \int_0^t e^{\boldsymbol{A}(t-\tau)} B\boldsymbol{u}(\tau)d\tau$$

5.3.2 Comparisons between Analytical Results and PAQS

Figure 5-5 shows the linear analytical system as it appears when implemented in PAQS. In this model, air flows in a clockwise direction. The zone outputs its temperature to a PID controller. In turn, this controller's output serves as the temperature setpoints for the air heater and air cooler. The computer algebra program Maple aided in performing the analytical calculations.

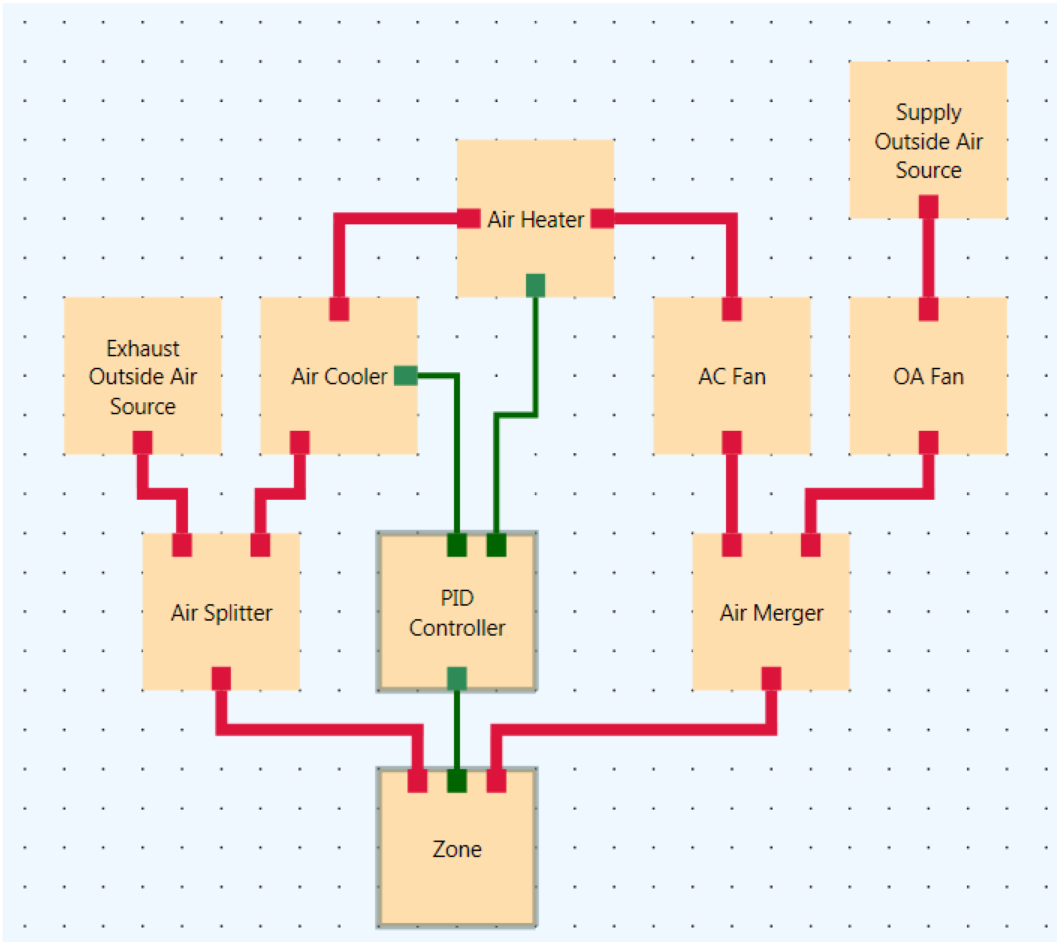

Figure 5-5 – The Linear Analytical System in PAQS

The system parameters used to compare the transient responses of an analytical system and PAQS derive from the parameters listed in Table 5-3. These values represent a typical office building in a cool climate.

Table 5-3 – Values Used to Derive the System's Inputs

Variable	Value
Outside Air Temperature	45°F
Floor Area	10,000 ft^2
Floor Height	10 ft
Room UA Value	100 Btu/h•°F
Room Temperature Setpoint	72°F
Room Solar Load	10,000 Btu/h
Room Sensible Load	60,000 Btu/h
Outside Airflow	1,000 CFM
AC Supply Airflow	9,000 CFM
ρc_p	0.018072 Btu/ft^3•°F

Figure 5-6 and Figure 5-7 show the transient response of PAQS and an analytical solution for an overdamped and underdamped case. The systems were simulated starting with the room temperature at its setpoint and integrated error initially equal to zero. Time step periods of 1/128th of a second were used to maximize accuracy. Since the supply air temperature setpoint comes directly from the PID controller output, the AC system starts in a cooling mode. However, the supply air temperature self-corrects as the integral response builds and the system eventually reaches a steady state with no error.

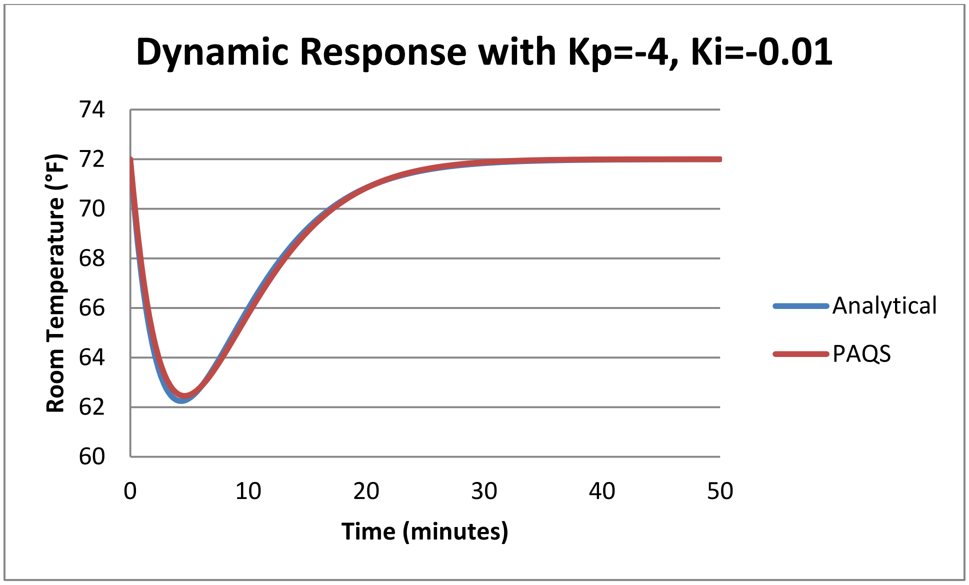

Figure 5-6 – A System with an Overdamped Response

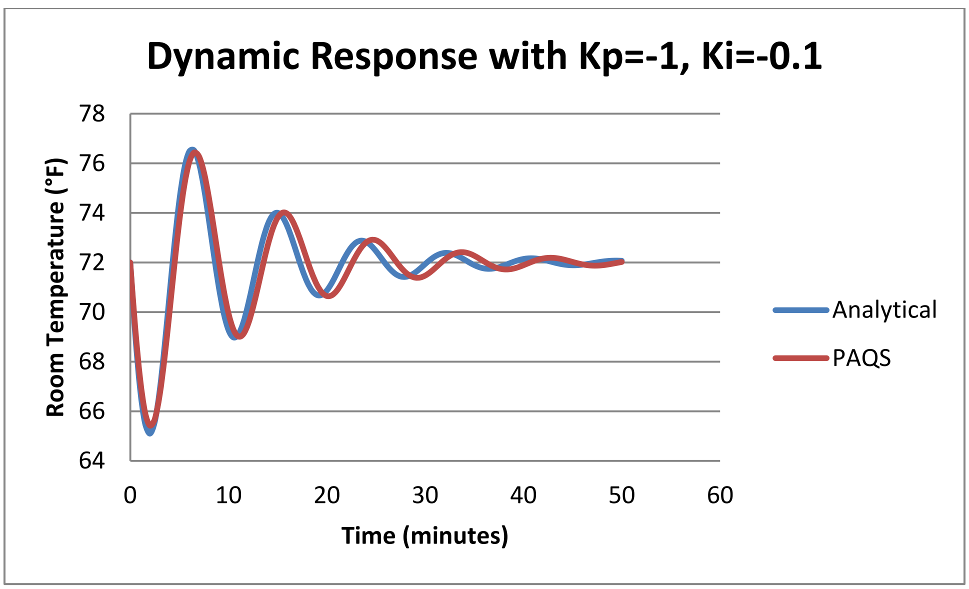

Figure 5-7 – A System with an Underdamped Response

6. CONCLUSION

Overall, PAQS demonstrates a fully functional HVAC system simulation methodology and prototype implementation. With it, users can create and test new and complete HVAC systems and HVAC control system methodologies. Test results show that all key features work together as intended to model HVAC systems.

6.1 Results Summary

PAQS allows dynamic HVAC system models to be constructed from basic components in the same way as real HVAC systems. With it, current HVAC systems and control methodologies can be tested and new systems and methodologies can be developed. Using available components, no practical limits exist on a system's size or complexity. PAQS's computational performance makes year-long dynamic calculations and iterative dynamic system optimizations practical on desktop PCs.

The test results show agreement between steady state simulations as well as an analytical dynamic building simulation. The steady state simulation validated the overall energy use prediction potential of PAQS within 3.0% of the tested results, by comparing the steady state results of PAQS against a published steady state building simulation methodology. Tests of the dynamic performance of PAQS showed near identical performance against an analytical dynamic system.

6.2 Future Work

6.2.1 Implementing and Testing New Component Models

With the core calculation engine completed and tested, the variety of thoroughly tested component models serves as the limiting factor for the scope of PAQS. Since any modeled feature of a building or HVAC system has to be implemented in a component, having a large component library helps to ensure that unique or novel systems or control configurations can be modeled. However, any component used requires validation to ensure that it does not cause errors in a simulation.

6.2.2 User Interface Refinements

Developed as a prototype for demonstrating the calculation methodology, the current user interface of PAQS can be extended to better encompass the work flow of HVAC and control system engineers and researchers. Extending this interface would also allow other types of calculations to be performed such as optimizing control settings or calibrating a model's input parameters to measured data.

6.2.3 Pressure Modeling Refinements

Further improvements can be made on the way PAQS models fluids. The linear, resistive fluid circuit modeling methodology fails to model pressures correctly when flow rates vary with time within a system. A nonlinear fluid circuit model would allow true system wide pressure modeling. The modular nature of PAQS allows for the current linear fluid circuit model to be directly replaced with a more sophisticated model.

REFERENCES

Ahmad, M., and Culp C. H. 2006. Uncalibrated Building Energy Simulation Modeling Results. HVAC&R Research 12(4): 1141-1155.

Anderson, M., Buehner M., Young P., Hittle D., Anderson C., Tu J. L., and Hodgson D. 2007. An experimental system for advanced heating, ventilating and air conditioning (HVAC) control. Energy and Buildings 39(2): 136-147.

Argüello-Serrano, B., and Vélez-Reyes M. 1995. Design of a nonlinear HVAC control system with thermal load estimation. Proceedings of the 4th IEEE Conference on Control Applications; Albany, NY.

ASHRAE 1996. 1996 ASHRAE Handbook - Heating, Ventilating, and Air-Conditioning Systems and Equipment. Inch-Pound Edition ed. American Society of Heating, Refrigeration, and Air-Conditioning Engineers, Inc., Atlanta, GA.

ASHRAE 2001. 2001 ASHRAE Handbook - Fundamentals. Inch-Pound Edition ed. American Society of Heating, Refrigeration, and Air-Conditioning Engineers, Inc., Atlanta, GA.

Aydin, C., and Ozerdem B. 2006. Air leakage measurement and analysis in duct systems. Energy and Buildings 38(3): 207-213.

Bendapudi, S., Braun J. E., and Groll E. A. 2005. Dynamic model of a centrifugal chiller system - Model development, numerical study, and validation. ASHRAE Transactions 111(1): 132-148.

Birdsall, B., Buhl W. F., Ellington K. L., Erdem A. E., and Winkelmann F. C. 1990. Overview of the DOE-2 building energy analysis program: Version 2.1d. Simulation Research Group: Lawrence Berkeley Laboratory: University of California. February 1990; Berkeley, CA

Buhl, W. F., Erdem A. E., Winkelmann F. C., and Sowell E. F. 1993. Recent improvements in SPARK: Strong component decomposition, multivalued objects, and graphical interface. Building Simulation 1993. International Building Performance Simulation Association; Adelaide, Australia.

CERL 1999. Building loads and system thermodynamics (BLAST) fact sheet. U.S. Army Corps of Engineers Construction Engineering Research Laboratory. March 1999; Champaign, IL

Chen, Q. 1997. Computational fluid dynamics for HVAC: Successes and failures. ASHRAE Transactions 103(1): 178-187.

CIBSE 2000. CIBSE Guide H: Building Control Systems. Butterworth-Heinemann, Woburn, MA.

Claridge, D. E., Haberl J. S., Abushakra B., and Sreshthaputra A. 2003. Electricity diversity profiles for energy simulation of office buildings. ASHRAE Transactions 110(1): 365-377.

Clark, D. R., Hurley C. W., and Hill C. R. 1985. Dynamic models for HVAC system components. ASHRAE Transactions 91(1B): 737-51.

Coulbeck, B. 1980. Dynamic simulation of water distribution systems. Mathematics and Computers in Simulation 22(3): 222-230.

Crawley, D. B., Winkelmann F. C., Lawrie L. K., and Pedersen C. O. 1999. EnergyPlus: A new-generation building energy simulation program. Building Simulation 1999. International Building Performance Simulation Association; Kyoto, Japan.

Djunaedy, E., Hensen J. L. M., and Loomans M. G. L. C. 2005. External couplings between CFD and energy simulation: Implementation and validation ASHRAE Transactions 111(1): 612-624.

Feustel, H. E. 1999. COMIS - An international multizone air-flow and contaminant transport model. Energy and Buildings 30(1): 3-18.

Filion, Y. R., and Karney B. W. 2003. Sources of error in network modeling: A question of perspective. Journal of the American Water Works Association 95(2): 119-130.

Fischer, D. E., Taylor R. D., Buhl F., Liesen R. J., and Strand R. K. 1999. A modular, loop-based approach to HVAC energy simulation and its implementation in EnergyPlus. Building Simulation 1999; Kyoto, Japan.

Fisk, W. J., Delp W., Diamond R., Dickerhoff D., Levinson R., Modera M., Nematollahi M., and Wang D. 2000. Duct systems in large commercial buildings: Physical characterization, air leakage, and heat conduction gains. Energy and Buildings 32(1): 109-119.

Garcia-Borras, T. 1983. Manual for Improving Boiler and Furnace Performance. Gulf Publishing Company, Houston, TX.

Griffiths, D., and Zuluaga M. 2004. An analysis of the effective R-value for insulation buried attic ducts. ASHRAE Transactions 110(2): 721-726.

Gupton, G. W. 2002. HVAC Controls: Operation & Maintenance. Third Edition ed. The Fairmont Press, Inc., Lilburn, GA.

Haberl, J., Saman N., and Bou-Saada T. 2001. "Air-side Accuracy Test Toolkit," ASHRAE software for air-side accuracy tests for seven HVAC systems: Dual-duct, Constant Volume (DDCAV); Dual-duct, Variable Volume (DDVAV); Constant Volume With Reheat (CAVRH); Variable Volume With Reheat (VAVRH); Four Pipe, Fan Coil Unit (FC); Four Pipe Induction Unit (FI); and Single Zone (SZ) Systems. Energy Systems Laboratory, Texas A&M University;

Haberl, J. S., and Cho S. 2004. Literature review of uncertainty analysis methods. Energy Systems Laboratory: Texas A&M University. November 2004; College Station, TX

Hepworth, S. J., and Dexter A. L. 1994. Neural control of non-linear HVAC plant. Proceedings of the Third IEEE Conference on Control Applications, Vol. 3, pp. 1849-1854; Glasgow, UK.

Ho, C.-W., Ruehli A., and Brennan P. 1975. The modified nodal approach to network analysis. IEEE Tansactions on Circuits and Systems 22(6): 504-509.

Irving, A. D. 1988. Validation of dynamic thermal models. Energy and Buildings 10: 213-220.

JetBrains 2011. Resharper; Prague, Czech Republic

Klein, S. A., Beckman W. A., and Duffie J. A. 1976. TRNSYS - A transient simulation program. ASHRAE Transactions 82(1): 623-633.

Knebel, D. E. 1983. Simplified Energy Analysis Using the Modified Bin Method. American Society of Heating, Refrigerating, and Air-Conditioning Engineers.

Langley, B. C. 2000. Fundamentals of Air Conditioning Systems. 2nd Edition ed. The Fairmont Press, Inc., Lilburn, GA.

LBNL 2003. SPARK 2.0 reference manual. Lawrence Berkeley National Laboratory; Berkeley, CA

Letherman, K. M. 1981. Automatic Controls for Heating and Air Conditioning: Principles and Applications. 15 Pergamon Press Inc., Elmsford, NY.

Levenhagen, J. I. 1999. HVAC Control System Design Diagrams. McGraw-Hill, New York, NY.

Majumdar, A., Bailey J. W., Schallhorn P., and Steadman T. 1998. A generalized fluid system simulation program to model flow distribution in fluid networks.

MathWorks 2011. Simulink. The MathWorks, Inc; Natick, MA

McDowell, T. P., Bradley D. E., Thornton J. W., and Kummert M. 2004. Simulation synergy: Expanding TRNSYS capabilities and usability. SimBuild 2004, IBPSA-USA National Conference; Boulder, CO.

Microsoft 2011. Visual Studio; Redmond, WA

Migliavacca, F., Dubini G., and Leval M. R. d. 2000. Computational fluid dynamics in paediatric cardiology. Images in Paediatric Cardiology 2: 11-25.

Moshiri, B., and Rashidi F. 2004. Self-tuning based fuzzy PID controllers: Application to control of nonlinear HVAC systems. Intelligent Data Engineering and

Automated Learning – IDEAL 2004, Vol. 3177, pp. 437-442. Springer Berlin / Heidelberg.

Newman, H. M. 1994. Direct Digital Control of Building Systems: Theory and Practice. (M. D. Morris, Ed.) John Wiley & Sons, Inc., New York, NY.

NUnit.org 2011. NUnit. www.nunit.org.

Park, C., Clark D. R., and Kelly G. E. 1985. An overview of HVACSIM^{+}, a dynamic building/HVAC/control systems simulation program. 1st. Annual Building Energy Simulation Conference; Seattle, WA.

Riederer, P., Marchio D., and Visier J. C. 2002. Influence of sensor position in building thermal control: Criteria for zone models. Energy and Buildings 34: 785-798.

Sowell, E. F., and Moshier M. A. 1995. HVAC component model libraries for equation-based solvers. Fourth International Conference on System Simulation in Buildings, IBPSA Building Simulation '95; Madison, Wisconsin.

Strand, R., Winkelmann F., Buhl F., Huang J., Liesen R., Pedersen C., Fisher D., Taylor R., Crawley D., and Lawrie L. 1999. Enhancing and extending the capabilities of the building heat balance simulation technique for use in energy analysis. Building Simulation 1999. International Building Performance Simulation Association; Kyoto, Japan.

Strand, R. K., Pedersen C. O., and Crawley D. B. 2001. Modularization and simulation techniques for heat balance based energy and load calculation programs: The experience of the ASHRAE loads toolkit and EnergyPlus. Building Simulation

2001. International Building Performance Simulation Association; Rio de Janeiro, Brazil.

Tashtoush, B., Molhim M., and Al-Rousan M. 2005. Dynamic model of an HVAC system for control analysis. Energy and Buildings 30: 1729-1745.

Vedavarz, A., Kumar S., and Hussain M. I. 2007. HVAC: Handbook of Heating, Ventilation and Air Conditioning for Design and Implementation. Industrial Press, Inc., New York, NY.

Walton, G. N., and Dols W. S. 2005. CONTAM 2.4b users guide and program documentation. National Institute of Standards and Technology. October 2005; Gaithersburg, MD

Wang, Y. W., Cai W. J., Soh Y. C., Li S. J., Lu L., and Xie L. H. 2004. A simplified modeling of cooling coils for control and optimization of HVAC systems. Energy Conversion and Management 45(18-19): 2915-2930.

Ward-Smith, A. J. 1980. Internal Fluid Flow: The Fluid Dynamics of Flow in Pipes and Ducts. Oxford University Press, New York, NY.

Webb, R. L. 1984. A unified theoretical treatment for thermal analysis of cooling towers, evaporative condensers, and fluid coolers. ASHRAE Transactions 90(2): 398-414.

Westerhof, N., Bosman F., Vries C. J. d., and Noordergraaf A. 1968. Analog studies of the human systemic arterial tree. Journal of Biomechanics 2: 121-143.

Wilson, F. W., Harvey P. D., and Jr. C. B. G. 1965. Pneumatic Controls for Industrial Application. American Society of Tool and Manufacturing Engineers, Dearborn, MI.

Xu, P., Haves P., and Curtil D. 2006. A library of HVAC component models for use in automated diagnostics.

York, D. A., and Cappiello C. C. 1981. DOE-2 engineers manual (Version 2.1A). Lawrence Berkeley National Laboratory; Berkeley, CA

Yu, X., Wen J., and Smith T. F. 2005. A model for the dynamic response of a cooling coil. Energy and Buildings 37: 1278-1289.

VITA

Name:	Clinton Paul Davis
Address:	Department of Electrical and Computer Engineering, Texas A&M University, 214 Zachry Engineering Center, TAMU 3128 College Station, Texas 77843-3128 c/o Charles Culp or Prasad Enjeti
Email Address:	cdoftx@gmail.com
Education:	B.S., Computer Engineering, Texas A&M University, 2006