DEVELOPMENT OF A DETAILED

HEAT PUMP SIMULATION

CONTROLS TESTBED

By

RACHEL SPITLER

Bachelor of Science in Mechanical Engineering

Oklahoma State University

Stillwater, Oklahoma

2012

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2019

DEVELOPMENT OF A DETAILED

HEAT PUMP SIMULATION

CONTROLS TESTBED

Thesis Approved:

Dr. Daniel Fisher

Thesis Adviser

Dr. Craig Bradshaw

Dr. Christian Bach

ACKNOWLEDGEMENTS

Firstly, thank you to my adviser, Dr. Fisher. You gave me the initial opportunity to pursue this and you have the distinction of being the sole original committee member remaining. Congratulations on graduating your last student, and thank you for seeing this through with me.

Thank you to my other committee members, Dr. Bach and Dr. Bradshaw. Thank you both for your willingness to join this process (even on very short notice!) and for your comments and advice. I greatly appreciate your service on the committee.

This work was partially funded by OCAST; I am thankful for their financial support.

Thank you to the denizens of 303 and 316 in those first couple of years – you all made it a joy to be working on this degree during that time. My thanks especially to Matt, who stayed the whole time; thank you for the advice, commiseration, and computer restarts – congratulations on graduating!

Thank you to my employer and coworkers for their understanding and encouragement, especially this year. I know we are all happy that my full attention can return to work!

Thank you to my colleagues, friends, and family members – near and far – who encouraged and supported me in many ways, including listening ears, food, and prayer. When I started, I never anticipated how many people around the world would take such a strong interest in my graduate degree.

Thank you to my immediate family for their support. My sisters and their families encouraged in many ways. Thank you especially to my parents: to my mother, for all the stepping stones that led to here, for the encouragement, prayers, food, and everything else she did to help me be able to finish. To my father, for starting me on this journey into both graduate school and this field, for acting as a "fourth committee member" and unofficial adviser, for praying, and for reminding me of the realities of graduate life.

Lastly, and most importantly, I must acknowledge my utter dependence on God in all matters, but especially in these past years. He has used graduate school (and many other circumstances) to sharpen and humble me in many ways that I am still learning to understand. It is beyond words to say how thankful I am that He who has begun a good work in me will finish it – and that He has allowed His faithfulness to show especially in the finishing of this work.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: RACHEL SPITLER

Date of Degree: JULY, 2019

Title of Study: DEVELOPMENT OF A DETAILED HEAT PUMP SIMULATION CONTROLS TESTBED

Major Field: MECHANICAL ENGINEERING

Abstract: In order to increase energy efficiency, it is desirable to reduce power consumption of HVAC systems. One way to reduce power consumption is through careful control of system components. To this end, this work covers the theoretical development of a detailed heat pump testbed for controls. The testbed was created by coupling a whole building simulation model, a detailed heat pump simulation model, an optimization engine, and two data exchange managers. Three component variables were investigated to reduce power: compressor refrigerant flow rate, condenser fan air flow rate, and evaporator air fan flow rate. It was seen that the most impactful variable on capacity and power was the compressor ratio, with the evaporator air fan flow rate having the second greatest impact, and the condenser air fan flow rate the least impact. In addition, an optimizing control that minimizes energy consumption and uses a penalty function to maintain space temperature was added to the testbed. Finally, a number of incremental test cases provide proof of concept for this testbed.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

Energy efficiency has become a major priority for countries across the world, and there has been a greater emphasis on the production and use of energy in recent years. In the US, 9% of 2018's total electricity consumption was used for cooling commercial and residential buildings, with 15% of residential electricity use directed for cooling (USEIA, 2019). Naturally, the reduction of energy use devoted to Heating, Ventilation, and Air Conditioning (HVAC) purposes is sought.

For years, HVAC equipment has become more efficient with monitoring and control being more widely implemented in building systems. As part of the increased drive towards efficiency, variable-speed and variable-flow heat pump components have been developed. Therefore, efficient components are readily available but need to be properly selected and used with appropriate control algorithms. The controls testbed presented in this thesis is an application of the developed monitoring and control capabilities to individual heat pump components. Heat pump components (such as variable-speed fans or variable-flow compressors) can be controlled individually to increase system capacity and efficiency, as opposed to controls applied to a heat pump as a whole. This allows more freedom for heat pumps to both meet the system load and reduce energy use. To implement this plan, efficient controls that meet system requirements need to be determined. Currently, control strategies that take advantage of component efficiencies and variability are crude and unoptimized.

Before control algorithms are implemented, they should be tested for feasibility. Experimental parametric studies of control strategies would be time-intensive and cost-prohibitive due to the number of different conditions required. Too many buildings and environmental conditions would be required to reasonably run full testing, even if they were easily available to test. In addition, any errors in the control algorithm would occur on a physical system with equipment that could be damaged. Less time and money are required to run a simulation of a whole building and associated equipment than to run experimental testing control systems. The testbed developed in this work shows the possibilities of this type of testbed.

One location where component control strategies have a place in the heat pump system is digital scroll compressors. Digital scroll compressors operate over very short time steps, turning on and off again within a matter of seconds. As such, any modeling of the compressor over a large timestep (such as an hour or fifteen minutes – both standard building simulation timestep lengths) loses the intricacy and benefits of the digital scroll. Therefore, a detailed heat pump model with an easily interchangeable compressor model was necessary.

Variable-speed fans also allow for controlled efficiencies. The motors for these fans can operate from low part-load to full-load capacity, which allows for control of cooling through variation of condenser and evaporator air flow rate. A variable air flow rate also had to be accounted for in the heat pump model.

The effect of these component controls on energy use and equipment performance required an investigation of their role in a whole building. In addition, a long simulation period and associated environmental conditions had to be accounted for in the model for completeness. A detailed building model including these elements had to be used for the testbed.

In addition, in order to control the variable-speed fans and digital scroll compressor, a method to vary their changing parameters was required. An optimization engine was therefore necessary for the testbed.

Thus, for a feasible testbed, several things are required: an optimizable heat pump model, a detailed building model, and an optimization engine to implement control algorithms. The heat pump model required the capability to model detailed and variable components. The building model had to allow for a realistic simulation of a given system at a given location. The optimization engine required the ability to read outputs and adjust inputs to the heat pump model for optimization of the equipment components.

The heat pump model chosen, HPSim, is a detailed vapor-compression cycle model with interchangeable component models. These component models allow for variation in air and refrigerant flow. HPSim is a steady state model operating on an instantaneous timestep.

The whole building model chosen for this work, EnergyPlus, models the indoor and outdoor environmental conditions of a specified building at a chosen location. With EnergyPlus and HPSim coupled, the entire system was modeled in detail. However, the control algorithms also needed to be included for the testbed; EnergyPlus thus called an optimization engine, using an intercessory data exchange manager.

The call to the chosen optimization engine, GenOpt, was placed deep inside the EnergyPlus model structure, in the same space the coil routine would be located in a calculation for a curve-fit heat pump model. GenOpt called HPSim, via a second data exchange manager, adjusting assigned variables until HPSim's energy use was minimized but the system load met. Finally, the optimized case outputs were reported back to EnergyPlus.

Creation and demonstration of the simulation testbed was the single most significant and unique contribution of this work. An optimization testbed coupling a detailed thermodynamic heat pump cycle simulation with a detailed building heat transfer simulation inside the same timestep had not been previously created. Three technical challenges were overcome to create the testbed:

1. The HPSim heat pump simulation was modified to achieve compatibility with EnergyPlus.
2. A coupling methodology was developed to achieve convergence in the testbed environment.
3. Optimization routines were created to achieve control of the heat pump simulation in the context of the building heat transfer simulation.

The following sections provide additional detail on the development of the simulation testbed. Chapter 2 reviews the literature in this area of study, and Chapter 3 describes an overview of the HPSim program and the modifications made for coupling. Chapter 4 briefly reviews EnergyPlus and discusses the overall testbed coupling methodology. Chapter 5 delves into the optimization routines, while Chapter 6 covers testbed verification. Finally, Chapter 7 gives a summary of results, project conclusions, and future work to be considered.

CHAPTER II

LITERATURE REVIEW

This chapter reviews literature on building simulation coupling environments and methodologies. Explanatory literature for the existing programs (EnergyPlus, HPSim, and GenOpt) is discussed. In addition, some of the foundational work for this project is reviewed.

2.1 Previously Developed Building System Testbeds

There are two known existing coupling environments for building simulation but neither have coupled a detailed heat pump simulation model with a detailed building heat transfer simulation. The most prominent existing testbed is the Building Controls Virtual Test Bed (BCVTB). BCVTB was developed by Lawrence Berkley National Lab as a coupling environment for simulation programs (LBNL, 2016a). It serves as a building energy analysis program, managing multiple simulation programs coupled through middleware (Wetter, 2011). The primary focus of the program is to assess and implement building controls in a simulation environment (LBNL, 2016a). This was achieved through co-simulation and loose-coupling (coupling outside of a simulation timestep but within the simulation runtime) building, controls, and equipment models, as well as physical hardware (LBNL, 2016a). BCVTB allows easy linking of a number of models including EnergyPlus and GenOpt (LBNL, 2016a).

The middleware for BCVTB allows coupled programs to start simulations at the same point in the run; BCVTB synchronizes between programs at a fixed timestep and without iteration (Wetter, 2011). Using the BCVTB to reset state variables in EnergyPlus is challenging due to the code structure and volume; BCVTB can only reset variables in the zone timestep (Wetter, 2011). For this work, in order to iterate between the building and heat pump models, EnergyPlus variables needed to be reset during calculations – that is, in the system timestep. In order to control the individual components of a heat pump model while running a full building simulation, a more tightly-coupled methodology was required than BCVTB could provide.

The other known thermal systems testbed program is Virtual Cybernetic Building Testbed, or VCBT. VCBT is a testbed developed by the National Institute of Standards and Technology (NIST) (NIST, 2017). VCBT simulates in real-time and runs on several physical computers in a NIST laboratory (Bushby et al., 2001). Its focus is on fault detection and diagnosis, not controls viability.

2.2 Previous Investigation of Coupling Methodologies

Based on literature reviewed, strong-coupling was chosen as the modeling methodology for this work. Strong-coupling is the iteration of coupled models inside a simulation timestep (Trčka et al., 2009). Loose-coupled models undergo no iteration, but use information from the previous timestep as inputs (Trčka et al., 2009). In one work (Trčka et al., 2010), it was concluded that a strongly-coupled model had greater accuracy than a loosely-coupled model over the same timestep. It was additionally noted (Trčka et al., 2009) that loose-coupling tended to result in oscillations in calculations; however, over larger timesteps (15 minutes or greater), strong-coupling could result in convergence issues if the iteration accuracy was held too tightly. This was solved by allowing a looser iteration accuracy.

In Trčka et al. (2005), three reasons were listed when coupling during the program simulation – as opposed to coupled models sharing information outside of program simulation timesteps – may be logically used. These reasons were when one modeling program does not contain all aspects of a system to be modeled, when an available model cannot represent the system accurately enough for the test, and when a real system needs to be modeled (Trčka et al., 2005). It is the second of these that has led to the coupled models presented in this thesis.

2.3 Selection of the Primary Programs

HPSim, EnergyPlus, and GenOpt were chosen as the three primary programs for the testbed. A discussion of each follows.

2.3.1 Selection of the Heat Pump Model

HPSim was chosen as the heat pump model for this testbed. HPSim was based on an ORNL solver (Fischer et al., 1988); it was developed by Iu (2007) and expanded by Padhmanabhan (2011). HPSim has detailed component models of the four main components in a heat pump (compressor, condenser, expansion device, and evaporator) as well as models of three secondary elements (accumulator, filter-drier, and distributor). This program can be used as a design tool to calculate expansion device size and refrigerant charge when run in design mode or run in simulation mode to determine the superheat for a piece of specific equipment. HPSim was experimentally validated by both Iu (2007) and Padhmanabhan (2011); as such, the modifications made to HPSim for testbed implementation did not require extensive validation. HPSim source code was also readily available, being already in use by the Oklahoma State University research team. This was an important consideration because source code changes would be required to couple the two programs.

2.3.2 Selection of the Whole Building Model

EnergyPlus uses a weather format for which data has been collected for over 2,100 locations worldwide, allowing for a reasonable determination of building loads in a number of global settings (EnergyPlus, 2013). It is also open-source and has a modular program structure. The modularity allows for easy expansion by developers, particularly intended to encourage development and inclusion of additional modules by multiple programmers (Crawley et al., 2000). Although there are hundreds of building models in existence (Crawley et al., 2008), EnergyPlus was swiftly chosen due to it being well validated, available, and modular in structure. Once again, the source code was freely available to edit and use, allowing program coupling.

2.3.3 Selection of the Optimization Engine

In order to be able to control the heat pump inside the timestep, an optimization engine was necessary. GenOpt was specifically created with a view to coupling whole building simulation programs with other compatible programs. It was therefore the most likely candidate for the optimization engine. GenOpt is open-source and capable of optimizing multiple variables at once. GenOpt was developed by LBNL to be a generic optimization program (Wetter, 2001). GenOpt operates by minimizing an objective function through adjustment of input variables bounded by user-constraints. GenOpt requires an external simulation model to calculate the objective function; input files for the external models are built by GenOpt with the adjusted variables (Wetter, 2001). The optimization algorithm used in GenOpt is user-set but recommended algorithms (or classes of algorithms) are given in the user's manual and documentation based on the problem type being solved. For this work, the GPS Hooke Jeeves optimization algorithm was used. Further explanation of GenOpt's mechanics are available in Wetter (2001) and on the program website (LBNL, 2016b).

One previous coupling of GenOpt and EnergyPlus was the optimization of a hybrid Ground-Source Heat Pump (GSHP) system (Padhmanabhan, 2005). The effect of optimizing the hybrid GSHP was studied over a twenty-year simulation period (Padhmanabhan, 2005). Because only the final system effects were of interest, it was possible to use a larger time step for the first nineteen years and a smaller time step for the twentieth and final year; this also allowed for optimization of the whole building simulation by GenOpt. The testbed presented in this work was concerned with a much smaller time period (sub-hourly) and with the instantaneous effects of the system. This concern drove the coupling methodology, preventing the optimization of the testbed simulation as a whole.

2.4 Other Considerations

Work by John Gall provided particular insight into the controllability of the HPSim cycle simulation, illustrating the anticipated challenges of creating the proposed testbed. Gall (2015) addressed controls modelling and dynamic optimization. Gall's test case was a single-zone variable-air volume (VAV) system with air conditioner. The evaporator fan speed modulated between completely off and completely on; for modeling purposes, the minimum fan speed was set to 30% on the assumption there would always be some airflow in the VAV system. The goal of modulation was an acceptable thermal environment in the room; a temperature sensor was located in the room model to monitor this condition. As the room temperature rose above the setpoint temperature, more cooling was required, calling for an increase in airflow. However, logically, if the evaporator fan was already running at 100% when the system needed more cooling, another component would have to change for the setpoint conditions to be met.

This led to modulating the compressor from completely-off to completely-on. The compressor was controlled based on the temperature of the air supplied to the space. If the supply air temperature was greater than the setpoint, the compressor was modulated by the control to increase cooling capacity. The compressor received no feedback from the room sensor to ensure that the room temperature setpoint was met; it only controlled on the supply air temperature. The stated controls setup could result in a situation where the evaporator fan was fully on and the supply air temperature met the setpoint, but the room temperature did not meet its setpoint. Gall's solution to this problem was to include a feedback controller that adjusted the supply air temperature setpoint if the room setpoint was not met. This changed the modulation of the compressor without affecting the evaporator fan speed; the limiting element in this case was the full capacity of the compressor.

In addition, the condenser fan speed was the third element available to adjust for optimal control. Gall did not specifically optimize the condenser fan speed but let it vary proportional to the compressor modulation. For situations where dehumidification drove the system, the leaving air temperature sensor could instead measure the leaving air dew point temperature. Additionally, the feedback controller measured the room dew point temperature and correspondingly modulated the compressor.

Gall also raised the issue of how long it could take a system to optimize. If a system could quickly optimize, then the controller might be a good one to use. However, if it took a substantial time to optimize – longer than an hour – its use would be harder to justify because both external and internal conditions could change significantly over longer periods. This would not be a major concern in every case, but it is a common enough concern that it must be considered.

Gall also considered a case where room air humidity was a driver of thermal comfort as opposed to room air temperature solely. Based on simulation, Gall concluded that controlling on the supply air conditions was only reasonable for systems unconcerned with humidity or in cases where the humidity of the room was within an acceptable range due to high static pressure. Gall explained that the optimized supply air temperature tended to be lower if the evaporator fan had a higher static pressure on it. However, fan power was generally smaller on an evaporator fan operating at a lower static pressure and almost negligible when compared with the compressor power. For the sake of simplicity, humidity concerns were neglected for the testbed discussed in this work.

CHAPTER III

MODIFYING THE STANDALONE HEAT PUMP MODEL FOR COUPLING WITH ENERGYPLUS

3.1 Overview of the Heat Pump Simulation Program

For close-coupling with EnergyPlus in a controls testbed environment, HPSim was previously identified as the best detailed system model. It is a steady-state vapor-compression (or heat pump cycle) component model with detailed circuiting and frosting modeling capabilities. HPSim has detailed component models of the four main components in a heat pump (compressor, condenser, expansion device, and evaporator) as well as models of three secondary elements (accumulator, filter-drier, and distributor). This program can be used as a design tool to calculate refrigerant charge and expansion device size when run in design mode or run in simulation mode to determine the superheat (the degrees of temperature the refrigerant vapor temperature is above the saturated vapor temperature) for a piece of specific equipment.

Although, in theory, the building heat transfer models of EnergyPlus and the system component models of HPSim are compatible, the basic assumptions of each program make coupling difficult. EnergyPlus assumes that heat pump performance is limited only by the capacity of the indoor coil (the evaporator coil), which is modeled as a function of air side conditions (temperature and humidity ratio) only. EnergyPlus does not consider any of the actual refrigerant side operating conditions of the heat pump, including:

1. The requirement for superheated vapor at the compressor inlet (evaporator outlet)

2. The requirement for subcooled liquid at the condenser outlet (expansion device inlet)
3. The requirement for two-phase liquid and gas mixture at the evaporator inlet (expansion device outlet)

Likewise, HPSim considers all airside conditions to be constant at the coil inlets. HPSim also considers the refrigerant compressor flow ratio to be constant.

3.1.1 Heat Pump Cycle

The heat pump cycle consists of four main components: a compressor, a condenser, an expansion device, and an evaporator. Three more secondary components can be added to the cycle: a distributor, a suction line accumulator, and a liquid line filter-drier. A diagram of this cycle is shown in Figure 3.1.

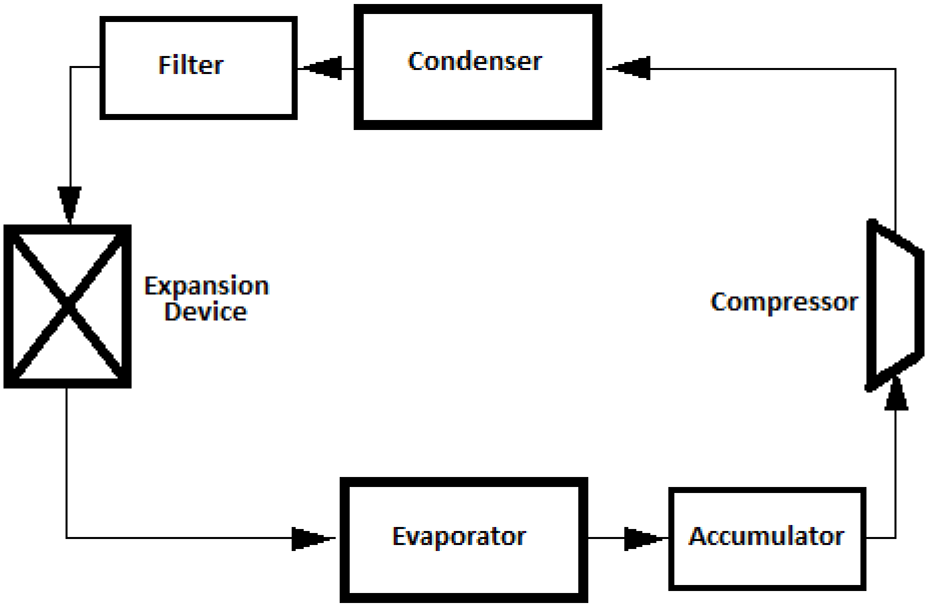


Figure 3.1 Simple Heat Pump Cycle

For explanation purposes, an ideal vapor compression cycle is discussed, with the cycle starting at the compressor and moving counterclockwise. The refrigerant enters the compressor in the superheated vapor state and is compressed to a higher temperature and pressure at constant entropy. After leaving the compressor, the superheated refrigerant enters the condenser, where it is cooled at constant pressure to a compressed liquid. The refrigerant leaves the condenser and enters the expansion device, where it is adiabatically expanded into a two-phase liquid and gas mixture. From there it enters the evaporator, absorbing heat at constant pressure until it reaches the superheated state at the cycle's starting point. The suction line accumulator and the liquid line filter-drier ensure that the heat pump operates successfully in actuality. The filter-drier is located between the condenser outlet and the expansion inlet, and the accumulator is located after the evaporator outlet and before the compressor inlet. In addition, if liquid line and suction line lengths are given in HPSim's input file, those components are modeled, addressing losses in refrigerant pressure and enthalpy. The liquid line is between the condenser and the expansion device, and the suction line is between the evaporator outlet and compressor inlet. The distributor is located at the entrance to the evaporator and modeled in the capillary and short tube models; the distributor model impacts refrigerant system charge and pressure drop over the capillary and short tube expansion devices.

3.1.2 Development of the HPSim Program

HPSim was created using a program solver developed by Oak Ridge National Laboratory (Fischer et al., 1988); a complex circuiting model for the coil heat exchangers was developed (Iu, 2007). Later, quasi-steady state frosting modeling capabilities were added to the program (Padhmanabhan, 2011). The system components are simulated using successive substitution (Padhmanabhan, 2011). The model was experimentally validated by Iu (2007) for thirty-one different cases, both on a component level and on an overall system level. For additional information on the heat pump model, see Iu (2007). This model was the basis for frost studies of microchannels, which included additional validation (Padhmanabhan, 2011).

The refrigerant and air property sets for the program were generated using NIST's RefProp program, and then formatted into tables (Iu, 2007). The code uses a lookup routine to retrieve tabulated property information, linearly interpolating as needed; this method is faster and more robust than using the RefProp routine in the simulation directly (Iu, 2007).

The air properties for HPSim as a standalone model are pre-set by the input file as constant values. The testbed needed adjustable inlet temperatures that could be impacted by a changing air flow rate and a changing heat pump cycle. Therefore, the method of setting air properties in HPSim had to be changed for testbed coupling – this is covered in Section 3.4.6.

HPSim runs in four modes: two simulation modes and two design modes. Each set of modes is differentiated by the known information entering HPSim and the information calculated by HPSim. Further explanation is contained in Section 3.2.1 and 3.2.2.

3.1.3 Solution Methodology and Assumptions

The vapor compression solution methodology first converges on the high pressure side saturation temperature while holding the low pressure side state variables constant as shown in Figure 3.2. The high side pressure is adjusted through modification of the compressor outlet saturation temperature ($T_{ref_{comp\ out\ sat}}$). Once the high side of the loop has converged, the low pressure side iterates to convergence by adjusting the evaporator air inlet temperature, also shown in Figure 3.2. If the evaporator air inlet temperature has not returned to the setpoint, the overall cycle is considered unconverged and control is passed back to the high side solver.

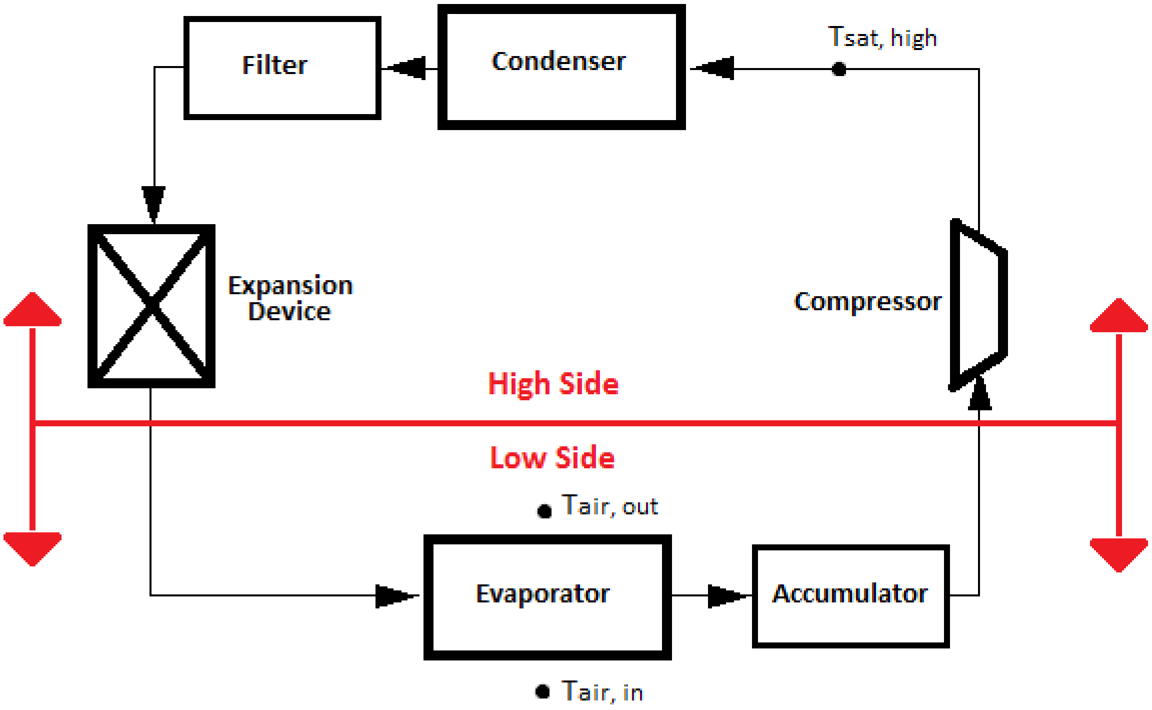


Figure 3.2 Heat Pump Cycle Showing High and Low Pressure Components

The high and low pressure sides are simulated individually, with refrigerant state information being passed to the low pressure side from the high pressure side after it converges on a solution. Once the low pressure side has converged on a solution, the overall cycle is checked for convergence. If the cycle has not converged, the process starts over with simulation of the high pressure side. Figure 3.3 shows the program flow for the high pressure side.

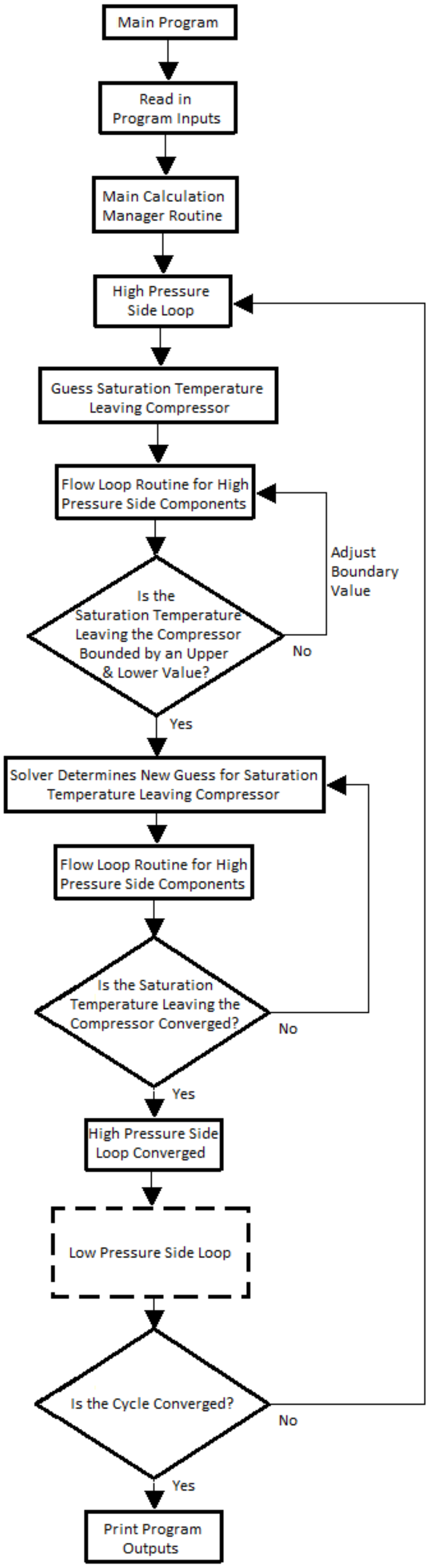

Main Program
Read in Program Inputs
Main Calculation Manager Routine
High Pressure Side Loop
Guess Saturation Temperature Leaving Compressor
Flow Loop Routine for High Pressure Side Components
Is the Saturation Temperature Leaving the Compressor Bounded by an Upper & Lower Value?
No
Adjust Boundary Value
Yes
Solver Determines New Guess for Saturation Temperature Leaving Compressor
Flow Loop Routine for High Pressure Side Components
Is the Saturation Temperature Leaving the Compressor Converged?
No
Yes
High Pressure Side Loop Converged
Low Pressure Side Loop
Is the Cycle Converged?
No
Yes
Print Program Outputs

Figure 3.3 HPSim High Pressure Side Loop and Overall Program Flow

HPSim's main program routine first reads in the inputs and then starts the main calculation routine, which calls the solver for both the high side and low side loops. The high side routine calls each of the components on its side: the compressor, condenser, and – if in simulation modes – the expansion device. The component routines interact with HPSim's solver to bound guesses for the adjusted variable – $T_{ref_{comp\ out\ sat}}$ – and then converge on either refrigerant mass flow rate or subcooling, depending on the mode. HPSim calculates the compressor outlet pressure – the high side pressure – based on the adjusted $T_{ref_{comp\ out\ sat}}$ and a quality of 1. Once the high side has converged, the low pressure side loop is simulated; Figure 3.4 shows the low side program flow.

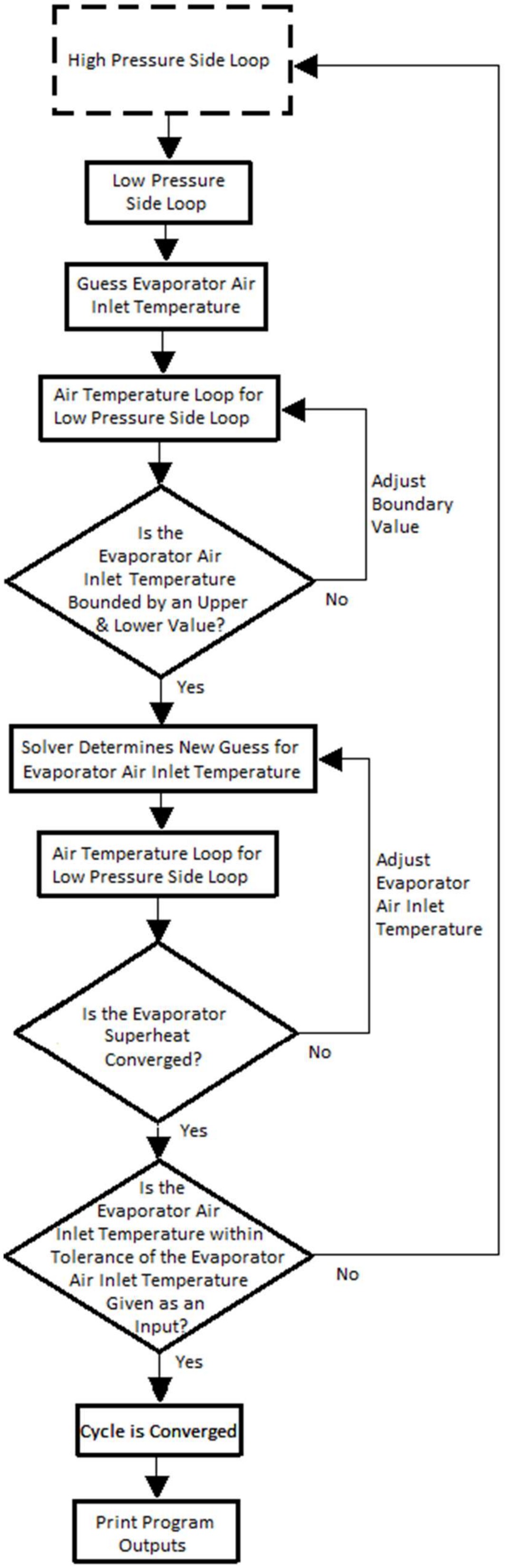

High Pressure Side Loop
Low Pressure Side Loop
Guess Evaporator Air Inlet Temperature
Air Temperature Loop for Low Pressure Side Loop
Is the Evaporator Air Inlet Temperature Bounded by an Upper & Lower Value?
No
Adjust Boundary Value
Yes
Solver Determines New Guess for Evaporator Air Inlet Temperature
Air Temperature Loop for Low Pressure Side Loop
Is the Evaporator Superheat Converged?
No
Adjust Evaporator Air Inlet Temperature
Yes
Is the Evaporator Air Inlet Temperature within Tolerance of the Evaporator Air Inlet Temperature Given as an Input?
No
Yes
Cycle is Converged
Print Program Outputs

Figure 3.4 HPSim Low Pressure Side Loop and Overall Program Flow

The low side loop always calls the evaporator routine; if HPSim is operating in the design modes, the low side also calls the expansion device. The low side loop and evaporator routines interact with the HPSim solver to bound guesses for the adjusted variable – the evaporator inlet air temperature ($T_{air_{evap\ in}}$) – and then converges on the evaporator superheat. Once the low side loop has finished iterating, the cycle is checked for convergence. The cycle convergence criteria is whether $T_{air_{evap\ in}}$ is within tolerance of the input file value. If convergence has been reached, the program finishes its run by printing the program outputs. If the cycle has not converged, the program adjusts the low side pressure and returns to the start of the high side loop to recommence calculations. The low side pressure is indirectly adjusted by the modifying the compressor inlet saturation temperature ($T_{ref_{comp\ in\ sat}}$) and thus the high side pressure; by changing the high side pressure, the low side pressure will also be adjusted through the high side calculations.

HPSim converges with a difference method. For the high side and low side loops, once the difference between the previous and current values meets the set convergence criteria the loop is considered sufficiently converged. For the cycle convergence, the difference between the current calculated value and the saved input value for the variable must be within a tolerance defined by the root solver. The root solver also determines the boundary conditions for the low and high side variable guesses. For more information, see Iu (2007), specifically Figures 2.3, 2.4, 3.9, and 3.10.

3.1.4 Root Solver

The root solver is a major program algorithm, iterating through both high side and low side loops and checking each for convergence. The HPSim root solver was a modification of the ZBRENT function by Press et al. (1996). This function uses the Van Wijngaarden-Dekker-Brent method, a combination between an open solving method and a bracketed solving method. Bisection, root bracketing, and inverse quadratic interpolation are all used in the method, which will converge as long as the starting evaluation interval contains a root. The solver uses a combination of set tolerances and an internally calculated tolerance to determine convergence. In addition to the main solver, there is also a root bracketing routine called at the start of each pressure loop; this determines the initial guesses for the boundary conditions.

These solvers and bracketing routines do not guarantee a converged refrigerant cycle; convergence is largely dependent on the refrigerant and air conditions entered as inputs. An unconverged solution can result from an improperly defined set of inputs, an error from which HPSim has no way to recover. Without a manager program to compensate, this would introduce unconverged or outdated results into the testbed, which would not only be incorrect but could also cause critical failures. Therefore, a manager program was introduced between GenOpt and HPSim; this is discussed in Section 4.8.

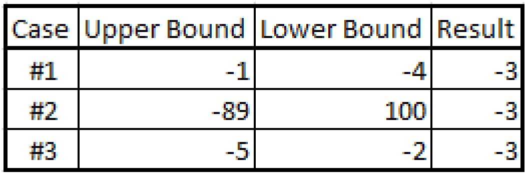
3.1.5 Root Solver Validation

The root solver routine was slightly modified to decrease calculation time; the machine precision was removed from the tolerance calculation, and a tolerance check was relocated from the end to the beginning of the solver routine. The solver was revalidated after it had been modified, using

$$F(x) = (x + 3)\ (x - 1)^2 \qquad (3.1)$$

for testing. The bounds were adjusted and starting guesses for the root were given for three different cases; these cases and their results are listed in Table 3.1.

Table 3.1 Test Cases of Equation 3.1



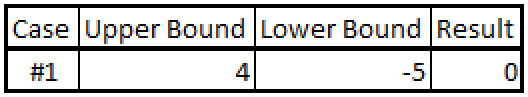
Case	Upper Bound	Lower Bound	Result
#1	-1	-4	-3
#2	-89	100	-3
#3	-5	-2	-3

In addition,

$$F(x) = (x)(x - 1)^2 \qquad (3.2)$$

was solved for a single case to check a different root could be found; the result is in Table 3.2.

Table 3.2 Test Case of Equation 3.2



Case	Upper Bound	Lower Bound	Result
#1	4	-5	0

Every case returned a correct root, even the cases where the upper and lower bound values were reversed. The number of iterations required for the solver to converge is dependent on the bounds, in particular how close they are to the root.

3.2 Heat Pump Simulation Solution Methodology and Testbed Coupling Considerations

HPSim's operating mode is set in an input file by the user in both standalone and testbed setups. As mentioned in Section 3.1.2, there are four different calculation modes: two simulation and two design modes. An overview of these modes, along with an explanation of the mode chosen for testbed coupling, follows.

3.2.1 Simulation Modes

Figure 3.5 shows the system logic for HPSim's simulation modes. Dashed lines indicate the second simulation mode, where the condenser subcooling is guessed instead of the evaporator superheat; the evaporator superheat is known.

Required simulation inputs include air-side boundary conditions, a physical description of the cycle components, refrigerant charge, and a defined refrigerant value. In one mode, the defined refrigerant value is the evaporator superheat; in the other, condenser subcooling (the degrees of temperature the refrigerant liquid is below the saturated vapor temperature) is the defined value. The system charge being defined in the input file is a distinctive for the simulation modes. After reading in the input file and setting initial properties, the solver simulates the compressor, condenser, and expansion device in that order, forming the high side loop.

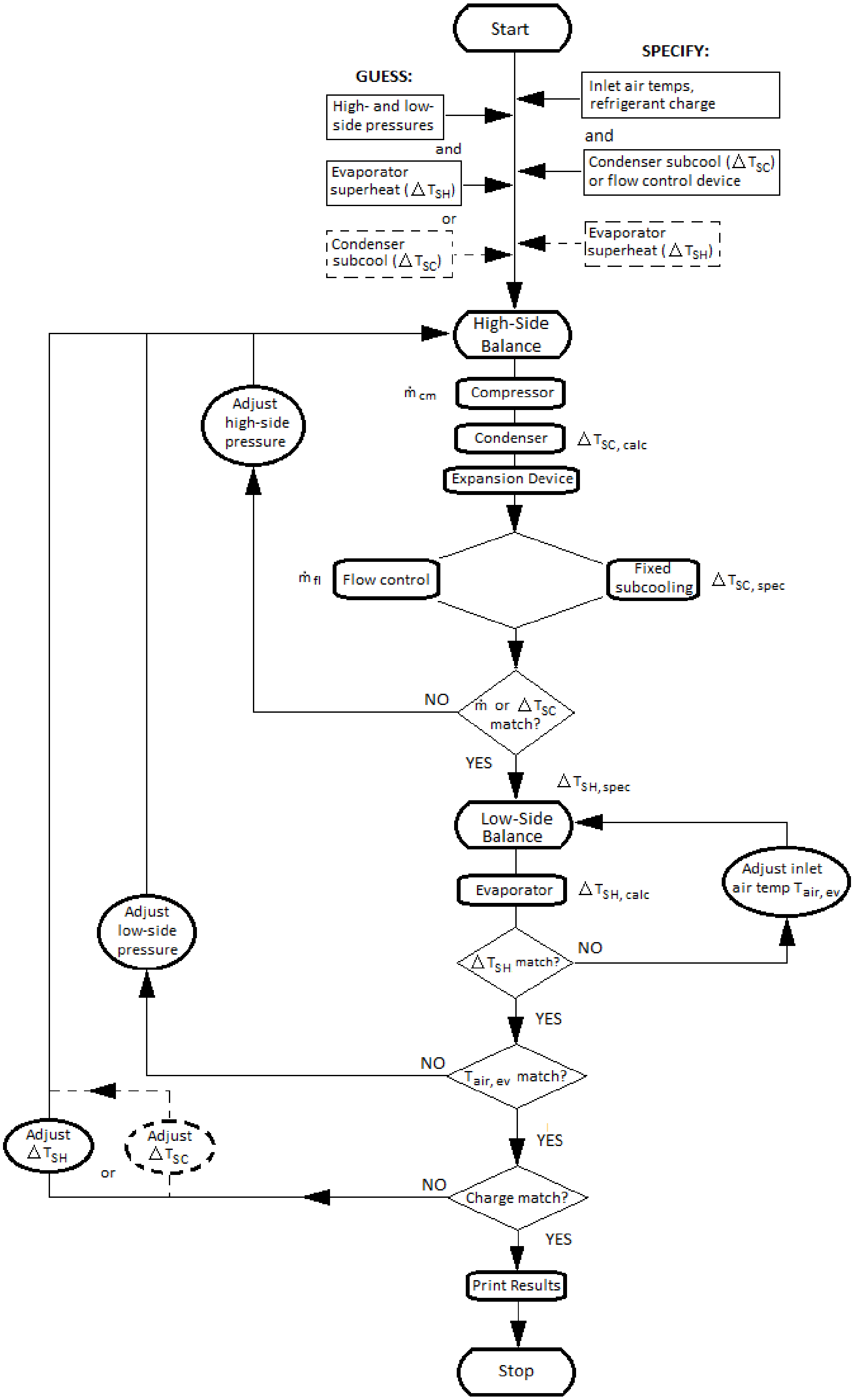


Figure 3.5 HPSim Program Flow - Simulation Mode (Fischer et al., 1988)

After the expansion device has been modeled, the solver checks for convergence on the refrigerant mass flow rate ($\dot{m}_{ref}$) or the condenser subcooling. If unconverged, the solver utilizes an updating algorithm to select the condensing temperature for the next iteration; the high side pressure is dependent on temperature. The compressor model updates the initial high side pressure guess calculated from the condensing temperature. For the mode where flow control is used, refrigerant charge is known and $\dot{m}_{ref}$ is the convergence check. If $\dot{m}_{ref}$ is within tolerance, then the loop has converged; a tolerance of 0.1 kg/s was set for $\dot{m}_{ref}$. For the mode where subcooling was known instead of refrigerant charge, condenser subcooling is checked for convergence.

After the high side has reached convergence, the low side loop – consisting solely of the evaporator model – is calculated. The low side loop is checked for convergence on evaporator outlet refrigerant superheat against a 0.1°F tolerance; if unconverged, $T_{air_{evap\ in}}$ is adjusted and the loop recalculated until superheat is converged. Once convergence on the low side loop has been reached, the overall system is checked for convergence on $T_{air_{evap\ in}}$; the tolerance for $T_{air_{evap\ in}}$ was set at 0.001°F. If convergence is not reached, $T_{ref_{comp\ in\ sat}}$is adjusted and the system iterates until it converges.

The evaporator outlet air temperature ($T_{air_{evap\ out}}$) is calculated based on the heat transferred across the evaporator coil and is not a control point. $T_{air_{evap\ in}}$ is both an adjusted value and a setpoint for convergence; the setpoint is a value from the input file. The evaporator superheat is calculated in the low side loop, and $T_{air_{evap\ in}}$ is adjusted inside the loop until calculated superheat is within tolerance. In the mode where superheat is known, the calculated superheat must be within tolerance of the specified superheat. In the mode where subcooling is the specified value, the calculated superheat theoretically be within tolerance of the previous iteration's superheat. Code behavior indicates that the tolerance is calculated against the input superheat still, though.

The user-defined setpoint is the starting value of $T_{air_{evap\ in}}$ to the low side loop. Once superheat has converged, the calculated $T_{air_{evap\ in}}$ is checked against the setpoint. If it is within tolerance, the overall system loop is considered converged and iterations are concluded. If $T_{air_{evap\ in}}$ is outside of tolerance, $T_{ref_{comp\ in\ sat}}$ is checked against the previous iteration's value. If $T_{ref_{comp\ in\ sat}}$ is within tolerance, the system will be considered converged. If $T_{ref_{comp\ in\ sat}}$ has not converged, the program will return to the high side loop and run again with the latest guess for $T_{ref_{comp\ in\ sat}}$. As such, the system conditions are satisfied by $T_{air_{evap\ in}}$ being within tolerance of the setpoint, or by $T_{ref_{comp\ in\ sat}}$ converging. The $T_{ref_{comp\ in\ sat}}$ convergence check is a deviation from code design, but is an existing deviation. A secondary deviation is that the final convergence check for the refrigerant charge does not exist in the version of HPSim used. This removes the final loop of Figure 3.5, the adjustment of superheat if subcooling is specified or of subcooling if superheat is specified. The nonexistent final convergence check also removes some of the benefit of operating in simulation modes.

3.2.2 Design Modes

The main distinctive of the design modes is that refrigerant charge is calculated by HPSim instead of being an user-defined value. The design modes also calculate the required size of the expansion device. Three different types of expansion device are sized: short tube, TXV, and capillary tube. A less significant difference is the expansion device modeling location in the system calculation. The expansion device – although it comes before the evaporator inlet in the cycle – is modeled after the evaporator in the design modes. Therefore, the expansion device is simulated outside the high loop when the HPSim is run in either design mode. This allows the expansion device to be sized for the system. Figure 3.6 shows the system logic for HPSim in design modes; the difference in design modes is whether the condenser subcooling or flow control device is specified.

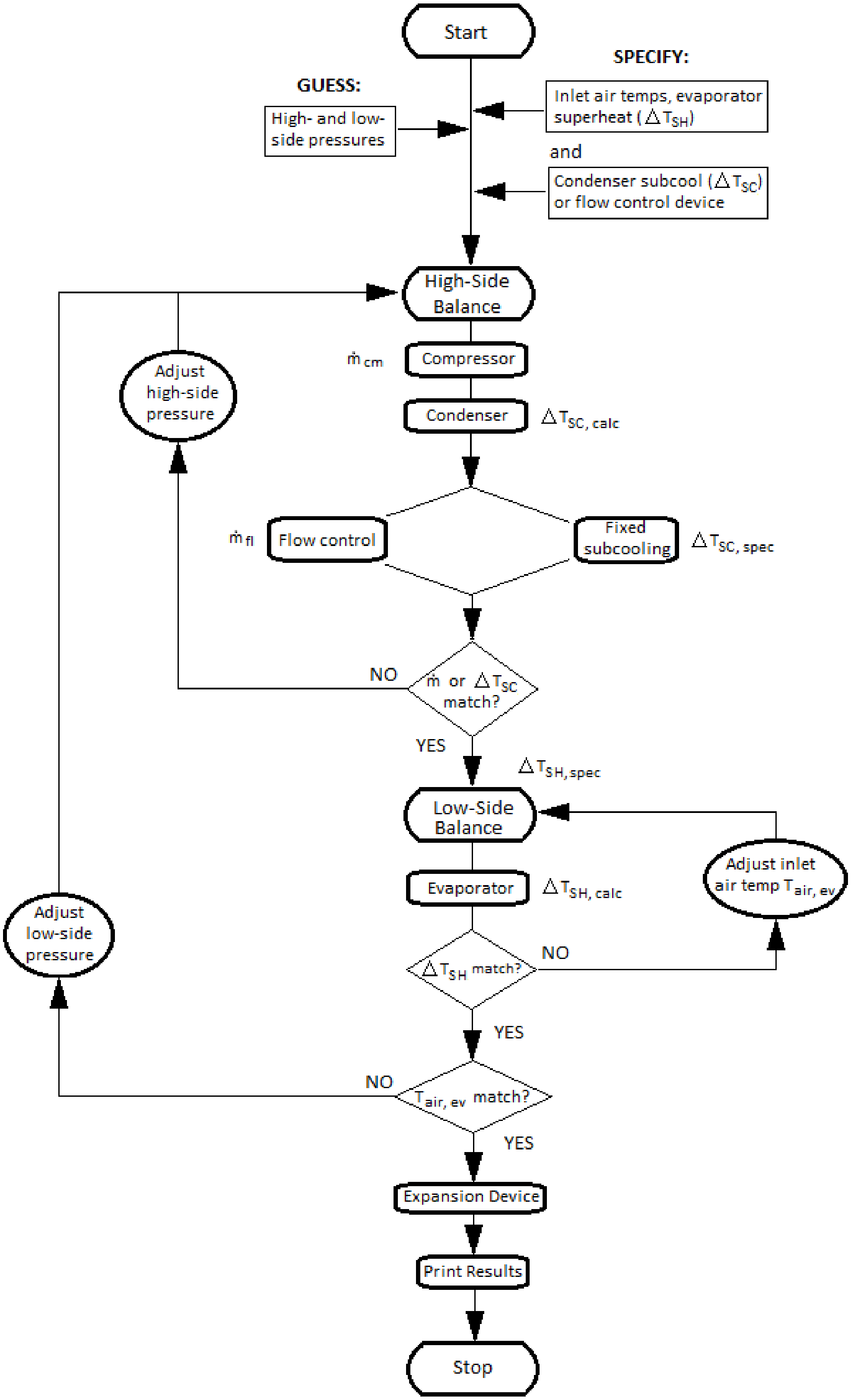


Figure 3.6 HPSim Program Flow - Design Mode (Fischer et al., 1988)

The air-side boundary conditions, a physical description of the cycle components, and evaporator superheat are required inputs for both design modes. In one mode, the condenser subcooling temperature is required, while the flow control device is required in the other. After reading in the input file and setting initial properties, the solver simulates the high side loop – here comprised of the compressor and condenser. If the mode has a specified subcooling, the solver checks for convergence on the subcooled temperature; the tolerance is set to 0.1°F. If the mode has a specified expansion device, the mass flow rate is checked for convergence. After reaching convergence on the high side loop, the evaporator – the only component in the low side loop – is calculated. Superheat is checked at the evaporator outlet; if the defined superheat tolerance (0.001 °F) has been reached, the overall system is checked for convergence. If the defined superheat tolerance has not been reached, $T_{air_{evap\ in}}$ is adjusted and the loop recalculated.

Once convergence on the low side loop has been reached, the overall system will be checked for convergence on $T_{air_{evap\ in}}$. In a deviation from code design, if $T_{air_{evap\ in}}$ has not converged, $T_{ref_{comp\ in}}$ is secondarily checked for convergence and then adjusted if the system has not converged. Once the system has converged, the expansion devices are sized and the overall system charge calculated, concluding the run for the design modes. The calculation of refrigerant charge and expansion device size after simulation convergence occurs means these components do not impact the state variable calculations. The expansion device models and charge calculations were not thoroughly investigated for this project.

3.2.3 Selection of HPSim Design Mode for Testbed Implementation

In order for the simulation modes to operate correctly, the refrigerant charge of the system must already be known. HPSim is designed to converge on refrigerant charge while running in either simulation mode. In order to run in either simulation mode for the testbed as designed, an accurate refrigerant charge would be required for every heat pump modeled. In addition, either the degrees of subcooling at the condenser outlet, the expansion device, or the degrees of superheat leaving the evaporator would need to be an input. According to code design, if subcooling is known, superheat could be calculated. However, this simulation mode does not behave as designed, as noted in Section 3.2.1. Also noted in Section 3.2.1 is that, despite the code design, the simulation modes' code behavior did not complete a refrigerant charge convergence check. This essentially removes most of the benefit of operating in simulation modes.

The system charge is not a required input for HPSim when it runs in either design mode. Instead, the superheat and subcooling or expansion device are required for use; the first two are more closely linked to refrigerant cycle operation than to specific pieces of equipment. However, in order to operate the testbed with a fixed heat pump model, a specified flow control device – or expansion device – would be needed.

All four simulation and design modes can model an accumulator, allowing for adjustments in refrigerant charge – this was important due to the modulating compressor ratio the testbed uses. All four modes struggle to converge on superheat, regardless of if superheat is set or calculated. It is worthwhile to note that the predecessor program to HPSim – and the originator of HPSim's solver – also had problems converging on evaporator superheat (Fischer et al., 1988). There seemed to be little benefit in constraining the system by including another input value when it would result little change in calculation; therefore, the simulation modes were not used. Due to this reason and others stated above, the design mode with a specified control device was chosen for the testbed.

3.3 Integration of the HPSim Refrigerant Side Models in the Simulation Testbed

One of the benefits of HPSim was the detailed refrigerant side modeling, which EnergyPlus lacked. This section provides an overview of the HPSim refrigerant side models.

3.3.1 Overview of the HPSim Refrigerant Side Simulation

The different system components are modeled in individual routines, allowing the refrigerant properties to be connected at the inlets and outlets of the routines. Every component but the expansion device uses the inlet properties and parameters to calculate the rate of heat transfer for the component. A diagram of the component information flow is shown below in Figure 3.7.

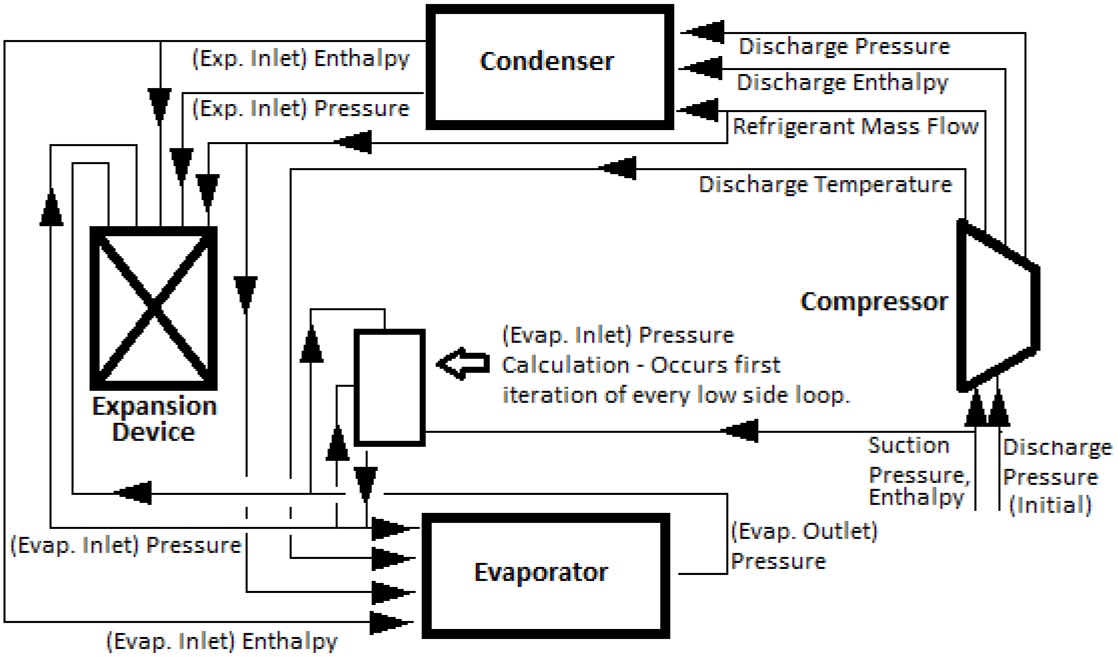


Figure 3.7 HPSim Information Flow Diagram

Only two independent refrigerant state variables need to be passed into a component node to allow for a complete calculation of the refrigerant properties. All four main refrigerant component models use inlet refrigerant pressure and enthalpy to determine the state variables. The compressor inlet conditions are initialized from user-defined inputs and updated by the solver. The condenser and expansion device use the outlet conditions from the previous components. For the evaporator, the refrigerant inlet enthalpy are taken from the condenser outlet, as enthalpy is held constant across the expansion device. The evaporator refrigerant inlet pressure is initialized as the compressor inlet pressure. Note that the evaporator model outputs do not serve as inputs to the compressor model since, for calculation purposes, the compressor marks the start of the cycle and the evaporator the end.

The evaporator refrigerant outlet pressure is the only evaporator outlet used by another model; it enters the expansion device model as an input in design modes. In addition, there is a suction line calculation at the end of the evaporator model, but as the suction line exiting properties do not have an effect on the compressor inlet, there is no direct connection between the compressor inlet and the evaporator outlet.

3.3.1.1 Differences Between Condenser and Evaporator

One of the benefits of a heat pump is its ability to switch between heating and cooling mode based on the system load requirements. In cooling mode, the outdoor coil uses outdoor air to cool the refrigerant and the indoor coil uses the refrigerant to cool the indoor air. In heating mode, the components switch roles with the outdoor coil cooling the outdoor air and the indoor coil cooling the refrigerant by rejecting heat to the indoor air. In HPSim, the condenser routine models the indoor air circuit and the evaporator the outdoor air circuit during heating mode. The same basic structure is used for both circuits, but their defined properties can have differences, such as the number of tubes and rows, tube or fin type, geometry, coil material, or a difference in circuiting patterns. The evaporator component model in HPSim also calculates the pressure loss and heat transfer rate across the suction line.

For the standalone HPSim model, the cooling mode is defined once in the input file. For the coupled testbed, though, the cooling mode is set by EnergyPlus, based on the space cooling or heating requirements.

3.3.2 Convergence and Stability of the HPSim Refrigerant Side Models

For testbed usability and stability, it was important to investigate the convergence and stability of the HPSim refrigerant side models; this discussion is contained in the following sections. One overarching consideration for the HPSim refrigerant models is the units. HPSim models contain values in both IP and SI units; in general, SI units are used for calculations and IP units are used for inputs and outputs, but some cross-mixing occurs. The equations in the following sections are in the same units as they appear in the code. The condenser and evaporator models share the same coil model structure and are discussed in Section 3.4.

3.3.2.1 Compressor

A compressor compresses the refrigerant leaving the evaporator, increasing the temperature and pressure of the superheated gas. HPSim originally had a map-based compressor model, but the model did not have a compressor cycling ratio and thus HPSim could not be optimized through adjustment of that ratio. Therefore, a new adjustable capacity compressor model was implemented for testbed development. The compressor model uses a 10-coefficient curve dependent on the saturation temperatures of the compressor inlet and outlet to determine the compressor power. The compressor volumetric efficiency (ϵ_{vol}) is calculated by a 5-coefficient curve dependent on suction density (ρ_{suc}, kg/m^3) and the compressor ratio. The new compressor model was written by John Gall; his dissertation (Gall, 2015) does not discuss the exact compressor model, but Section 3.2 provides a good explanation of the power and volumetric efficiency curves.

The system mass flow rate, $\dot{m}_{ref}$, is calculated by

$$\dot{m}_{ref} = \rho_{suc} * V_{suc} * \epsilon_{vol} \qquad (3.3)$$

Where V_{suc} is the suction volume (m^3); V_{suc} is a function of the compressor size (for a 2 ton unit, $3.2984e^{-5}$ m^3 was used). $\dot{m}_{ref}$ is not directly affected by the compressor ratio, but the compressor ratio impacts it through ϵ_{vol}.

The inlet refrigerant properties are initialized by the enthalpy and pressure set in the input file; for following iterations, the inlet refrigerant properties are updated using the calculated inlet saturation temperature and a 100% quality. Figure 3.8 shows the most critical inputs, parameters, and outputs for the compressor model. Inputs refer to variables entering the model that can be changed by other models or the solver, parameters to variables defined once (usually by the user), and outputs to variables that are calculated by the model itself.

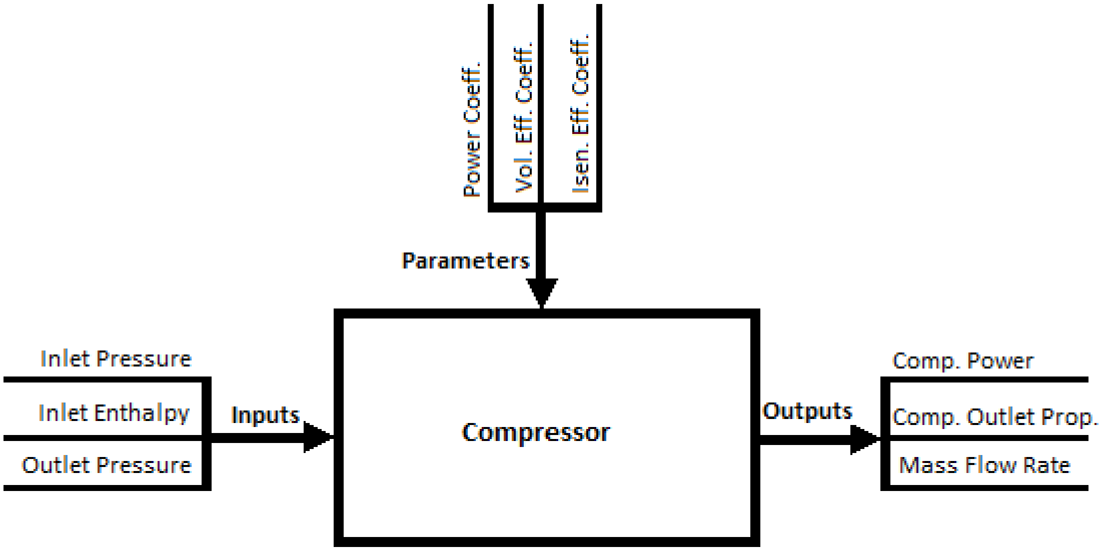


Figure 3.8 Compressor Model Inputs, Parameters, and Outputs

The most useful outputs for the simulation are $\dot{m}_{ref}$, outlet enthalpy, and outlet temperature, which are used as inputs to the condenser immediately downstream of the compressor. The power required by the compressor is reported for model sizing requirements. The discharge temperature is also passed to the evaporator routine to calculate the suction line heat transfer rate.

The routine uses a calculated discharge pressure and an isentropic efficiency across the compressor to calculate the refrigerant outlet properties. The discharge pressure is calculated based on the current iteration's discharge refrigerant temperature and a quality of 100%. The discharge enthalpy is determined by

$$h_{ref_{out}} = h_{ref_{in}} + \frac{\left(h_{ref_{is\,dis}} - h_{ref_{in}}\right)}{\eta} \quad (3.4)$$

Where $h_{ref_{out}}$ is the enthalpy of the refrigerant leaving the compressor (kJ/kg), $h_{ref_{in}}$ is the enthalpy of the refrigerant entering the compressor (kJ/kg), $h_{ref_{is\,dis}}$ is the discharge enthalpy at constant entropy (kJ/kg), and η is the isentropic efficiency of the compressor. The discharge temperature is updated through the refrigerant properties routines and the discharge pressure and enthalpy.

The compressor model's primary component is an unbounded polynomial curve; therefore, the compressor model is stable within the data range. Errors could occur in the model if improper map coefficients or other inputs were supplied or if an out-of-range compressor ratio was supplied by the optimization engine. The largest concern with the compressor model is the possibility of an unreasonable mass flow rate being calculated. Even so, those errors would not necessarily guarantee nonconvergence, but could result in physically impossible refrigeration cycles.

3.3.2.2 Fan Models

Originally, the fans were modeled by Equations 3.5 and 3.6 in the condenser and evaporator routines.

$$h_{air_{coil\ out}} = h_{air_{coil\ in}} + \frac{\dot{W}_{fan}}{\dot{m}_{air}} \qquad (3.5)$$

$$T_{air_{coil\ out}} = T_{air_{coil\ in}} + \frac{\dot{W}_{fan}}{c_{air}} \qquad (3.6)$$

Where $h_{air_{coil\ out}}$ is the coil outlet air enthalpy (kJ/kg), $h_{air_{coil\ in}}$ is the coil inlet air enthalpy (kJ/kg), $\dot{W}_{fan}$ is the fan power (W), $\dot{m}_{air}$ is the mass flow rate of the air (kg/s), $T_{air_{coil\ out}}$ is the coil outlet air temperature (°C), $T_{air_{coil\ in}}$ is the coil inlet air temperature (°C), and c_{air} is the specific heat of the air (kJ/(kg-s)). The equations found the change in enthalpy and temperature by manipulating Equations 3.5 and 3.6. The pressure drop was calculated by an air property routine in the coil calculation module as pressure drop over the coil; the pressure loss over the fan was ignored in the fan equations. The fan power was set in the input file.

A modified version of the EnergyPlus "SimSimpleFan" routine was implemented in HPSim so that fan power was dependent on the fan motor efficiency. This fan model disregarded night ventilation and fan schedules as the fan operates with the coil. The constant volume case was also removed from the fan routine since the fans operate as variable volume ones. The EnergyPlus model assumed the humidity ratio of the air was constant across the fan; this assumption remained. The HPSim assumption that the pressure loss of the air over the fan was insignificant in comparison to the pressure loss over the coil was removed, allowing the user to set a pressure drop over the fan. The outlet enthalpy was calculated by

$$h_{air_{out}} = h_{air_{in}} + \frac{\dot{W}_{fan\ to\ air}}{\dot{m}_{air}} \qquad (3.7)$$

Where $h_{air_{out}}$ is the enthalpy of the air leaving the fan (kJ/kg), $h_{air_{in}}$ is the enthalpy of the air entering the fan (kJ/kg), $\dot{W}_{fan\ to\ air}$ is the fan power transferred to the air (kW), and $\dot{m}_{air}$ is the mass flow rate of the air (kg/s). The calculation of the fan power transferred to the air is

$$\dot{W}_{fan\ to\ air} = \dot{W}_{shaft} + \left(\dot{W}_{air} - \dot{W}_{shaft}\right) * F_{motor_{air\ in}} \qquad (3.8)$$

Where $\dot{W}_{shaft}$ is the power supplied to the fan shaft (kW), $\dot{W}_{air}$ is the power required to move the air through the fan (kW), and $F_{motor_{air\ in}}$ is the fraction of the motor that is in the air stream (-). $F_{motor_{air\ in}}$ is set by the user in the input file. This power calculation was more detailed and sophisticated than in the original fan model.

3.3.2.3 Expansion Device

The expansion device allows the compressed liquid refrigerant leaving the condenser to expand back into a two-phase mixture before entering the evaporator. The model calculation methodology depends on the model used for the expansion device: thermal expansion valve (TXV), short tube, or capillary tube.

The TXV routine interpolates between minimum and maximum capacities using temperatures to determine the total capacity (tons). The minimum and maximum temperatures are dependent on refrigerant properties; the interpolation point is the saturation suction temperature of the refrigerant entering the compressor. The TXV routine does not model the refrigerant passing through it as the entering and exiting properties are already known.

For both the short and capillary tube models, the calculation methodology depends on the refrigerant enthalpy being constant throughout the expansion device and the exiting pressure being known; from these two state variables, all other exiting refrigerant properties can be determined. There are several different correlations (Payne (Payne and O'Neal, 2004 via Iu, 2007), Choi (Choi et al., 2004 via Iu, 2007), and curve-fit) available for calculating expansion device $\dot{m}_{ref}$ for the tube routines. All expansion device models calculate $\dot{m}_{ref}$ and exiting refrigerant properties.

In simulation modes, the model is part of the high side loop, calculating the system refrigerant properties. The short tube model calculates $\dot{m}_{ref}$ for a convergence check through use of the Payne correlation. The difference in refrigerant flow rates is reported back to the solver; if the difference is less than 0.0001 lbm/h, then the high side loop is considered converged. In design modes, the model sizes the expansion device component after system convergence; the calculated $\dot{m}_{ref}$ is used to check sizing in design modes. In design modes, the short tube diameter is calculated, the capillary tube diameter or the capillary tube length is calculated, and the TXV capacity is calculated.

All three types of expansion device are modeled in the design modes, with each model iterating until the calculated diameters have converged. Inputs are variables entering the model that are changed by other models or the solver and outputs internally calculated variables; the definition of parameters here depends on whether HPSim runs in design or simulation modes. In simulation modes, parameters are set once in the input file and never adjusted by HPSim. In design modes, parameters are initialized by the input file values but are adjusted by a small sizing loop until the expansion device $\dot{m}_{ref}$ is within tolerance of the compressor $\dot{m}_{ref}$. The parameters and most useful inputs and outputs for the expansion device models are shown in Figure 3.9.

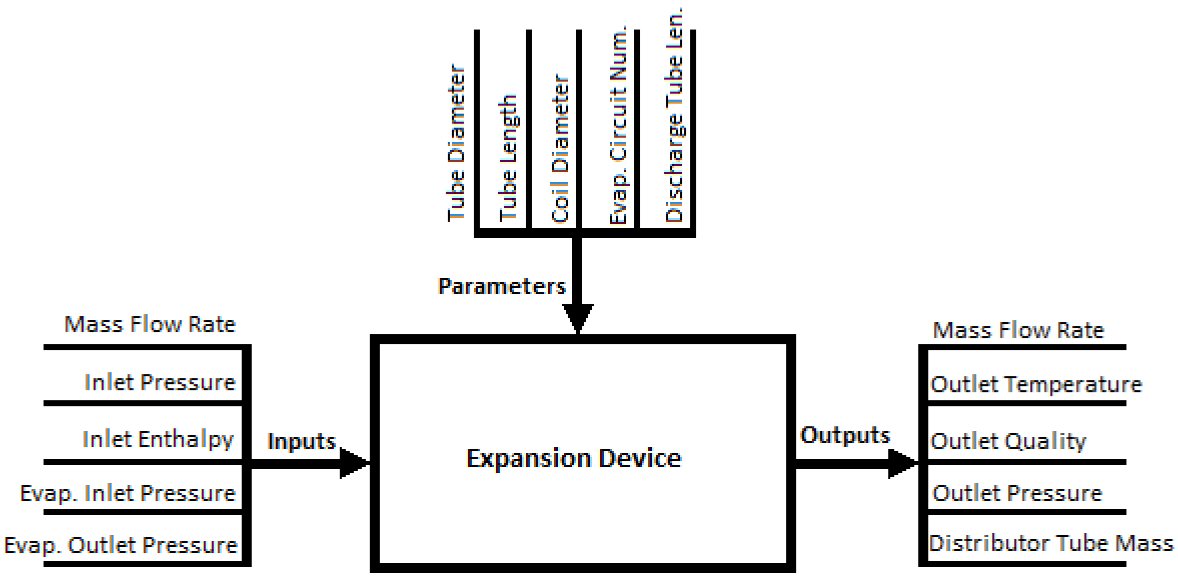


Figure 3.9 Expansion Device Model Inputs, Parameters, and Outputs

In simulation modes, the most critical output is $\dot{m}_{ref}$, which is used for a high side convergence check when the expansion device type is known. In design modes, the most critical output is $\dot{m}_{ref}$ also, as it is used for an expansion device sizing convergence check. In addition, the parameters of tube diameters, tube lengths, and other sizing information are important; these are reported to the user at the end of the simulation. The expansion device also has inlet connections to other refrigerant models in HPSim. These connections are the compressor outlet ($\dot{m}_{ref}$), the condenser outlet (refrigerant pressure and enthalpy), the number of circuits in the evaporator, the evaporator outlet (refrigerant pressure), and the pressure loss across the distributor.

The expansion device model is moderately stable, although incorrect inputs could create an error. Additionally, any errors upstream in the refrigerant cycle could result in flawed refrigerant properties entering the expansion device and creating more errors. Once again, these errors might create a nonconvergent or physically impossible cycle. In design modes, there are no critical convergent elements in this model.

3.3.2.4 Secondary Component Models

In addition to the two heat exchangers, the compressor, and the expansion device, there are several secondary components represented by models: the accumulator, filter-drier, and an optional distributor model. If a distributor tube length was assigned in the input file, the distributor model is called by the expansion device models for short and capillary tubes.

The accumulator stores any liquid refrigerant that may exit the evaporator; this ensures the refrigerant entering the compressor is completely in the vapor state, as the compressor would be damaged by any liquid entering (ACHR News, 2001). The filter-drier absorbs stray water in the system, as well as removing any foreign particulates from the refrigerant; water can react with refrigerant, forming acids that will corrode metal (ACHR News, 2000). The filter-drier is located after the condenser in the liquid line in HPSim.

In addition, the distributor divides and directs refrigerant into circuits in the evaporator coil model. It has an impact on system refrigerant charge and causes a loss in pressure, and thus is worth including.

3.3.2.5 Accumulator

The accumulator handles any extra liquid refrigerant and oil in the heat pump (ACHR News, 2001); it is located between the evaporator and the compressor, ensuring only superheated refrigerant is passed into the compressor. The accumulator model calculates the accumulator's pressure drop and the mass of the refrigerant inside it. The accumulator routine is called on the low side loop after the evaporator routine. Figure 3.10 shows the accumulator model's major inputs, parameters, and outputs. Inputs refer to variables entering the model that can be changed by other models or the solver, parameters to variables defined once (usually by the user), and outputs to variables that are calculated by the model itself.

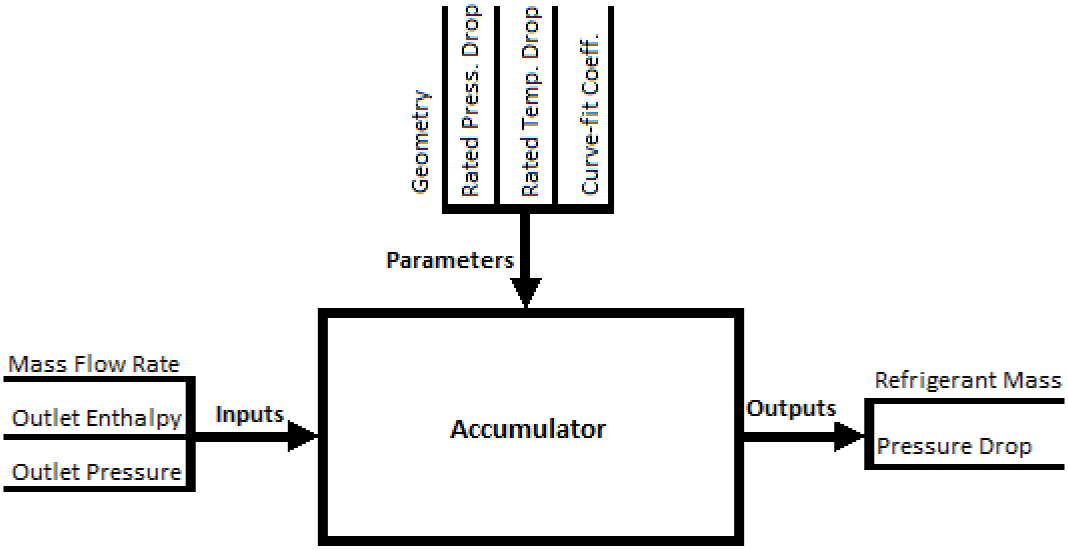


Figure 3.10 Accumulator Model Inputs, Parameters, and Outputs

The model checks for refrigerant phase and determines the refrigerant mass and liquid level in the accumulator before calculating the accumulator pressure drop,

$$\Delta p_{acc} = \frac{\dot{Q}_{sys}}{\dot{Q}_{sys_{rated}}} * \Delta p_{acc_{rated}} \qquad (3.9)$$

where Δp_{acc} is the pressure drop across the accumulator (kPa), $\dot{Q}_{sys}$ is the system capacity (ton), $\dot{Q}_{sys_{rated}}$ is the rated system capacity (ton), and $\Delta p_{acc_{rated}}$ is the rated pressure drop across the accumulator (kPa). The accumulator refrigerant mass is then calculated by

$$M_{acc} = A_{acc} * (H_{col} * \rho + \frac{H_{acc}}{V_{acc_{out}}}) \qquad (3.10)$$

Where M_{acc} is the accumulator refrigerant mass (lbm, converted to kg before the routine is finished), A_{acc} is the area of the accumulator (ft^2), H_{col} is the height of the liquid column in the accumulator (ft), ρ is the refrigerant density (lbm/ft^3), H_{acc} is the height of the accumulator (ft), and $V_{acc_{out}}$ is the volume of the refrigerant leaving the accumulator (ft^3). Overall, the only system impact is on the system refrigerant charge.

The accumulator model is stable, though if out-of-range pressure or temperature drops are inputted, the accumulator model could create errors. Similarly, errors in earlier refrigerant component models could manifest in the accumulator model. There are no critical convergent elements in this model.

3.3.2.6 Filter

In practice, the filter absorbs moisture and traps contaminants, and is located immediately after the condenser in the system. The input, parameters, and output for the filter model are shown in a diagram in Figure 3.11. The input is a variable entering the model that is calculated by another model, parameters are variables defined in the input file, and the output is a variable calculated by the model.

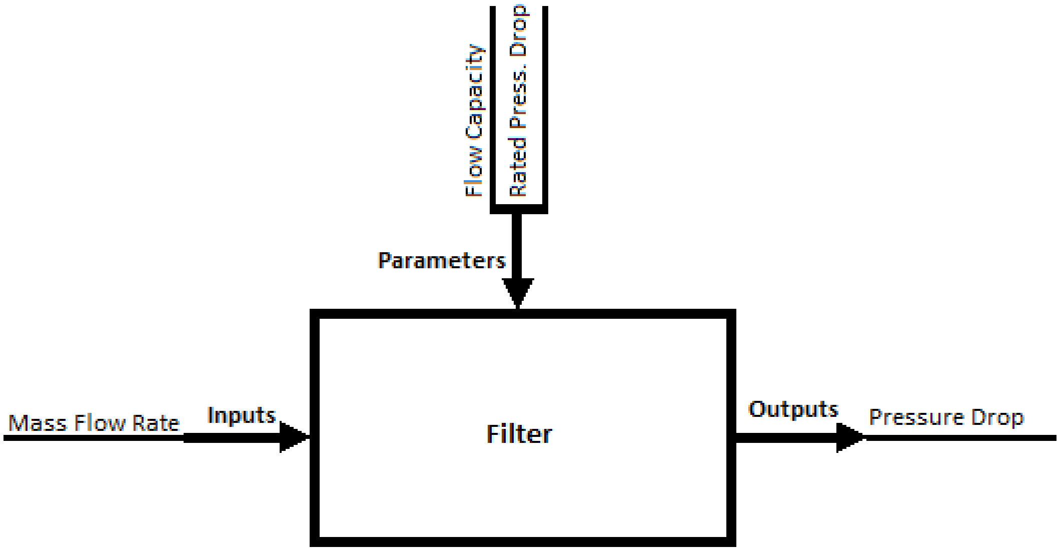


Figure 3.11 Filter Model Inputs, Parameters, and Outputs

The filter is modeled with a pressure drop calculation, through use of a catalog curve fit:

$$\Delta P = \frac{\Delta P_{rated}}{F_{rated} * F_{per\ ton}} * \dot{m}_{ref} \qquad (3.11)$$

Where ΔP is the pressure drop across the filter (kPa), ΔP_{rated} is the rated filter pressure drop (kPa), F_{rated} is the rated refrigerant flow capacity across the filter (tons), $F_{per\ ton}$ is the rated refrigerant flow rate per ton (kg/s/ton), and $\dot{m}_{ref}$ is the refrigerant mass flow rate across the filter (kg/s).

If an out-of-range pressure drop or flow capacity were inputted, the accumulator model could create errors. However, unless there is an error in $\dot{m}_{ref}$, this model is otherwise very stable. There are no critical convergent elements in this model.

3.3.2.7 Distributor

The distributor is located at the evaporator coil entrance and divides the refrigerant amongst the coil circuits. Although a distributor could be present at the condenser as well, HPSim only models it at the evaporator. The distributor model's inputs, parameter, and outputs are shown in a diagram in Figure 3.12. Inputs refer to variables entering the model changed by other models or the solver, the parameter to an user-defined value, and outputs to variables that internally calculated by the model.

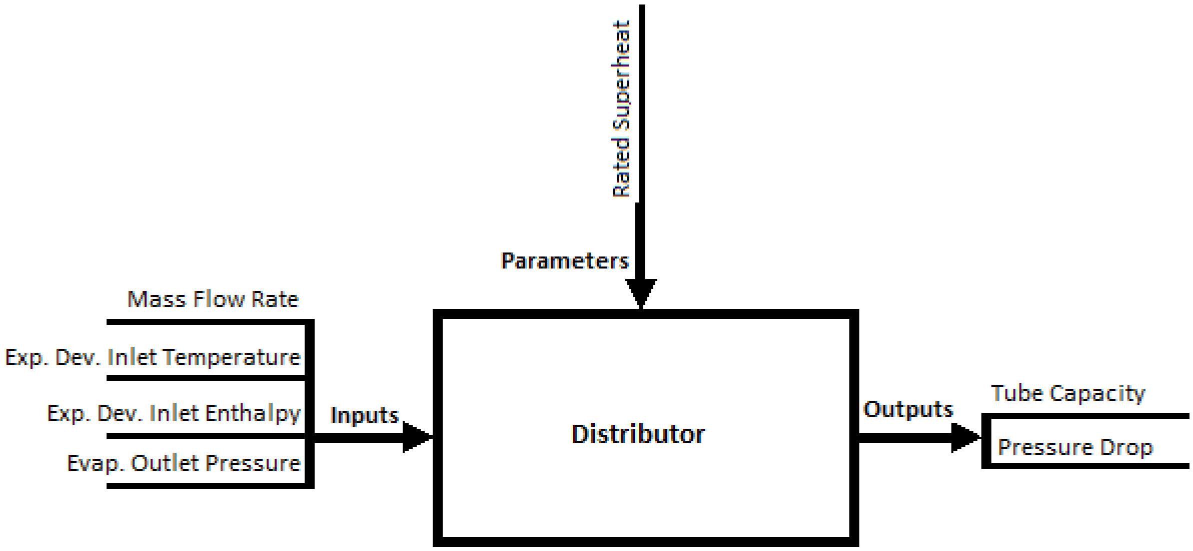


Figure 3.12 Distributor Model Inputs, Parameters, and Outputs

The distributor tube length is set in the input file, and the distributor model called by two of the expansion device models (capillary tube and short tube). The distributor capacity is calculated by a heat transfer equation,

$$\dot{Q} = \frac{\dot{m}_{ref} * (h_{evap_{out\,rated}} - h_{exp_{in}})}{Circuit\ Number} \qquad (3.12)$$

Where $\dot{Q}$ is the heat transfer rate across the distributor (kW), $\dot{m}_{ref}$ is the refrigerant mass flow rate (kg/s), $h_{evap_{out\,rated}}$ is the rated outlet enthalpy of the evaporator (kJ/kg), $h_{exp_{in}}$ is the inlet enthalpy of the expansion device (kJ/kg), and $Circuit\ Number$ is the number of circuits (-).

If an out-of-range superheat was inputted, the distributor model could develop an error. In addition, errors from earlier component models (particularly the expansion device model, which passes inputs to the distributor) could manifest in the distributor model. Otherwise, the distributor model is stable, and there are no critical convergent elements.

3.4 Integration of the HPSim Coil Models in the Simulation Testbed

The HPSim coil models serve as connection points to EnergyPlus in the simulation testbed. The condenser coil model connects to EnergyPlus through the outdoor air conditions. The evaporator coil model connects to EnergyPlus through the indoor air conditions and air flow rate. A connection diagram is given in Section 4.3 as Figure 4.2.

3.4.1 Overview of the HPSim Coil Models

The heat exchanger coil model structure is the same for both condenser and evaporator coils. In cooling mode, the condenser model primarily acts as the outdoor air circuit and the evaporator model as the indoor air circuit; during heating mode, the condenser routine models the indoor air circuit and the evaporator routine the outdoor air circuit. The coil models have two calculation modes: a simple one, usually used for initialization, and a detailed one.

The coils in the heat exchanger are divided into circuits, circuits are subdivided into tubes, and tubes are subdivided into segments, which are then modeled (Iu, 2007). The ε–NTU method is used to determine the heat transfer in the coils (Iu, 2007). The major assumption for the coil properties calculation is that, for multi-row coils, the input to any coil row is an average of the rows upstream; if the rows have yet to be calculated, the air outlet conditions of a segment are the inlet conditions to the adjacent segment in the next row (Iu, 2007). This assumption governs the iterations (Iu, 2007). Figure 3.13 shows a circuit with its tube and segment divisions.

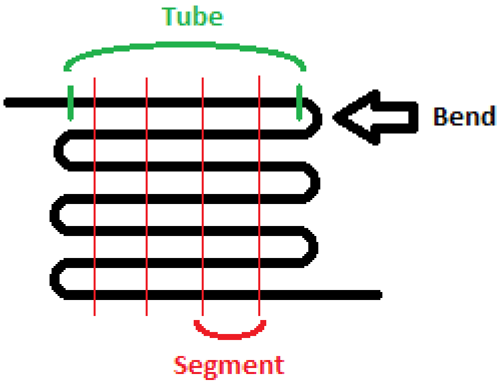


Figure 3.13 Heat Exchanger Circuit, Segments, and Tubes

The circuit is a section of the coil made up of tubes and bends; there can be multiple circuits in a coil when the flow is split before entering the coil and then joined again after the refrigerant leaves the coil. The tube is a straight section of piping between bends; one is denoted as being between the thick vertical lines. Segments are subdivisions of individual tubes, as shown by the thin vertical lines; here, each tube is divided into five segments (the bends are usually assumed to have negligible tube length).

The heat exchanger model divides each coil exchanger into circuits, which are made up of individual tubes. Each tube is divided into segments, and the heat transfer is found across each segment and then summed to result in the total heat transfer across the tube. The heat transfer is calculated across the heat exchanger as the air moves crosswise over the coil while the refrigerant flows through it. The heat exchanger heat transfer rate is then used to determine the properties of refrigerant leaving the segment, as calculated by

$$\dot{Q} = \varepsilon * C_{min} * \Delta T \qquad (3.13)$$

Where $\dot{Q}$ is the heat transfer rate across the heat exchanger (kW), ε is the heat exchanger effectiveness (-), C_{min} is the minimum heat capacity of the air and refrigerant heat capacities (kW/°C), and ΔT is the difference in inlet temperatures of the two fluids (°C). The correlation used for ε depends on which side has the minimum heat capacity rate. In a vapor compression cycle with a two-phase coil, the air-side will always have the minimum heat capacity rate since the refrigerant-side heat capacity is undefined in the two-phase region. The definition for specific heat is dependent on the partial derivative of either internal energy or enthalpy taken with respect to temperature; in the two-phase region, temperature is constant and thus the derivative is zero.

C_{min} is the minimum of the refrigerant specific heat, C_{ref}, and the air specific heat, C_{air}, which are each calculated by

$$C = \dot{m} * c_p \qquad (3.14)$$

Where C is the heat capacity (kW/°C), $\dot{m}$ is the mass flow rate of the fluid (kg/s), and c_p is the specific heat of the fluid (kJ/(kg-s)). The tube $\dot{m}_{ref}$ is determined by dividing the coil $\dot{m}_{ref}$ by the number of circuits in the coil; $\dot{m}_{ref}$ is constant through each tube and circuit. The mass flow rate of the air ($\dot{m}_{air}$) across the tube is dependent on the tube length and is determined by summing the mass flow rates of the air across the segments, as calculated by

$$\dot{m}_{air_{segment}} = \dot{m}_{air_{coil}} * \frac{L_{tube_{segment}}}{L_{tube} * N_{tube}} * V_{dev} \qquad (3.15)$$

Where $\dot{m}_{air_{segment}}$ is the mass flow rate of the air across the tube segment (kg/s), $\dot{m}_{air_{coil}}$ is the mass flow rate of the air across the entire coil (kg/s), $L_{tube_{segment}}$ is the length of the tube segment (m), L_{tube} is the full length of the tube (m), N_{tube} is the number of tubes in the coil (-), and V_{dev} is a weighting factor used to impose a non-uniform air velocity profile on the coil face. For uniform velocity, V_{dev} is set to 1.

The segment length can change depending on the coil model and the segment and iteration number. The detailed model divides each tube into a set number of equal-length segments. In the simple model, the first segment length is set equal to the circuit length and then decreased as needed in the following iterations until the transition boundary between superheated and two-phase states is found. Once the transition boundary is found, the first segment length is reset to be the length from the start of the tube to the point where transition begins to occur. If the transition boundary does not occur in the first segment, the length is not reset but kept as the full length of the tube; due to the segmenting logic, other lengths are set to zero but calculation routines are still called. In cases where the transition boundary is found in the first segment, the second segment length is initially determined by subtracting the first segment length from the circuit length; it is later reset as the length required for the refrigerant to go from two-phase to subcooled. The third segment length is found by subtracting the first segment length from the originally calculated second segment length. The fourth and following segment lengths are calculated by subtracting the first segment length from the previous segment's length. The number of segments per tube is specified in the input file. If any of the segments have negative lengths calculated, the program sets their lengths to zero and no heat transfer is calculated over those segments.

Simply put, the simple model treats a circuit as a single segmented tube while the detailed model treats a circuit as a collection of tubes by addressing circuiting (the order and pattern in which the tubes are connected). A combination of simple and detailed circuiting models were used in the coupled testbed, with the simple models used for preliminary calculation and the detailed models used for refining results.

The detailed circuiting model does require, by definition, more calculations and thus more modeling time. In addition, the circuiting sequences are inputs required only for the detailed model. Nonconvergence in some cases resulted from an error in the simple coil model circuiting routine; this is discussed further in Section 3.4.5.

3.4.2 Coil Model Structure

At the start of the calculation, refrigerant inlet properties are brought in, and the coil is broken down into circuits, tubes, and segments. The number of tube segments is set by the user in the input file along with the tube and circuit geometry. The refrigerant property outputs from one segment are the refrigerant property inputs for the next segment. Once the refrigerant properties for the circuits have internally converged, the overall coil heat transfer and outlet pressure are checked for convergence. When both have converged, the outlet refrigerant properties are passed back to the component model. Figure 3.14 shows the flow chart for the heat exchanger model.

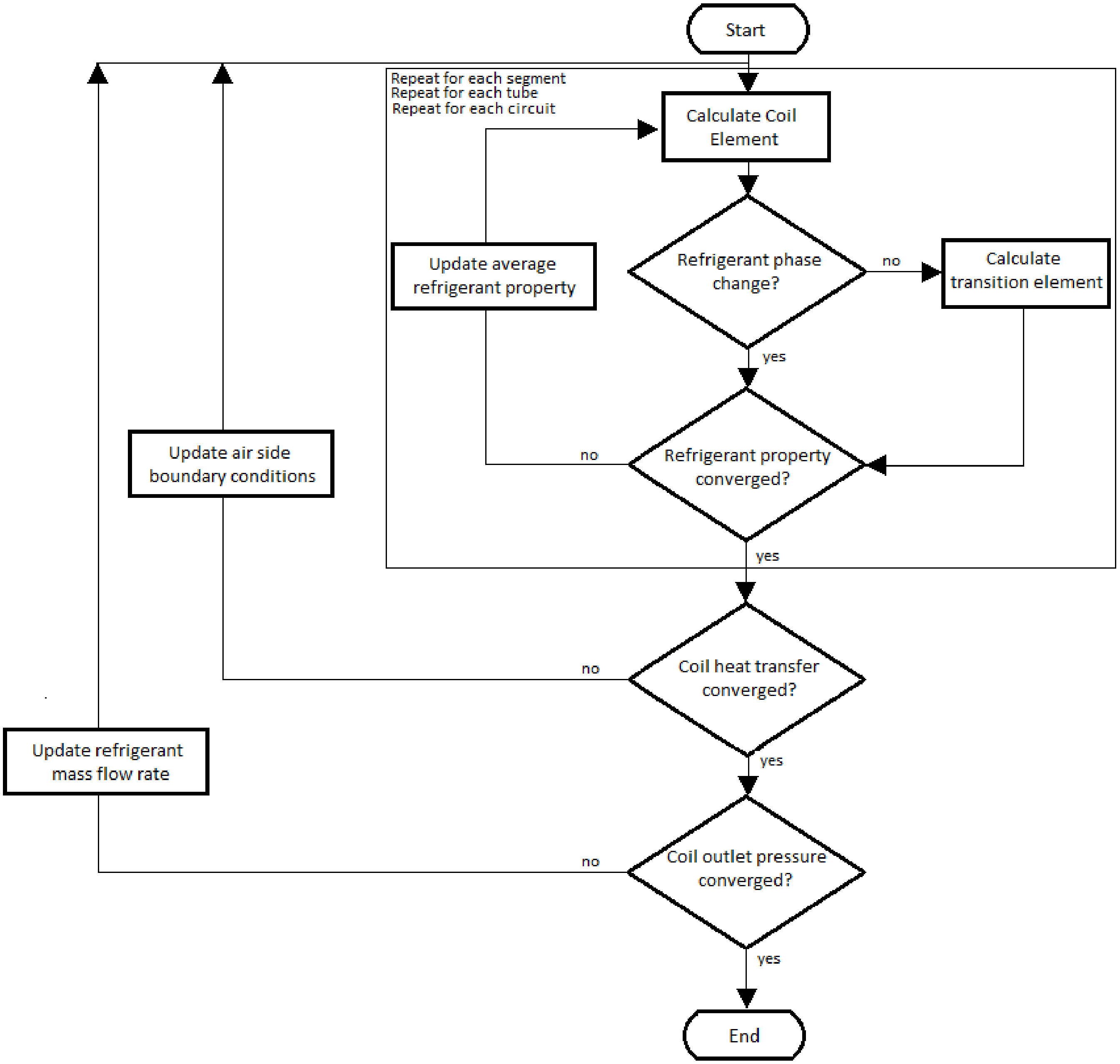

Start
Repeat for each segment
Repeat for each tube
Repeat for each circuit
Calculate Coil Element
Update average refrigerant property
Refrigerant phase change?
no
Calculate transition element
yes
Update air side boundary conditions
no
Refrigerant property converged?
yes
no
Coil heat transfer converged?
yes
Update refrigerant mass flow rate
no
Coil outlet pressure converged?
yes
End

Figure 3.14 Heat Exchanger Model Program Flow (Iu, 2007)

The refrigerant system mass flow rate is divided into each heat exchanger circuit equally at the start of the calculation. If the convergence check for the coil outlet pressure fails, the refrigerant mass flow rate is redistributed based on circuiting sequences and calculated pressure drop. The pressure drop of the refrigerant is calculated for each tube segment, taking into account gravity, friction, and acceleration (Iu, 2007). The segment pressure losses are summed to determine the pressure drop across the coil. The coil pressure drop is small in comparison to the expansion device's pressure drop (Iu, 2007); however, the pressure loss marks a real-life departure from the ideal of a constant pressure heat exchanger.

The coil models are relatively stable, but can be affected by improper user inputs or by errors from upstream refrigerant models. The air-side properties and equations add another opportunity for incorrect or out-of-range information to create errors. In addition, Section 3.4.5 discusses an instance where a flaw in the circuiting modeling routine allowed for a sudden end of calculations due to nonconvergence.

3.4.3 Condenser

Figure 3.15 shows the major inputs, parameters, and outputs for the condenser model. Inputs refer to variables entering the model that can be changed by other models or the solver, parameters to variables defined once (usually by the user), and outputs to variables that are calculated by the model itself.

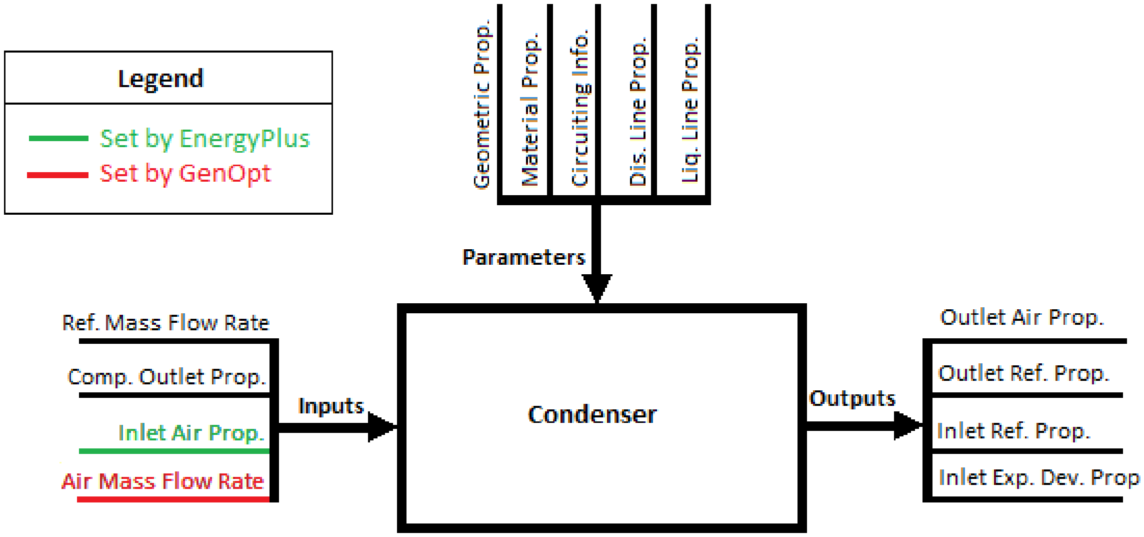


Figure 3.15 Condenser Model Inputs, Parameters, and Outputs

In the coupled mode, the inlet air properties are set by EnergyPlus and the air mass flow rate can be set by GenOpt or left as an user-defined value. The condenser is primarily a heat exchanger model; the coil and circuiting setup was covered in Section 3.4.1. For calculation of heat transfer, the coil is divided into circuits, circuits are divided into tubes, and tubes divided into segments. The heat transfer across each segment is calculated, then summed together to determine the tube, circuit, and coil heat transfer rates. Each segment calculation contains a loop to ensure convergence of the outlet conditions. The coil code also contains a loop to check convergence of the heat transfer rate across the entire heat exchanger.

The condenser model includes a routine accounting for the length of the liquid line and its effect on the refrigerant in the system. The temperature drop across the liquid line is provided in the input file, and then the heat transfer rate is calculated using

$$\dot{Q} = \dot{m}_{ref} * c_p * \Delta T \qquad (3.17)$$

Where $\dot{Q}$ is the heat transfer rate over the line (kW), $\dot{m}_{ref}$ is the mass flow rate of the liquid refrigerant (kg/s), c_p is the specific heat of the refrigerant (kJ/(kg-s)), and ΔT is the change in temperature through the line (°C). Once the heat transfer rate is known, the outlet refrigerant enthalpy is determined by

$$h_{ref_{out}} = \frac{-\dot{Q}}{\dot{m}_{ref}} + h_{ref_{in}} \qquad (3.18)$$

Where $h_{ref_{out}}$ is the outlet refrigerant enthalpy (kJ/kg), $\dot{Q}$ is the heat transfer rate across the considered tube length (kW), $\dot{m}_{ref}$ is the refrigerant mass flow rate (kg/s), and $h_{ref_{in}}$ is the inlet refrigerant enthalpy (kJ/kg).

The outdoor entering air conditions are required inputs, provided by the input file in the standalone HPSim but provided by EnergyPlus in the testbed; this allows exterior weather to change with time. Refrigerant properties are the primary simulated inputs to the condenser model, and the input file provides the coil and circuiting geometry and connections parameters. The exiting refrigerant properties are the most critical outputs from the condenser model. None of the condenser outputs are returned to EnergyPlus in the testbed.

3.4.4 Evaporator

For the purposes of this explanation, the heat pump is run in cooling mode, so the evaporator routine models the indoor circuit. The evaporator model follows the same structure as the condenser model, as discussed in Sections 3.4.1. and 3.4.2. The difference between the two models are some minor differences in inputs, outputs, and calculations due to one being the indoor air circuit and the other being the outdoor air circuit. The defined operating mode sets the loading of the correct tube circuiting for the evaporator model, dependent on whether it acts as an indoor or outdoor circuit. The evaporator routine models the outdoor circuit when the heat pump is in heating mode, and the indoor circuit in cooling mode. A diagram showing the most critical inputs, parameters, and outputs for the evaporator model is seen in Figure 3.16. Variables entering the model that can be changed by other models or the solver are inputs, variables defined once in the input file are parameters, and variables calculated by the model are outputs.

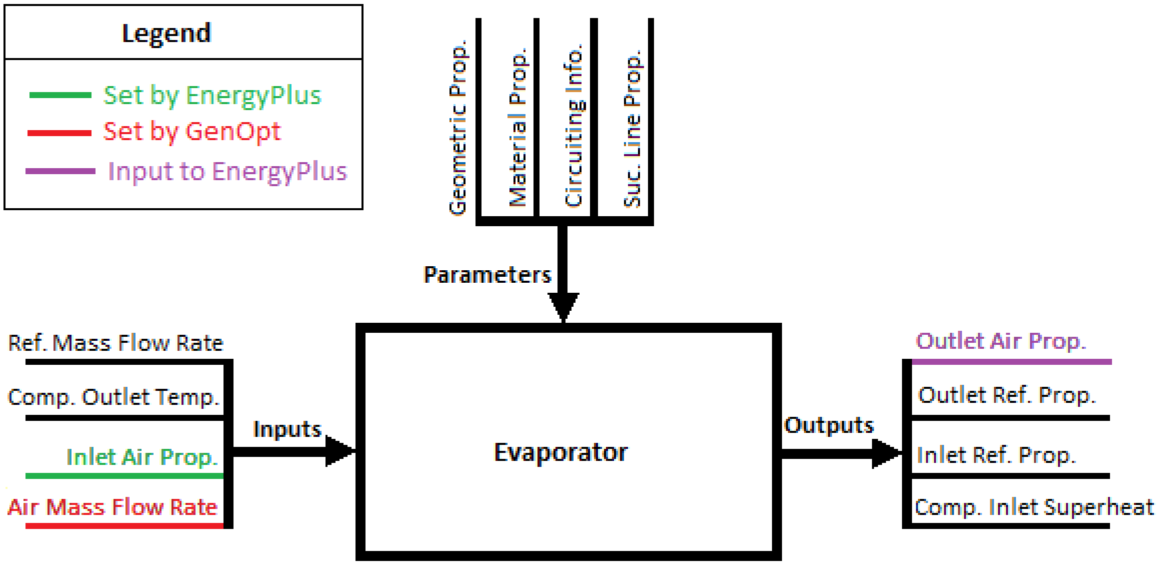


Figure 3.16 Evaporator Model Inputs, Parameters, and Outputs

In the testbed, the inlet air properties are set by EnergyPlus, the air mass flow rate is set by GenOpt, and the outlet air properties are reported back to EnergyPlus. The evaporator inlet refrigerant pressure is adjusted in the first iteration of each low side loop. The adjustment is calculated by

$$p_{evap\ in} = p_{comp\ in} + (p_{evap\ in} - p_{evap\ out}) \quad (3.19)$$

Where $p_{evap\ in}$ is the evaporator inlet refrigerant pressure (kPa), $p_{comp\ in}$ is the compressor inlet refrigerant pressure (kPa), and $p_{evap\ out}$ is the evaporator outlet refrigerant pressure (kPa). The $p_{evap\ in}$ on the right hand side of the equation is the compressor inlet pressure at the first calculation and the previous iteration's calculated $p_{evap\ in}$ for every iteration after. $p_{evap\ out}$ is initialized in the evaporator routine call prior to the first low side loop calculation; this call is to determine coil case (detailed or simple coil mode). In all following calculations, $p_{evap\ out}$ is saved from previous iterations.

The evaporator model accounts for the length of the suction line and its effect on the refrigerant in the line. The change in enthalpy over the suction line is calculated using

$$h_{ref_{out}} = UA * (T_{ref_{comp\ out}} - T_{ref_{comp\ in}}) + h_{ref_{in}} \qquad (3.20)$$

where $h_{ref_{out}}$ is the refrigerant enthalpy leaving the suction line (kJ/kg), UA is the heat transfer coefficient (kW/K), $T_{ref_{comp\ out}}$ is the compressor discharge temperature (°C), $T_{ref_{comp\ in}}$ is the compressor suction temperature (°C), and $h_{ref_{in}}$is the refrigerant enthalpy entering the suction line (kJ/kg). $h_{ref_{out}}$ is then reported as the evaporator refrigerant outlet enthalpy and used, along with a set pressure drop, to calculate the remaining refrigerant outlet properties.

The main simulation variables passed from one component model to another are refrigerant properties. The coil and circuiting geometry and connections are read in from the input file. One other significant input set is the indoor entering air conditions. In the coupled version of HPSim, these are defined by EnergyPlus instead of by the user, allowing dynamic air conditions based on the exterior weather conditions and internal space conditions. This allows the heating or cooling impact of HPSim on the room conditions in one timestep to be reflected in the air returned to HPSim at the next timestep. The important outputs from the evaporator component model are the exiting refrigerant and air properties; the exiting air properties are reported back to EnergyPlus in the testbed configuration.

3.4.5 Saturated-Superheated Transition

In the circuiting model, refrigerant phase transition points are calculated using heat transfer equations, and the resulting qualities are used in the program to determine the refrigerant phase. The superheated-to-two-phase transition segment length is calculated by adjusting the segment length until the refrigerant begins to go into the two-phase region. The two-phase-to-subcooled transition length is similarly calculated by adjusting the segment length until the refrigerant begins to go into the subcooled region. Both transition points are checked by slightly perturbing the entering enthalpy at the start of the heat transfer equations.

An error can occur when the initial length of the second segment is too long for the two-phase refrigerant calculation. If the length is too great, the heat capacity of the refrigerant will not be calculated for the subcooled region, resulting in an incorrect minimum heat capacity and an error in the calculation of the heat transfer rate. The segment outlet refrigerant enthalpy is calculated by Equation 3.18. When $\frac{-\dot{Q}}{\dot{m}_{ref}}$ is larger than $h_{ref_{in}}$, $h_{ref_{out}}$ will be calculated as negative. Since $h_{ref_{out}}$ is used to determine the outlet refrigerant properties, a negative value creates errors in the property routines and can cause the program run to abort.

The transition segment length routine calculates the length of the segment required to bring the refrigerant from two-phase to subcooled liquid; the calculated length is used to determine the rate of heat transfer across the heat exchanger, which in turn is used to find the properties of the refrigerant leaving the segment. The transition segment length calculation is called in the second iteration of the coil exchanger program, but if $h_{ref_{out}}$ is calculated as negative in the first iteration, the program may abort the run before it reaches its second iteration. As shown in Equations 3.13, 3.14, and 3.15, the heat exchanger heat transfer rate can be dependent on $\dot{m}_{air}$ across the segment length: the longer the segment, the greater the fraction of the coil air flows across it and thus the larger the $\dot{m}_{air}$.

The implemented solution to the above error was to abort the first iteration early and cycle to start the second iteration so that the transition length could be properly calculated. An “IF” statement checked the segment $h_{ref_{out}}$; if $h_{ref_{out}}$was less than zero the program returned to the start early and began the next iteration. Enthalpy was chosen as the checked variable for a couple of reasons: one, it is directly calculated using the heat exchanger $\dot{Q}$ and two, it is used (along with exiting refrigerant pressure) to calculate the leaving refrigerant properties. When $h_{ref_{out}}$ is negative, temperature and other properties are calculated as negative values as well. In addition to this being incorrect, the program cannot handle these negative values, resulting in non-convergence and an early stop to the run most of the time the situation occurs.

A check was also added to the “IF” statement that allowed a shortened iteration round only on the first iteration. This helped keep the program from continually skipping out of failed iterations and then continuing as if the iteration had been successfully carried out. The “IF” statement was located almost directly after Equation 3.18 was calculated; there was an entering refrigerant property calculation routine called immediately after Equation 3.18, and the “IF” statement was located after this call.

The iteration-restart option was chosen because of the opportunity for the transition segment length to be calculated in the second iteration. This option does not “smooth over” an incorrect calculation or change any of the system’s physical properties. It merely allows the solver to properly calculate the transition length and determine the refrigerant properties across it. If the transition segment cannot be properly calculated and the refrigerant cycle is out of bounds after the second iteration, the program is allowed to continue iterating or to abort the run if it cannot handle any calculated negative properties.

The iteration-restart option was tested by the following methods. Initially, a case known to fail due to negatively calculated properties was run to determine: a) if the “IF” statement correctly aborted the first iteration and started a second iteration, b) if the second iteration’s calculation of the transition segment could be correctly carried out, and c) if the transition segment calculation determined correct and positive refrigerant properties. Once the first case was satisfactorily completed, a couple of known successful cases were run to ensure that the “IF” statement did not have any unforeseen effects upon correctly calculated cases. Lastly, a few more known failed cases were run to check that the method worked for more than one case.

Test case a) was shown to be correct by debugging and stepping through a pair of iterations. Test case b) was shown to be correct as well; the transition boundary point was calculated with use of a “DO” loop that did not stop until converged. The “DO” loop converged on the segment length and the $\dot{Q}$ across the segment length. Test case c) was correct for the given case provided case b) was also correct: if the transition boundary was found, the properties returned were real and positive.

The first fully run case completed after multiple iterations and multiple uses of the “IF” statement. However, the final converged result for the condenser did not reach subcooled conditions at the outlet. Instead, two-phase refrigerant left the condenser, traveled through the expansion device, and entered the evaporator. There were multiple cases where this situation occurred; likely causes are discussed in Section 6.3.

In addition, several cases that had previously completed successfully were run for comparison. The results for these cases were recorded before the changes were made and then again after the addition of the “IF” statement. All but two of these sets of heat transfer rates matched. The first unmatched case had a difference of 0.57% from the original case, which was insignificant, and the second had a difference of 2.34%, which was only slightly less insignificant. These cases were briefly stepped through in the code, and they both triggered the “IF” statement several times on first segment iterations. However, since they completed successfully before the addition of the “IF” statement, it appears that the difference in results was due to the slight change in calculation algorithm made by the “IF” statement.

Several additional cases known to abort mid-run before the “IF” statement was added were also rerun with the correction. About half completed successfully, while the other half exited out of calculations early because the pressure drop across the coil was too large. In the latter case, the exit was controlled through error handling and not as the result of a sudden program error.

3.4.6 Air-Side Models and Coupling

In the original HPSim model, the entering condenser and evaporator air wetbulb and drybulb temperatures were set by the user in the input file. The associated air properties were then calculated using a property lookup routine. Because of the need for new entering air temperatures each time HPSim was called by the testbed, this setup had to change.

The testbed coupling required creating a new input file – written and controlled by EnergyPlus – from which HPSim could read information. This input file was independent of the main program input file, containing only the values changing each timestep – the entering air conditions for both the evaporator and the condenser. The input-reading routines of HPSim were necessarily modified to account for the additional input file. Air properties at points along the coils are still determined by the property lookup routine as needed.

3.5 Updates to HPSim for Coupling

A number of changes were made to HPSIM to prepare it for coupling; the majority of changes were general modifications for cleanness, flexibility, and robustness, but there were also specific ones for testbed coupling. Error handling was also a critical consideration.

3.5.1 General Modifications to HPSim

General modifications are listed below.

1. The underlying code structure was significantly modified through replacement of the existing array structure with a structure oriented around the component models. Originally, many arrays holding input and output refrigerant and air data, as well as general system parameters, were

passed between routines. The new configuration placed the inputs, parameters, and outputs in data structures that were globally available for reading and updating. This improved simulation run times.

2. All “GOTO” statements were removed and replaced with flags and IF statements for more linear program flow. This gave a cleaner code for modification and improvement.
3. Variable names were changed for clarity; this facilitated debugging and future development.
4. Existing documentation was updated and new documentation added for a better understanding of the code; this also facilitated debugging and future development.
5. An error that calculated the pressure and heat loss over the suction line twice was fixed; the suction line is only accounted for once in the cycle now.
6. Overall cleanup was undertaken to remove deprecated routines and lines, and for general streamlining and improvement in efficiency.

Several larger modifications were made, as discussed in Sections 3.1.5, 3.3.2.2, and 3.4.5. Changes to the input/output structure were also made, as seen in Section 3.5.2.

3.5.2 Coil Model Air-Side Inputs and Outputs

In the testbed, the coil model air-side inputs are set by EnergyPlus every timestep instead of once by the user. The correct outdoor air conditions are found by EnergyPlus from a weather file, and then passed through to HPSim. The indoor air conditions are calculated by EnergyPlus – taking into account internal building loads, external envelope loads, and the conditioning provided by HPSim in the previous timestep – and then passed to HPSim. EnergyPlus writes a file with the air conditions, which HPSim reads in addition to its usual input file. This required HPSim’s input structure to be modified to accommodate the new provision of entering air conditions.

Finally, HPSim's evaporator leaving air conditions are written to a secondary output file for use by EnergyPlus. When optimization has finished, EnergyPlus reads in the final optimized output file from HPSim, which allows for accounting of HPSim's heating or cooling.

3.6 Conclusions

HPSim as originally written was not suitable for coupling in a thermal controls testbed environment. Therefore, modifications were required for testbed development. Major modifications were fan model updates, changing the coil model air condition input location, and increasing the coil model robustness. Additional modifications included general cleanup and convergence checks. With these changes made, HPSim was ready for coupling with EnergyPlus to create the testbed.

CHAPTER IV

TESTBED COUPLING METHODOLOGY AND PROGRAMS

In order to create a controls testbed, a detailed building model was necessary to simulate the building and outdoor environmental conditions; EnergyPlus was chosen to fill this need. A brief explanation of EnergyPlus is contained in this chapter, along with an overview of the testbed coupling methodology and additional programs employed.

4.1 EnergyPlus Model

EnergyPlus is a fully integrated component-based modeling tool for equipment sizing, equipment selection, and energy performance analysis (Trčka and Hensen, 2010). It is a modular whole building simulation program with individual system component models; these models are useful for their flexibility and inter-changeability. The modularity was a benefit for the testbed development, as new code was added to EnergyPlus for the testbed.

EnergyPlus operates over two timesteps: zone timesteps and system timesteps. Building heat transfer and thermal loads are calculated on the zone timestep, while the system timestep simulates the HVAC system and related calculations. The zone timestep is a fixed value set by the user – from a range of one hour to ten minutes – while the system timestep is dependent on calculation convergence and ranges between the length of the zone timestep and one minute. EnergyPlus simulates room physics to determine space load, simulates the HVAC system's removal of the load, and then determines the instantaneous power required by the system components at that operating point. EnergyPlus is a quasi-steady state simulation, while HPSim is a steady state simulation. One of the challenges of the testbed was to run HPSim as quasi-steady state. The HVAC system is assumed to be at a constant state over each system timestep; the integration of the instantaneous power over the system timestep allows for the computation of system energy. HPSim and the testbed operate in the system timestep.

EnergyPlus also operates using a detailed nodal structure, which stores the state variable information; every component contains an inlet and outlet node for connecting into fluid (air) loops. Component model inputs convey the fluid mass flow rate and the inlet state of the fluid. The model determines the exiting fluid mass flow rate and the outlet fluid state, setting the node outputs. Because the inputs and outputs share the same structure and information, users are allowed to select the ordering of objects in fluid loops, as well as easily calculate parallel loop segments. This nodal structure had to be kept intact for successful operation of EnergyPlus and the testbed; breaking into the nodal structure was one of the challenges faced in this project.

4.2 EnergyPlus Zone and Heat Pump Models

The simplest and cleanest way to integrate into the EnergyPlus nodal structure was to investigate EnergyPlus's heat pump models. The standard EnergyPlus heat pump model places its components of mixing box, fan, and coil in separate routines. Figure 4.1 shows a simple diagram of how the heat pump connects to the zone (also an air-side component). For the testbed heat pump case, the basic routine structure remained the same but only the mixing box was modeled by EnergyPlus; the fan and coils were modeled by HPSim. The primary air properties passed through the air-side components are temperature and humidity ratio. The zone model computes the zone air heat and moisture balances, then solves for the mean air drybulb temperature and humidity ratio. The outside air properties are updated by the weather routine and data file.

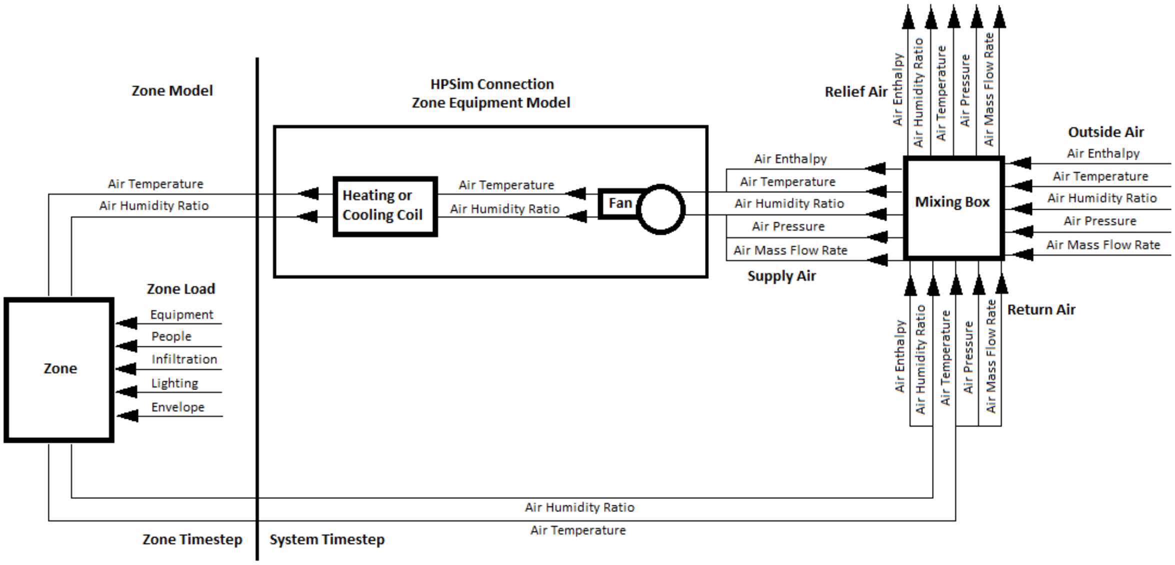


Figure 4.1 Heat Pump and Zone Connections

The EnergyPlus coil model is much simpler than the HPSim coil model described in Section 3.4.1. It starts its run by setting up nodes and nodal properties; if EnergyPlus has reduced the air flow rate from full-load to part-load, the coil capacity is similarly reduced. The Coil Bypass Factor (CBF) is also adjusted based on actual air flow vs. rated air flow. Then EnergyPlus checks the boundaries for any low ambient temperature conditions it needs to warn on. The total capacity is determined off the curve based on full mass flow and the air mass flow ratio; the capacity curve is biquadratic and defined in the input file. The part-load factor (PLF) is also calculated and checked to ensure it is neither too high nor too low. The coil leaving conditions are calculated for full load capacity and used to calculate the part-load capacity coil leaving conditions; these are calculated using the PLF and air flow ratio, as well as the entering coil conditions. If any air bypassed the coil, the coil leaving air conditions and the bypass air conditions are weighted together to determine the mixed air conditions leaving the evaporator. The energy use of the evaporator fan is calculated based on airflow and user inputs, as described in Section 3.3.2.2.

4.3 Coupling Methodology

In order to determine where the call to HPSim could be placed, an EnergyPlus heat pump model's flow path was traced to locate the air node connections. The air node connections were located in the coil routine, so it was decided to locate the call to HPSim in a coil model routine. Another equipment case for HPSim was created inside EnergyPlus Zone Equipment, allowing for a simple connection to the air nodes. The fan component was bypassed in EnergyPlus and modeled instead in HPSim. The coupled system diagram is shown in Figure 4.2, with refrigerant nodes in red and air nodes in blue.

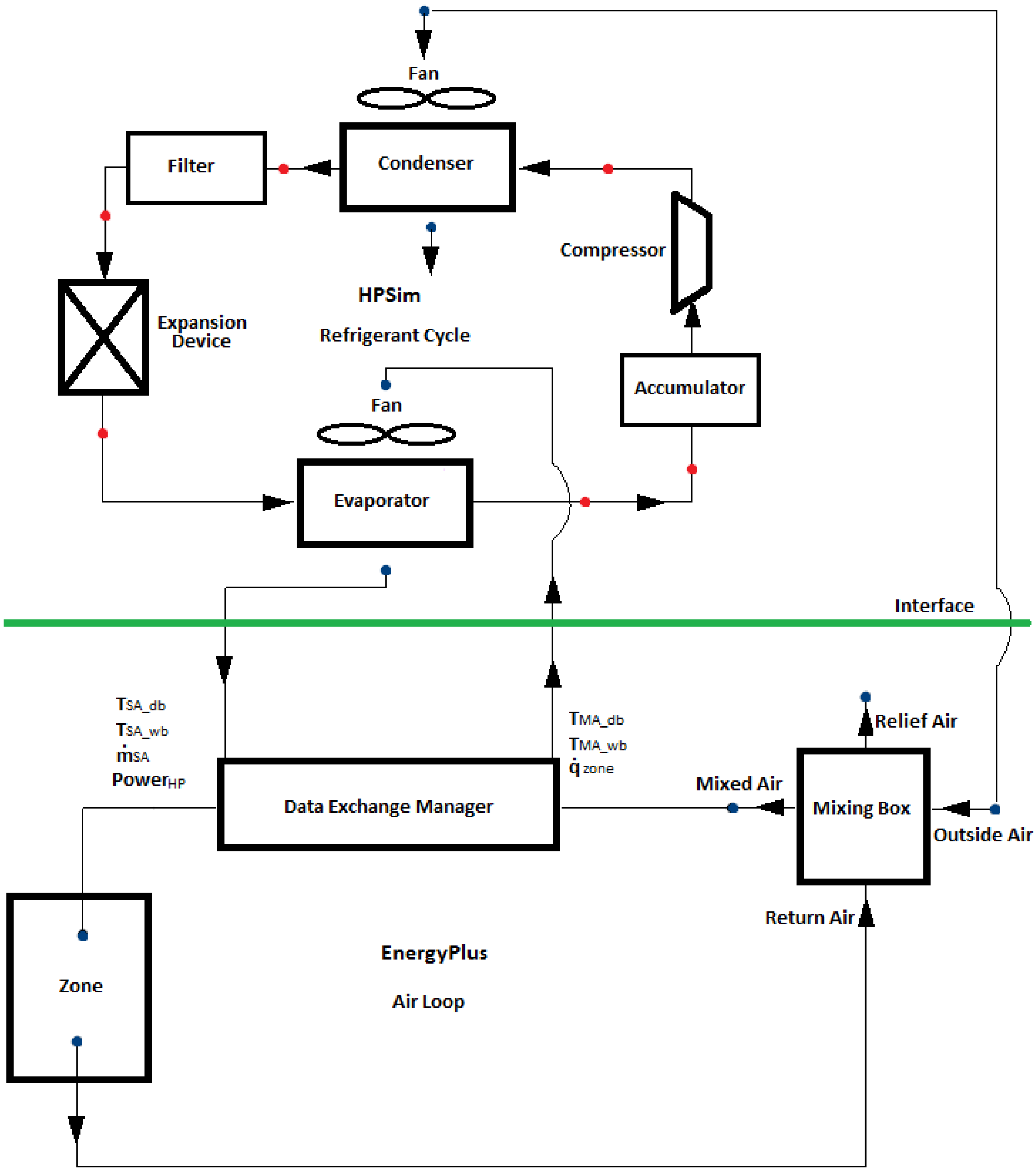


Figure 4.2 HPSim and EnergyPlus Coupled System Diagram

The refrigerant cycle is modeled by HPSim, consisting of an accumulator, compressor, condenser, filter, expansion device, and evaporator. In order to model the refrigerant cycle, HPSim also models the air passing over the evaporator and condenser coils. The connection between the refrigerant and air cycles is located at the evaporator coil, which transfers heat between the refrigerant and the air. The air loop consists of a zone, mixing box, fan, and heating or cooling coil; the zone and mixing box are modeled by EnergyPlus in the testbed. Therefore, the air-side of the coil is the connection point between EnergyPlus and HPSim: HPSim models the air through the fan and over the evaporator coil and then returns the outputs to EnergyPlus.

The EnergyPlus model "CalcDoe2DXCoil" was the basis for the routine containing the connection point to HPSim. This allowed HPSim to integrate into the existing nodal structure with few modifications. From EnergyPlus' view, the nodal structure remained intact as it passed in and received out the same variables it would with any other heat pump routine. The code handling the nodal structure and other entering and exiting variables, including the outdoor air conditions, remained in place. Care was required to determine which variables were necessary to the nodal structure and which could be removed. The coil calculations native to the EnergyPlus routine were stripped out and replaced with the call to the testbed; the power calculations were removed as well.

The EnergyPlus call to HPSim is executed only if three conditions are met: the mass flow rate of the air leaving the mixing box is greater than zero, the coil is scheduled on, and the set point of the zone thermostat is less than the temperature of the return air entering the mixing box. These conditions are simple but keep HPSim from being called when the system is off or the zone load satisfied. Although GenOpt sets the system flow rate of the air through the evaporator, EnergyPlus must initialize the need for air flow. For the proof-of-concept work of the testbed, HPSim is hardcoded to run in cooling mode only.

The variables passed into HPSim are the outdoor air entering properties, the indoor coil air entering drybulb and wetbulb temperatures ($T_{air_{evap\ in}}$ and $T_{air_{evap\ in\ wb}}$, both °F), and the zone cooling load ($\dot{Q}_{zone}$, Btu/h). The air property inputs allow HPSim to update its starting air properties every call from EnergyPlus based upon the heat gains in the zone. The zone cooling load is used as part of the penalty function, as discussed in Section 5.2. The variables passed back to EnergyPlus after HPSim completes its run are total coil heat transfer rate ($\dot{Q}_{tot}$, Btu/h), total sensible coil heat transfer rate ($\dot{Q}_{sens}$, Btu/h), indoor and outdoor coil exiting air temperatures ($T_{air_{evap\ out}}$ and $T_{air_{cond\ out}}$, both °F), mass flow rate of the air leaving the indoor coil ($\dot{m}_{air}$, kg/s), compressor ratio (-), and the total system power of the compressor and fans ($Power$, W). These outputs update EnergyPlus with the cooling provided by HPSim and the leaving coil air conditions for the current timestep.

The compressor power, evaporator fan power, and the condenser fan power were summed together and reported as the total system power. Originally, the HPSim condenser and evaporator fan models did not calculate fan power but required power as an user input value. In order for power to be impacted by the flow rate of the air, a simple fan model from EnergyPlus was modified and implemented in HPSim. This modification was discussed in Section 3.3.2.2.

HPSim's coil models take the circuiting details into consideration, including tube and segment lengths, calculating refrigerant properties as they travel through the coil, circuit by circuit and segment by segment. HPSim uses refrigerant tables to determine properties as the refrigerant travels through the coil. HPSim's coil model calculates where the transition boundary between superheated and two-phase states is located. HPSim's coil model was explained in detail in Section 3.4.1.

4.4 Simulation Methodology

In order to develop the testbed, instabilities had to be addressed with the coupled programs. There were two major potential instabilities with optimizing and coupling HPSim:

1. A poor initial guess; the initialization function will attempt to bracket the solution and accommodate for bad initial guesses but it can still fail.

2. Scalability; if the optimization engine adjusted the compressor ratio but not the air flow rate, HPSim could return a non-convergent psychrometric cycle.

The HPSim modifications for coupling were discussed in Chapter III.

There are two primary types of simulation integration as listed by Hensen (1999): a loosely coupled or “ping pong” method, where the outputs of one model are used as the inputs of the next model at the next timestep, and a tightly coupled or “onion” one, where the models iterate within the same timestep until the values converge. The second method requires a greater accuracy for the internal iterations, but allows for the use of larger timesteps (Trčka et al., 2010). Trčka et al. (2007) noted that problems with convergence could result from this method, but these could be solved by relaxing the controlling variable; for the testbed, this variable is the coil entering air temperature (or return air temperature). The GenOpt program, as discussed in Section 4.6, was selected for simulation integration and optimization. The testbed, as discussed in Section 4.5, follows the “onion” coupling methodology.

4.5 Testbed Program Structure

Although the purpose of the testbed is to couple a whole building model (EnergyPlus) with a detailed heat pump model (HPSim), a few more program components are necessary for the testbed to properly function. An optimization engine, GenOpt, is involved, as are two data exchange managers. The general testbed program structure is shown in Figure 4.3.

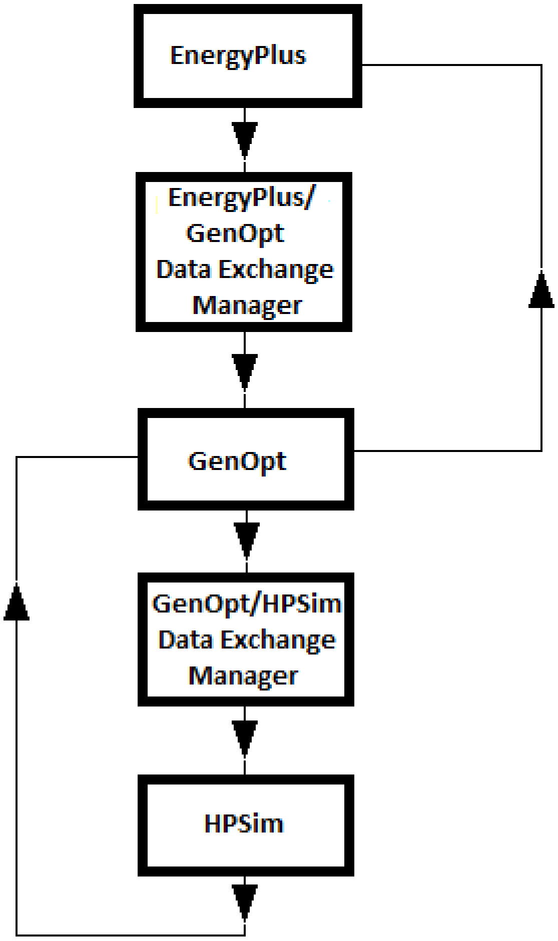

EnergyPlus
EnergyPlus/ GenOpt Data Exchange Manager
GenOpt
GenOpt/HPSim Data Exchange Manager
HPSim

Figure 4.3 Testbed Program Structure

As discussed earlier, EnergyPlus is the main program with calls to Genopt and HPSim as subroutines. Two additional subroutines are also called as data exchange managers. The first subroutine is the EnergyPlus/GenOpt data exchange manager; this is responsible for checking the data coming from EnergyPlus and determining if a call to GenOpt is necessary. The second subroutine is the GenOpt/HPSim data exchange manager; this subroutine calls HPSim and applies a penalty function to HPSim's power if certain conditions are not met. The GenOpt to HPSim loop is one layer of the "onion" coupling structure, iterating within a single EnergyPlus call. The EnergyPlus to GenOpt loop is a second layer, iterating within an EnergyPlus timestep. The testbed calling sequence is shown in Figure 4.4.

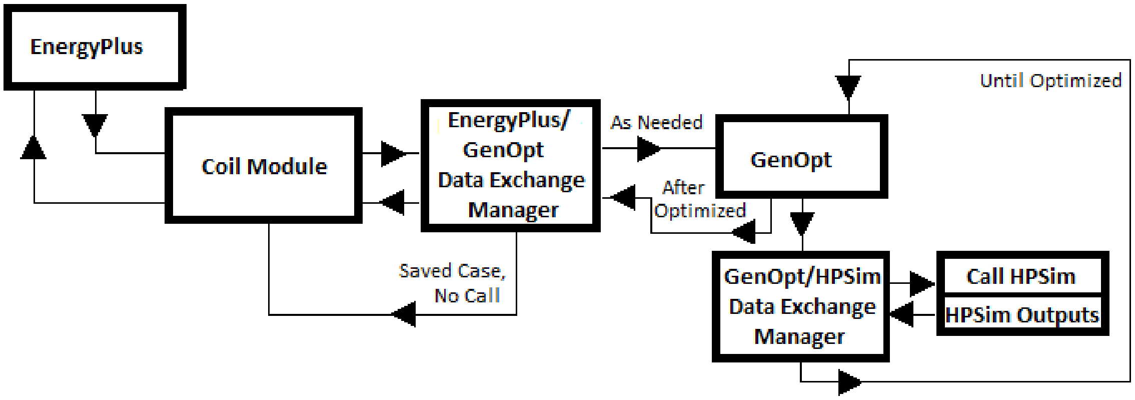


Figure 4.4 Testbed Flow Chart

When EnergyPlus' run reaches the coil module, the call to the EnergyPlus/GenOpt data exchange manager occurs. Figure 4.5 shows the overall testbed architecture, including input and output files to each program.

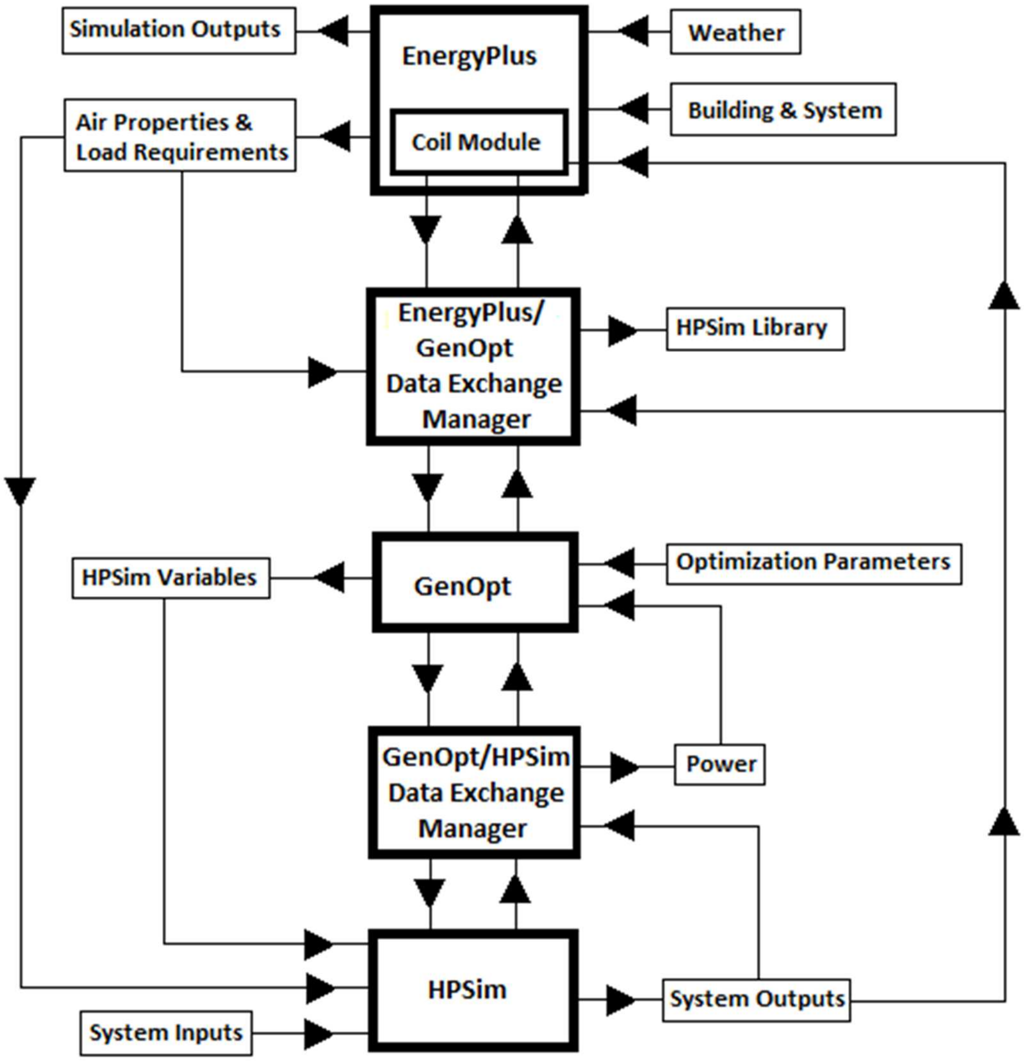


Figure 4.5 Overall Testbed Architecture

4.6 GenOpt Optimization Algorithm

The GPS Hooke Jeeves optimization algorithm is used in GenOpt for this testbed work. GenOpt initializes the optimization by running three different points and then minimizes the objective function (power) through guessing new points. GenOpt then calls the GenOpt/HPSim data exchange manager, which in turn calls HPSim. Once HPSim has finished its run, the GenOpt/HPSim data exchange manager completes its routine and returns to GenOpt. GenOpt is necessary due to the timestep differential; in order to optimize inside the timestep, the control strategy had to be implemented inside the EnergyPlus timestep. The other alternative would have been to create new optimization logic inside the EnergyPlus heat pump routine.

GenOpt has the ability to adjust multiple variables to reduce the objective function. For the testbed, the available variables for optimization were the three variable-speed components: condenser fan, evaporator fan, and compressor. Each of these components can operate at part-load conditions, so they were the best options for optimization. A detailed discussion of the variables is contained in Section 5.1.

4.7 The EnergyPlus/GenOpt Data Exchange Manager

GenOpt's runtime to optimize HPSim is not insignificant; HPSim is called at least four times for each optimization (each iteration calling HPSim and the GenOpt/HPSim data exchange manager). The computer used for this simulation work was a Windows 7 machine with a hard disk drive, running an Intel Core i7-2600K with 16 GB of RAM. In normal operation, HPSim takes around 7 seconds per run. Several cases were run to determine typical computation times for GenOpt when optimizing HPSim. GenOpt takes around 33 seconds for an optimization with four iterations; this is with a step reduction of 1 and a step size of 0.1, with a starting point of 0.9 and a range of 0.8 to 1. GenOpt was restricted to the larger step reduction and step size due to time constraints. With a step reduction of 1 and a step size of 0.01 with a starting point of 0.9 and a range of 0.8 to 1, GenOpt would take around 1 minute and 34 seconds and undergo ten iterations. With a step reduction of 2 and a step size of 0.1 with a starting point of 0.9 and a range of 0.8 to 1, GenOpt would take about 48 seconds and undergo five iterations. Under ordinary circumstances, the data exchange programs' run times are insignificant.

To help minimize the number of times GenOpt is called and the corresponding runtime, the optimized outputs and matching inputs are written to a database as the simulation runs. Before GenOpt is called, the EnergyPlus/GenOpt data exchange manager checks the current set of inputs against the database; if a match is found (or a set within a reasonable margin of error), the optimized outputs are returned to the EnergyPlus/GenOpt data exchange manager and GenOpt is not called. This schema reduces the number of calls inside each system timestep iteration, and eventually can reduce the number of calls over zone timesteps with changes in temperatures and loads. If there is no matching case in the database, the EnergyPlus/GenOpt data exchange manager checks HPSim at the given boundaries of the optimization to ensure that at least one boundary condition is a valid case. An invalid output from both boundaries is indicative that the case cannot be optimized due to the inputs resulting in a non-convergent psychrometric cycle. If at least one boundary returns a valid output, GenOpt is called. If both boundaries return invalid outputs, the data exchange manager adjusts the condenser air flow rate in an attempt to bring the boundaries back into convergence. Figure 4.6 shows the general structure of the EnergyPlus/GenOpt data exchange manager as it calls GenOpt, the GenOpt/HPSim data exchange manager, and HPSim.

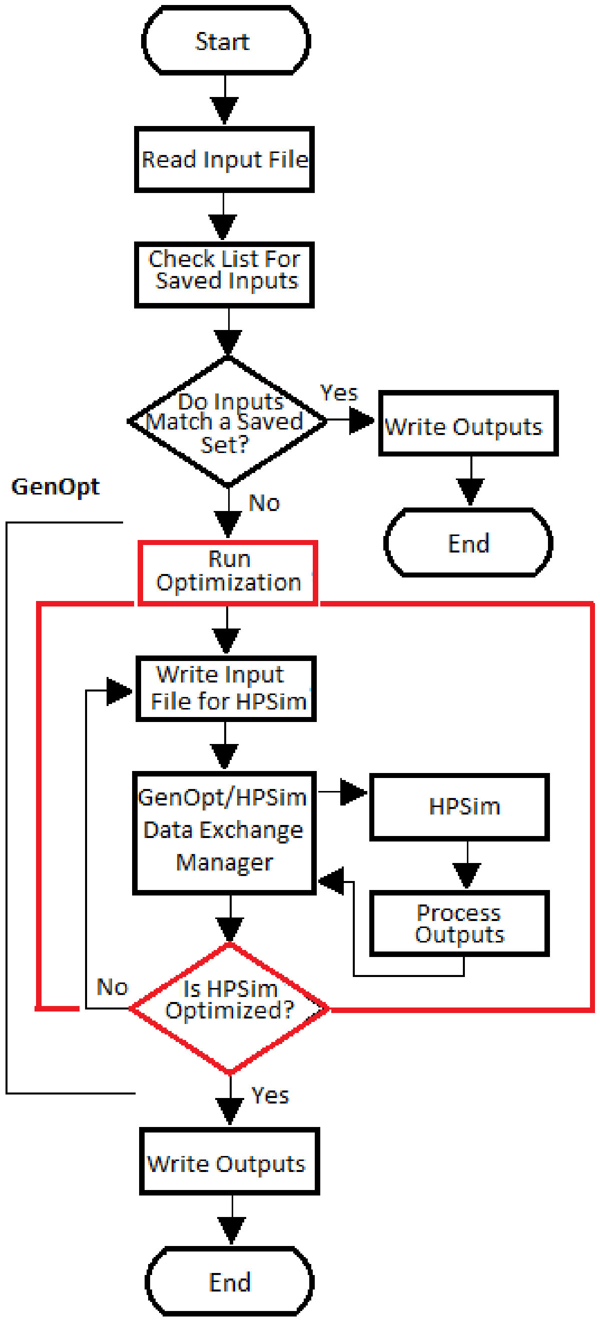


Figure 4.6 EnergyPlus/GenOpt Data Exchange Manager Flow Chart

4.8 The GenOpt/HPSim Data Exchange Manager

The GenOpt/HPSim data exchange manager's purpose is to steer GenOpt away from nonconvergent refrigerant cycle cases by applying a penalty function when the cycle is out-of-bounds. It is called by GenOpt and serves as a go-between for GenOpt and HPSim. It calls

HPSim, reads its outputs, and determines if HPSim crashed, did not converge, or converged on an incomplete refrigerant cycle (where the evaporator outlet conditions do not return to compressor inlet conditions). If HPSim crashed without completing the run or did not converge, an artificially high power (listed as 999999 Btu/h for easy distinction) is returned by the GenOpt/HPSim data exchange manager to steer GenOpt away from the inputs that resulted in the aborted run. If HPSim converged on a case with an incomplete refrigerant cycle, the GenOpt/HPSim data exchange manager calculates a penalty function component to slightly increase the returned power and nudge GenOpt away from the case. In addition, the GenOpt/HPSim data exchange manager calculates a penalty function component if the cooling supplied by HPSim is insufficient to meet the cooling load that EnergyPlus requires. The penalty function is discussed more thoroughly in Section 5.2. At the end of its run, the GenOpt/HPSim data exchange manager writes power (either altered or unaltered) to an output file, which GenOpt uses in its optimization.

4.9 Managing Variable Timesteps

EnergyPlus operates as a quasi-steady state simulation over multiple simulation timesteps, while HPSim operates at a single point in time. GenOpt and the data exchange managers operate outside of the simulation time in conjunction with HPSim. Careful consideration of the timesteps involved was needed for proper operation of the testbed.

The zone air properties are held constant throughout the run of HPSim, and through every run of GenOpt. The entering outdoor temperature ($T_{air_{outdoor\ in}}$) is held constant through every HPSim loop, but the entering indoor temperature ($T_{air_{evap\ in}}$) is one of the variables adjusted for convergence. Each internal HPSim loop starts with $T_{air_{evap\ in}}$ at the set temperature, adjusts for convergence, and tries to come back to its initial set point; $T_{air_{evap\ in}}$ has to come back to the setpoint at the very end of the HPSim run for successful convergence. Section 4.8 discussed the case where HPSim does not converge. Every time EnergyPlus issues a call to the

EnergyPlus/GenOpt data exchange manager, the zone air properties (as well as the outdoor air properties) for HPSim are updated.

EnergyPlus can run multi-year simulations, which would require HPSim to be called many times over the run. Figure 4.7 shows the different time scales that HPSim and EnergyPlus operate over, as well as the timestep for which weather is recorded.

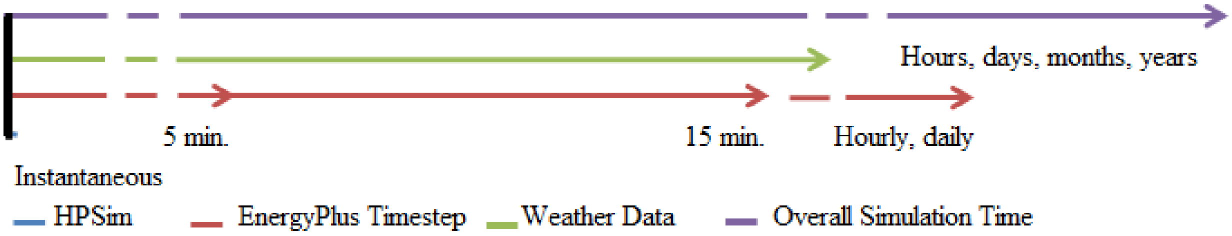


Figure 4.7 Simulation Timestep Scales

HPSim operates at an instantaneous timestep, while EnergyPlus operates at a range of timesteps. The weather data is reported in hourly timesteps. The overall simulation time itself ranges from hours to years. The interaction between the timesteps is shown in Figure 4.8.

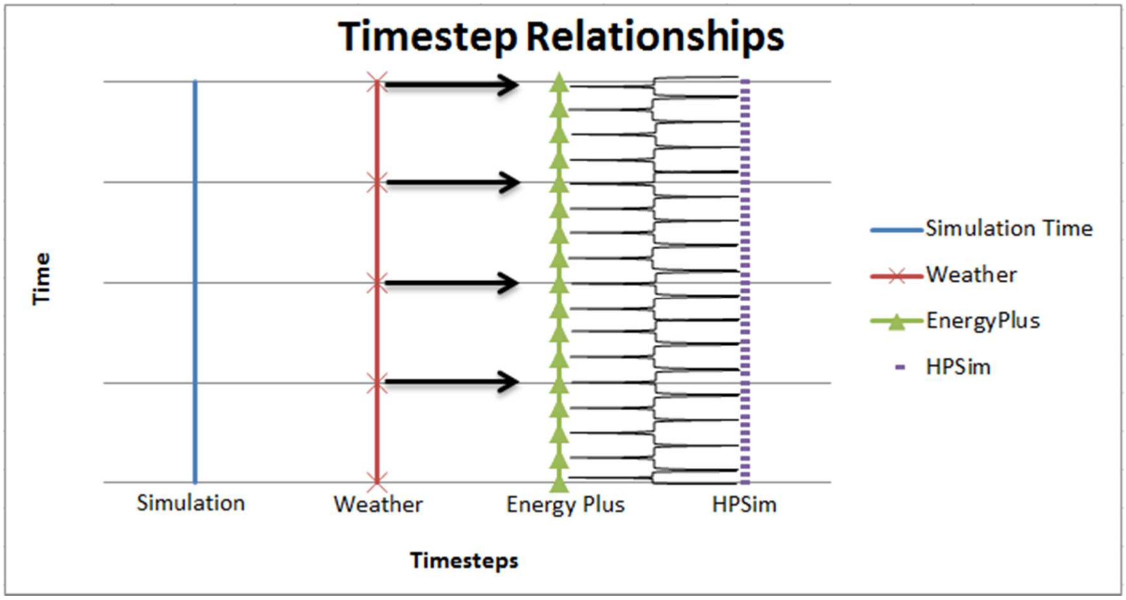


Figure 4.8 Simulation Timestep Interactions

EnergyPlus receives updated weather information every weather timestep. Inside each EnergyPlus zone timestep can be multiple simulation timesteps, which is where HPSim is called. The simulation timestep varies based on calculation requirements and iterates inside the timestep; this results in multiple calls of HPSim. The overall simulation time is user-set and can be any length.

4.10 Routine Considerations

In addition to the coupling point location and structure, there were a number of small but significant considerations dealt with in order for the testbed to operate successfully.

4.10.1 Minimizing Solution Time

Since HPSim is a steady state model at a single point in time, the testbed is only called every system timestep. However, even if the largest system timestep is exclusively used – one hour – the EnergyPlus/GenOpt data exchange manager will be called 24 times to simulate a single day, and 8,760 times for an annual simulation; this does not take into account multiple iterations with a system timestep. If GenOpt is called every time the EnergyPlus/GenOpt data exchange manager is, then HPSim will be called at least four times for each optimization. This would result in HPSim being called 384 times for a single day simulation and over 104,000 times for an annual simulation. As previously seen in Section 4.7, HPSim has a measurable run time, so the testbed run time increases every instance HPSim is called; GenOpt also takes up precious calculation time each call. It was therefore imperative for HPSim to run over as little time as possible while still being accurate; the modifications required to make the code run efficiently and quickly are covered in Section 3.5.1. Also previously discussed in Section 4.7 were the steps taken by the EnergyPlus/GenOpt data exchange manager to reduce the number of calls to GenOpt and HPSim.

The overall simulation runtime was additionally decreased through use of a simple EnergyPlus curve-fit heat pump model during the warm-up days, allowing HPSim to be called only during the full simulation. Another method employed for the reduction of simulation runtime was a deadband allowing a small temperature fluctuation from the setpoint during optimization (+/- 1.8 °F/1 °C). This represented thermostat insensitivity and allowed the zone temperature to "float" a little without requiring cooling.

4.10.2 Ensuring Solution Convergence

The GenOpt/HPSim data exchange manager was designed to make the simulation more robust. It protects GenOpt in two ways: reporting the power variable even when HPSim has crashed and steering GenOpt away from non-convergent psychrometric cycles. If the inputs provided by GenOpt cause the heat pump cycle to operate out of range, HPSim can catastrophically fail and provide no outputs; in this case, the GenOpt/HPSim data exchanger manager will return a very large power value to GenOpt. This keeps GenOpt from converging on the failed point. Similarly, if HPSim completes its run for an out of range cycle, the GenOpt/HPSim data exchanger manager adds penalty factors to the returned power value. These penalty factors and the overall penalty function are discussed in Section 5.2.

CHAPTER V

OPTIMIZATION ROUTINES

In addition to the previously discussed optimization programs, there are two more pieces critical for proper functioning of the testbed. These pieces are the variables adjusted for optimization and the penalty function, as this chapter discusses.

5.1 Optimization Variables

Because HPSim is a refrigerant-based model and does not allow a setpoint for the evaporator leaving air temperature (e.g. the air leaving the coil and entering the room), the air temperature cannot be directly controlled. There are three variables that can be adjusted: the evaporator air flow rate (air-side of system), the condenser air flow rate (air-side of system), and the compressor ratio (refrigerant-side of system). The following sections discuss how control strategies based on these variables impact optimization.

5.1.1 Evaporator Air Flow Rate

The first option is changing the evaporator air flow rate ($\dot{m}_{air_{evap}}$). This affects the evaporator leaving air temperature ($T_{air_{evap\ out}}$), as seen by a simple "dry coil" heat balance in Equation 5.1.

$$T_{air_{evap\ out}} = T_{air_{evap\ in}} - \frac{\dot{Q}_{evap}}{c_p * \dot{m}_{air_{evap}}} \qquad (5.1)$$

If $\dot{m}_{air_{evap}}$ is varied, the inlet air temperature ($T_{air_{evap\ in}}$) held constant, and the heat transfer rate ($\dot{Q}_{evap}$) set by the refrigerant-side, then $T_{air_{evap\ out}}$ will also vary. As such, $T_{air_{evap\ out}}$ is free to vary while HPSim converges on the refrigerant-side by adjusting $\dot{Q}_{evap}$. Adjusting $\dot{m}_{air_{evap}}$ to minimize power is, on its own, a poor control strategy since $T_{air_{evap\ out}}$ cannot be controlled and thus a specified setpoint cannot be met. Therefore, another dimension of the control strategy had to be added.

In his investigation of evaporator air flow optimization, Gall (2015) added a secondary setpoint control to address this. Gall controlled cooling rate by varying the evaporator fan up to full-speed as cooling was required. If the cooling requirements were still not met, adjustment of the compressor ratio was added, dependent on the air temperature leaving the evaporator coil compared to the leaving air temperature setpoint. However, neither of these controls accounted for the room setpoint; therefore, Gall added in a secondary control, which adjusted the evaporator leaving air setpoint if the return air temperature did not meet the room setpoint. For this testbed, the need for a secondary control was addressed by the penalty function, which is discussed in Section 5.2.

Although the dry coil assumption for heat transfer is not strictly accurate, it was a simplification for the proof of concept work. The heat transfer equations should be expanded to include wet coil conditions as part of future work.

5.1.2 Condenser Air Flow Rate

Secondly, the condenser air flow rate ($\dot{m}_{air_{cond}}$) can be adjusted. The variable most affected by this adjustment is the condenser leaving air temperature ($T_{air_{cond\ out}}$) for the same reasons $T_{air_{evap\ out}}$ was affected by changing $\dot{m}_{air_{evap}}$. This method has the additional advantage of the air leaving the condenser being outdoors and thus the outlet temperature is of little concern to the system. However, there is another side that should be considered – the effect of $\dot{m}_{air_{cond}}$

variation on $T_{air_{evap\ out}}$. This can be investigated by the consideration and manipulation of the capacity Equations 5.2, 5.3, and 5.4.

$$\dot{Q} = \dot{m}_{air} * cp * \Delta T_{air} \qquad (5.2)$$

$$\dot{Q} = \dot{m}_{ref} * \Delta h \qquad (5.3)$$

For an air-cooled coil, assuming the air results in a dry coil and that any heat added by the fan is upstream of the coil:

$$\dot{Q}_{coil} = (\dot{m}_{air} * cp * \Delta T_{air})_{coil} = (\dot{m}_{ref} * \Delta h)_{coil} \qquad (5.4)$$

Since the heat transferred to the refrigerant by the compressor and absorbed by the evaporator must be rejected by the condenser, the heat the condenser is rejecting is:

$$\dot{Q}_{cond} = \dot{Q}_{evap} + \dot{Q}_{comp} \qquad (5.5)$$

Taking the refrigerant rate of heat transfer of the compressor, assuming $\dot{Q}_{evap}$ has a dry coil condition, and knowing that the air and refrigerant rates of heat transfer for the evaporator are equal:

$$(\dot{m}_{air} * \Delta \mathrm{T}_{air} * cp)_{cond} = (\dot{m}_{air} * \Delta \mathrm{T}_{air} * cp)_{evap} + (\dot{m}_{ref} * \Delta h)_{comp} \qquad (5.6)$$

Assuming the specific heat of air is essentially the same for the condenser air and the evaporator air and rearranging:

$$(\dot{m}_{air} * \Delta \mathrm{T}_{air})_{evap} = (\dot{m}_{air} * \Delta \mathrm{T}_{air})_{cond} + \frac{(\dot{m}_{ref} * \Delta h)_{comp}}{cp} \qquad (5.7)$$

When looking for a generalized relationship, the constant or quasi-constant variables can be ignored. The constant variables here are the evaporator air mass flow rate ($\dot{m}_{air_{evap}}$), the specific heat of air (cp), and the compressor capacity ($(\dot{m}_{ref} * \Delta h)_{comp}$).

$$\Delta \mathrm{T}_{air_{evap}} \propto (\dot{m}_{air} * \Delta \mathrm{T}_{air})_{cond} \qquad (5.8)$$

Solving for $\mathrm{T}_{air_{evap\ out}}$:

$$\mathrm{T}_{air_{evap\ out}} \propto (\dot{m}_{air} * \Delta \mathrm{T}_{air})_{cond} - \mathrm{T}_{air_{evap\ in}} \qquad (5.9)$$

$\mathrm{T}_{air_{evap\ in}}$ is independent of HPSim calculations and thus can be considered constant. $\dot{m}_{air_{cond}}$ is varying, which means $\mathrm{T}_{air_{evap\ out}}$ (and thus the change in the air temperature) varies with regard to the capacity of the condenser. However, the capacity of the condenser on the air-side must be equal to the capacity of the condenser on the refrigerant-side, or $(\dot{m}_{ref} * \Delta h)_{cond} = (\dot{m}_{air} * \Delta \mathrm{T}_{air})_{cond}$. Since the refrigerant capacity will only change a little based on the refrigerant cycle location (and such changes are needed by the iterative process to find a good cycle), this means it is relatively constant. The air-side capacity is also relatively constant and thus $\mathrm{T}_{air_{evap\ out}}$will remain fairly constant.

Therefore, it appears adjusting $\dot{m}_{air_{cond}}$ should not have a significant effect on the evaporator side and thus, while it may be useful for ensuring the vapor-compression cycle is valid and decreasing power, it does not seem that varying it would be of much use in trying to control $\mathrm{T}_{air_{evap\ out}}$. Gall (2015) concurs, noting that optimizing on the condenser fan speed offers no meaningful improvement.

5.1.3 Compressor Ratio

Thirdly, the compressor ratio – representing the percentage of time in a cycle that the compressor is on – can be adjusted. The compressor sets the refrigerant flow rate ($\dot{m}_{ref}$), therefore adjusting the compressor ratio affects the entire system: the maximum $\dot{m}_{ref}$ will be multiplied by the compressor ratio to give $\dot{m}_{ref}$ over the cycle time. From Equation 5.1, it is seen that, if $\mathrm{T}_{air_{evap\ in}}$

and cp are constant, $\mathrm{T}_{air_{evap\ out}}$ varies with respect to $\dot{Q}_{evap}$ and $\dot{m}_{air_{evap}}$. Knowing $\dot{Q}_{air}$ is equal to $\dot{Q}_{ref}$ across a coil, and assuming a dry coil condition,

$$\mathrm{T}_{air_{evap\ out}} = \mathrm{T}_{air_{evap\ in}} - \frac{(\dot{m}_{ref} * \Delta h)_{evap}}{cp * \dot{m}_{air_{evap}}} \tag{5.10}$$

If only the compressor ratio is varying, $\dot{m}_{air_{evap}}$ is constant and thus $\mathrm{T}_{air_{evap\ out}}$ changes only with respect to the heat transfer across the coil ($(\dot{m}_{ref} * \Delta h)_{evap}$). The change in enthalpy over the evaporator (Δh_{evap}) is dependent on state variables, and will be somewhat impacted by the change in $\dot{m}_{ref}$; however, there is no direct inverse correlation between $\dot{m}_{ref}$ and Δh_{evap} to assume their multiplication would produce a constant value. Therefore, $\dot{m}_{ref}$ and $\mathrm{T}_{air_{evap\ out}}$ can be seen to have an inverse relationship; as $\dot{m}_{ref}$ increases, $\mathrm{T}_{air_{evap\ out}}$ will decrease and vice-versa.

However, the same issue arises with the compressor ratio variable as arose with the evaporator fan ratio: neither GenOpt, which sets the compressor ratio variable, nor HPSim, which calculates $\mathrm{T}_{air_{evap\ out}}$, have any knowledge of EnergyPlus' cooling load. There is no means for GenOpt (or HPSim) to adjust the compressor ratio or other inputs in direct response to EnergyPlus' cooling load. As Gall (2015) noted, in order for both power to be minimized and the cooling load met, there is a need for a secondary control based off the room setpoint. For the testbed, this secondary control took the part of a penalty function as discussed in Section 5.2.

5.2 Penalty Function

The purpose of the penalty function is to guide GenOpt's optimization by adjusting the objective function (power minimization) based on HPSim's calculated performance. The objective function has the form:

$$Power = y_1 + y_2 + y_3 + y_4 \tag{5.11}$$

Where the components are:

$$y_1 = 0.9 * (\Delta T_{sat_{cond}})^2 \qquad (5.12)$$

and $\Delta T_{sat_{cond}}$ is defined in equation 5.20.

$$y_2 = Power_{HPSim} \qquad (5.13)$$

$$y_3 = 0.9 * (\Delta T_{sat_{evap}})^2 \qquad (5.14)$$

and $\Delta T_{sat_{evap}}$ is explained in equations 5.21 and 5.22.

$$y_4 = F_{penalty} * (\Delta T_{air})^2 \qquad (5.15)$$

and ΔT_{air} is shown in equation 5.34.

y_1, y_3, and y_4 are the penalty function components; y_2 is the unmodified power reported by HPSim. y_1 is the penalty function component for the condenser saturation temperature, y_3 is the penalty function component for the evaporator saturation temperature, and y_4 is the penalty function component for the cooling rate supplied by the heat pump. Since power has units of Watts, each y-term is in Watts. y_1, y_3, and y_4 are ideally zero, meaning that the power reported to GenOpt would be just y_2 – the power calculated by HPSim.

A significant item in creating the penalty function was the consideration of how much weight each component should be given. The power, in Watts, has an order of magnitude of a thousand, but differences in temperature have an order of ones or tens; squaring the differences result in orders of magnitude of tens or potentially hundreds. The multiplicative factor of the penalty components was chosen as 0.9, which slightly reduced their weight without losing their significance.

The required heat transfer rate to bring the zone temperature to the setpoint, $\dot{Q}_{setpoint}$, is calculated in EnergyPlus before the heat pump simulation is called. $\dot{Q}_{setpoint}$ is written to a text file and then used in the GenOpt/HPSim data exchange manager as part of y_4 to help guide the optimization to the lowest power rate that still allows for the heat transfer rate to be met. The heat transfer provided by HPSim at the provided timestep conditions is also reported to the GenOpt/HPSim data exchange manager for this calculation.

The following sections give details on the development of the penalty function components. Note that all of these values can – and should – be adjusted based on the specific system and as part of the controls strategy.

5.2.1 Condenser Penalty Component

The condenser penalty function is concerned with keeping condenser refrigerant from having too high or too low of a pressure. A range of differences in temperature between the saturation temperature of the refrigerant ($T_{ref_{cond\ out\ sat}}$, °C) and $T_{air_{cond\ in}}$ approximated the appropriate pressure range. In the penalty factor, this is reflected by requiring $T_{ref_{cond\ out\ sat}}$ to stay within an acceptable temperature range (dependent on $T_{air_{cond\ in}}$) to avoid impacting the penalty function. Note that HPSim and the penalty function use Fahrenheit temperatures, though any temperature outputs to EnergyPlus are converted to Celsius. The upper bound of the range is described by the following equation:

$$Upper\ Bound = T_{air_{cond\ in}} + 25\ \text{°F} \qquad (5.16)$$

If the condensing temperature is too high, it affects the compressor by returning a high pressure. For the heat pump modeled in this work – a two-ton, light-commercial unit – 25 °F was chosen as an appropriate maximum change in temperature to control the pressure. If $T_{ref_{cond\ out\ sat}}$ is greater than the upper bound, then

$$\Delta T = T_{ref_{cond\,out}}\,sat - 25\ \text{°F} \qquad (5.17)$$

The lower bound of the temperature range is described by the following equation:

$$Lower\ Bound = T_{air_{cond\,in}} + 10\ \text{°F} \qquad (5.18)$$

If the lower bound is too close to $T_{air_{cond\,in}}$, the heat transfer rate will be inefficient; 10 °F was chosen as an appropriate minimum change in temperature for heat transfer for the two-ton unit modeled. If $T_{ref_{cond\,out\,sat}}$ is less than the lower bound, then

$$\Delta T = 10\ \text{°F} - T_{ref_{cond\,out\,sat}} \qquad (5.19)$$

If $T_{ref_{cond\,out\,sat}}$ is between the upper and lower bounds, ΔT is 0 and the penalty function component is not applied. If the penalty component is applied, it is multiplied by a scalar of 0.9, resulting in a linear function.

5.2.2 Evaporator Penalty Component

The evaporator penalty component was determined through an exercise investigating the coil operation. With the assumption of a dry coil, the temperature of the air leaving the evaporator coil is calculated by

$$T_{air_{out}} = \frac{\dot{Q}_{coil}}{cp_{air} * \dot{m}_{air_{coil}}} + T_{air_{in}} \qquad (5.20)$$

Where, $\dot{Q}_{coil}$ is the rate of heat transfer across the coil (kW), cp_{air} is the specific heat of air across the coil (kJ/kg-K), $\dot{m}_{air_{coil}}$ is the mass flow rate of the air entering the coil (kg/s), $T_{air_{in}}$ is the temperature of the air entering the coil (°C), and $T_{air_{out}}$ is the temperature of the air leaving the coil (°C). If $T_{air_{in}}$ and $T_{air_{out}}$ are both constant, $\frac{\dot{Q}_{coil}}{c_{p_{air}} * \dot{m}_{air_{coil}}}$ must also be constant. With the assumption that any change in $c_{p_{air}}$ is negligible, if $\dot{m}_{air_{coil}}$ is increased, $\dot{Q}_{coil}$ must also increase, and if $\dot{m}_{air_{coil}}$ is decreased, $\dot{Q}_{coil}$ must also decrease. Because the evaporator is rejecting heat to the refrigerant, $T_{air_{out}}$ must be less than $T_{air_{in}}$, and thus $\dot{Q}_{coil}$ has a negative sign. In order for the refrigerant to receive the rejected heat, the refrigerant temperature – and thus the entering refrigerant saturation temperature ($T_{ref_{evap\ in\ sat}}$, °F) of the two-phase liquid and gas mixture – must be colder than $T_{air_{out}}$. Therefore, $T_{ref_{evap\ in\ sat}}$ must be less than $T_{air_{in}}$.

The entering refrigerant evaporator temperature ($T_{ref_{evap\ in}}$, °F) is set by the rest of the cycle and thus can be considered a dependent variable. Because entering air will usually contain at least trace amounts of moisture, $T_{ref_{evap\ in\ sat}}$ must be greater than the freezing point of water (32 °F) to keep frost from accumulating on the coil; 35 °F was chosen to allow for a margin of safety of a few degrees above freezing. In addition, to keep the relative humidity of the air at an acceptable level, there must be an upper limit for the saturation temperature as well; this has been chosen as 55 °F, a typical temperature for air leaving the coil. If $T_{ref_{evap\ in\ sat}}$ is below 35 °F, then the penalty function is

$$y_3 = 0.9 * (T_{ref_{evap\ in\ sat}} - 35°\text{F})^2 \qquad (5.21)$$

Similarly, if $T_{ref_{evap\ in\ sat}}$ is above 55 °F, the penalty function is

$$y_3 = 0.9 * (55°\text{F} - T_{ref_{evap\ in\ sat}})^2 \qquad (5.22)$$

If $T_{ref_{evap\ in\ sat}}$ is between 35 and 55 °F, there is no penalty. This penalty function is linear. A note that the 35 and 55 °F may require adjustment for heat pumps other than the one modeled.

5.2.3 Room Return Air Temperature Penalty Function

There was a dearth of information on return air temperature penalty functions, but research from both Gayeski (2010; et al., 2011) and Zakula (2013) was of great help in this area. Gayeski (2010) investigated predictive cooling strategies for low lift cooling systems. His objective function was the sum of the cooling energy consumed, the weighted operative temperature penalty function, and the evaporating temperature penalty function, as shown:

$$F_{obj} = \Sigma((r * P_{cool}) + \left(\varphi * P_{pen_{opt}}\right) + P_{pen_{evp}}) \qquad (5.23)$$

Where F_{obj} is the objective function to be minimized, *r* is a weighting factor for the system cooling power, P_{cool} is the system cooling power, φ is the operative temperature penalty function weighting, $P_{pen_{opt}}$ is the operative temperature penalty function, and $P_{pen_{evp}}$ is the penalty function for the evaporator temperature. $P_{pen_{evp}}$'s purpose is to prevent the chiller from freezing.

The components of the operative temperature penalty function, φ and $P_{pen_{opt}}$ are of the most interest in this work. Gayeski (2010) gave three equations for $P_{pen_{opt}}$, depending on where the operative temperature (T_{op}) was with respect to the acceptable operating range. If T_{op} is less than or equal to the sum of the minimum operative temperature (T_{min}) in degrees Kelvin plus 0.5, then:

$$P_{pen_{opt}} = ((T_{min} + 0.5) - T_{op})^2 \qquad (5.24)$$

If T_{op} is greater than the sum of T_{min} and 0.5 but less than the value of the maximum operative temperature (T_{max}) minus 0.5, then $P_{pen_{opt}}$ is zero. If T_{op} is greater or equal to the value of T_{max} plus 0.5, then:

$$P_{pen_{opt}} = \left(T_{op} - \ (T_{max} - 0.5)\right)^2 \qquad (5.25)$$

One interesting aspect of Gayeski's penalty function is that, on the low side, it was applied before the temperature left the acceptable range. Gayeski (2010) specifically mentioned that φ was chosen so that if T_{op} was outside of range by 0.5 K or more, $P_{pen_{opt}}$ would be larger than P_{cool}, forcing the system to run in cooling mode. Using that reasoning, by the time T_{op} had exceeded the operating range limits, the system would already be cooling.

Additionally, Gayeski et al. (2011) stated that the quadratic form of the penalty function was chosen because it gave a continuous derivative. He also reiterated the decision to weight the function so that it was more cost-effective to run the cooling equipment than to let the temperature rise once it was outside of the comfort range by more than half a degree.

In her dissertation, Zakula (2013) described her research into predictive controls for buildings and equipment. The controls were optimized to minimize energy use, and a penalty function was used to keep the building within comfort levels. The objective function was a sum of the power required to cool the building (both the power used directly by the cooling equipment and the power used to distribute the cooling) and a penalty component calculated based on the air temperature. Her penalty function built on Gayeski's (2010) work, but her penalty function limits were dependent on if the operative temperature was strictly greater than the upper limit or less than the lower limit. However, her penalty factor still carried a greater weight than the cooling power if the operative temperature was outside of the temperature limits by more than about 0.5 °C. The penalty function Zakula used was of the following form:

$$T_{penalty} = F_{penalty} * (\Delta T)^2 \qquad (5.26)$$

$T_{penalty}$ is the penalty component in the objective function sum; it is only applied when T_{op} is greater than the upper temperature limit or less than the lower temperature limit. The temperature difference is calculated between T_{op} and the closer temperature limit. If T_{op} is greater than the upper limit, then:

$$\Delta T = T_{op} - T_{upper\ limit} \qquad (5.27)$$

If T_{op} is less than the lower limit, then the temperature difference has the following form:

$$\Delta T = T_{lower\ limit} - T_{op} \qquad (5.28)$$

$F_{penalty}$ is a multiplier to apply an appropriate weight to the penalty component. According to Gayeski (2010, via Zakula, 2013), the multiplier must be sufficiently large so that less power is required to cool the building than would be to let the temperature stay out of the comfort range. Following Gayeski's and Zakula's lead, the penalty factor developed for this case is:

$$y_4 = F_{penalty} * (\Delta T)^2 \qquad (5.29)$$

where y_4 is the fourth component of the overall penalty function ($T_{penalty}$ in Equation 5.26) and $F_{penalty}$ is the penalty factor. Continuing to follow Gayeski and Zakula, $F_{penalty}$ was weighted so that y_4 was about equal to the heat pump power when T_{op} is outside of the comfort range by about 1 °F/0.56 °C. Solving for $F_{penalty}$,

$$F_{penalty} = \frac{T_{penalty}}{(\Delta T)^2} \qquad (5.30)$$

The power required by the heat pump to operate varies according to the cycle it is running, which in turn varies according to the input conditions. Nevertheless, the power required is generally between 1000 and 1200 W. Using the smaller power of 1000 W for $T_{penalty}$ and a ΔT of 1 °F for the calculation,

$$F_{penalty} = 1000 \frac{W}{°F^2} \qquad (5.31)$$

This results in an updated Equation 5.29 of

$$y_4 = 1000 \frac{W}{°F^2} * (\Delta T)^2 \qquad (5.32)$$

ΔT is the difference between the setpoint temperature ($T_{setpoint}$, °F) and T_{RA} (°F; the temperature of the air leaving the zone). Where $T_{setpoint}$ is known and T_{RA} is calculated based on a) the cooling load required by the space, b) the air flow rate set by HPSim, and c) the leaving coil air temperature. This requires the assumption that, between one system timestep and the next, the internal and envelope loads in the space will not change significantly. Essentially, if a specific rate of heat transfer is required to cool the room at the first minute, the same rate of heat transfer will be required to cool the room at the second minute. From a practical perspective, this is reasonable – the cooling provided in the first minute will not cool the entire room by the second minute. Note that if the system timestep is larger or if a zone timestep is used, the T_{RA} calculated in the first iteration will be inaccurate due to a change in the internal and envelope loads. However, due to EnergyPlus' iterative process, T_{RA} should be corrected by the second iteration when the cooling rate required is updated. With this assumption, and the assumption of a dry coil, it is reasonable to determine T_{RA} by the following calculation:

$$T_{RA} = \frac{\dot{Q}_{zone}}{c_{p_{air}} * \dot{m}_{air}} + T_{zone_{in}} \qquad (5.33)$$

Therefore,

$$y_4 = 1000 \frac{W}{°F^2} * (T_{setpoint} - T_{RA})^2 \qquad (5.34)$$

With only one setpoint value instead of a deadband range, there is only one equation for y_4. If T_{RA} is larger than $T_{setpoint}$, the squaring of the temperatures will remove the negative. Note that the penalty function is only applied when the rate of cooling provided by HPSim is less than the rate of cooling EnergyPlus requires to meet the setpoint.

A final note on the above y_4 – this also assumes a compressor ratio modulating between 0 and 1. If the compressor ratio modulation is constrained, the $T_{penalty}$ component of Equation 5.30 should change accordingly.

CHAPTER VI

TESTBED VERIFICATION

This chapter contains the results showing proof of concept for this work. Several tests showing balances across heat pump components are discussed. HPSim's sensitivity to the optimization variables is also investigated, as tests proving HPSim's and EnergyPlus's connectivity. Finally, operational testbed results for a baseline case are shown.

6.1 Balance Tests

A number of tests were run to show the heat balanced over the primary HPSim components. These components were the compressor, condenser, and evaporator. The expansion device operates adiabatically and thus no heat transfer balance could be accounted over it. All three balances are for a test case with a compressor ratio of 0.8, a condenser air flow rate of 4000 CFM, and an evaporator air flow rate of 800 CFM. The outdoor air conditions were 69.8 °F/21°C drybulb and 63.9 °F/17.7 °C wetbulb; indoor air conditions were 75.2 °F/24 °C drybulb and 50.5 °F/10.3°C wetbulb. These conditions result in wet coil conditions, though with little moisture.

6.1.1 Potential Errors

Heat capacities are, by nature, very dependent on the calculated properties. On the air side, the heat transfer equation greatly depends on the change in temperature across the coil. The evaporator air outlet temperature is calculated and the inlet temperature is adjusted for calculation, with the intention that it returns to the input value before the end of the simulation. Failure to converge would result in non-steady conditions and an error in the heat balance equation. In addition, the evaporator fan now contributes heat to the air stream. In HPSim, the evaporator fan is modeled after the evaporator coil calculations have finished, so the fan heat added to the air is not accounted for in the refrigerant-air heat balance.

On the refrigerant side, the impact of non-steady conditions is a bit harder to see, but consider the refrigerant heat transfer equation

$$\dot{Q}_{coil} = (\dot{m}_{ref} * \Delta h)_{coil} \qquad (6.1)$$

It is easily seen that as $\dot{m}_{ref}$ increases, Δh has a greater impact on the heat transfer rate and vice-versa. Therefore, a deviation in Δh – especially in subcooling or superheat – can have a large impact on the overall refrigerant heat transfer rate.

HPSim determines the air side latent rate of heat transfer simply by taking the difference of the total and sensible rates of heat transfer. The total rate of heat transfer is a refrigerant side calculation, while the sensible rate of heat transfer is an air side calculation, so there is potential for an error to be introduced. Additionally, HPSim ignores any heat lost by the condensate discharge.

In addition, there are deviations due to conversion calculations, rounding errors, and psychrometric routines. The latter has a particular impact, especially in extreme air cases: If very dry air (with near-zero humidity ratio) is used as an input to HPSim, the psychrometric routines can produce faulty results. In one instance, the fan added heat to very dry air, increasing the air's enthalpy. This enthalpy was used by a psychrometric routine to calculate the leaving air temperature; the routine returned a lower air temperature than entered the fan. It is also possible that very wet air (with a very high humidity ratio) could cause problems for psychrometric calculations, but this has not been investigated.

Lastly, there is an inherent solver issue wherein the final reported case is not necessarily the converged case. Instead, the final reported case is the run immediately prior to the converged case. This is due to the convergent values being guessed for the next iteration, but – because they are convergent – the iteration loops reach convergence and exit out before actually running the convergent iteration. Although the exact impact is hard to quantify, this is a reasonable cause of errors in the results.

6.1.2 Compressor Heat Balance

The test case was previously described in Section 6.1. The compressor has refrigerant enter at superheated vapor and then uses power to apply heat to the refrigerant, compressing it into an even more superheated vapor state. The heat balance is shown by considering the power used by the compressor and the change in enthalpy across the compressor. The heat balance is shown in Figure 6.1.

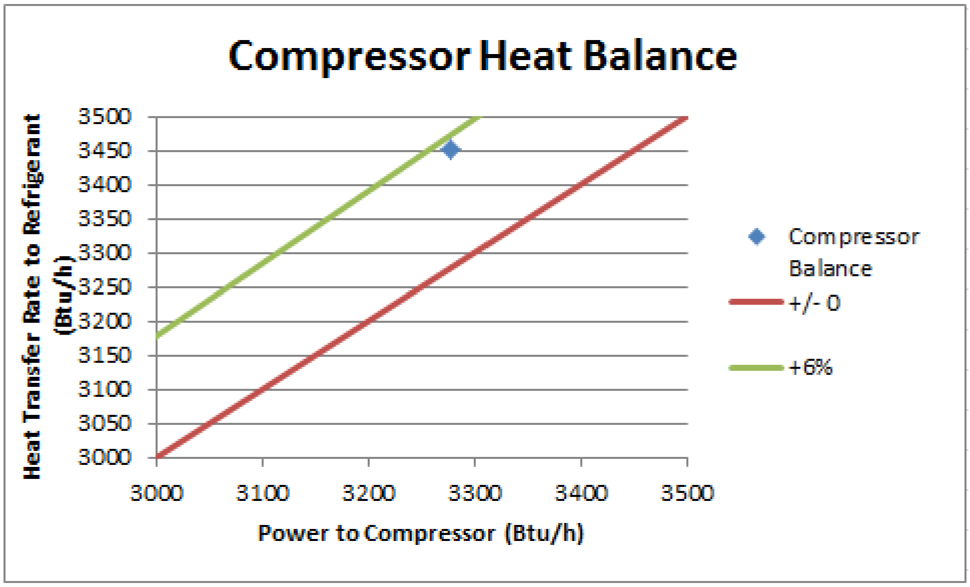


Figure 6.1 Compressor Heat Balance

The heat transferred to the refrigerant is within 6% of the power used by the compressor. Power is calculated using a map – full-load power is determined by a map based on refrigerant temperatures and map coefficients – so a mismatch in the map variables could cause the error. In addition, the full-load power is weighted by the compressor ratio to determine the part-load power, so a variable mismatch could also be at fault.

However, a system-wide consideration of the refrigerant heat transfer rates revealed a good agreement. The rejected heat transfer rate of the condenser was compared against the heat transfer rate added to the refrigerant in the compressor and evaporator. This balance is seen in Figure 6.2.

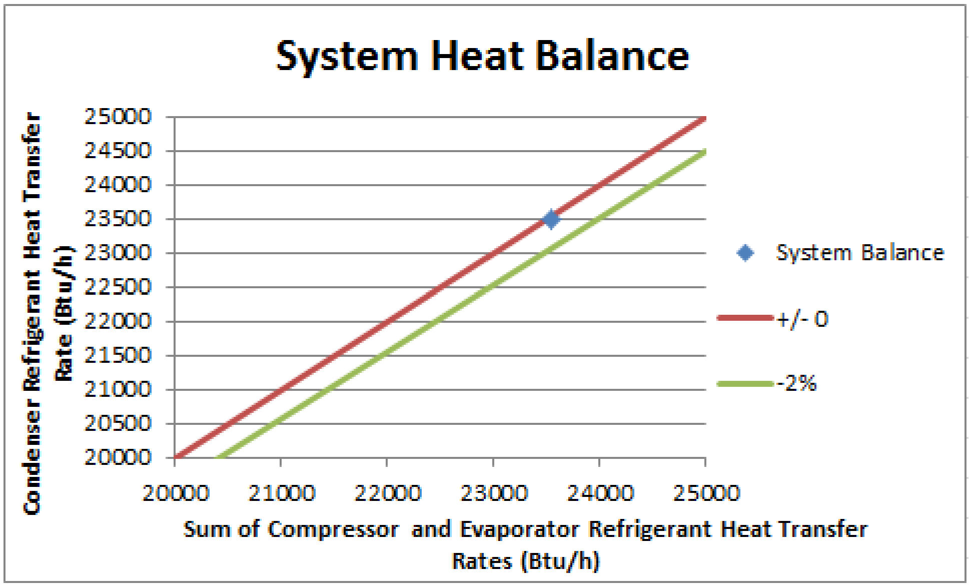


Figure 6.2 System Heat Balance

The overall error is less than half-a-percent, completely insignificant, and entirely within the realm of rounding errors, conversion errors, and line heat loss.

6.1.3 Condenser Heat Balance

For the condenser model, the heat balance was taken across the refrigerant and air sides. The change in enthalpy was used to calculate the refrigerant heat transfer rate, and changes in air temperature and humidity ratio were used to calculate the air heat transfer rate. The heat balance is shown in Figure 6.3.

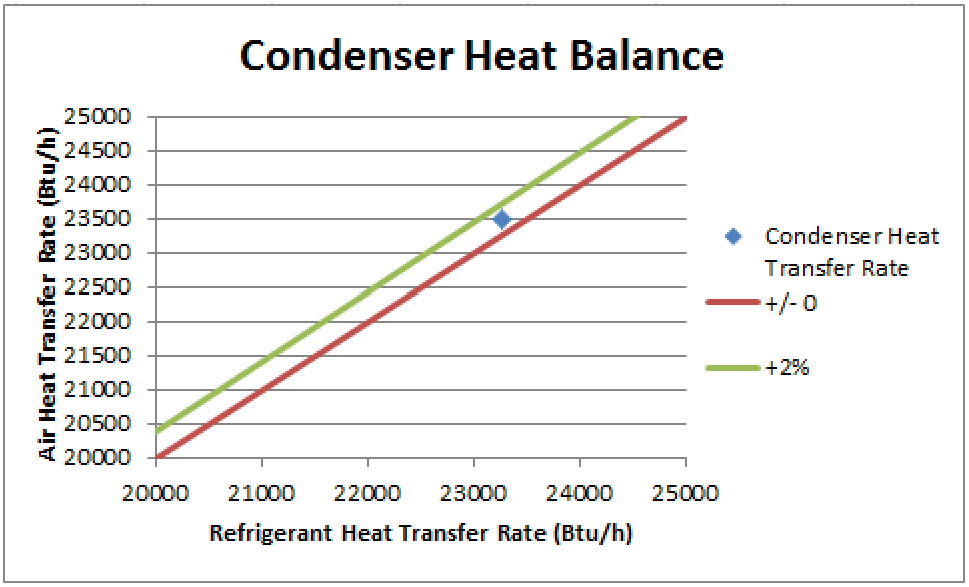


Figure 6.3 Condenser Heat Balance

The condenser shows a close agreement between the refrigerant and air heat transfer rates, with less about a 1% error. This error is likely due to the calculation and rounding errors.

6.1.4 Evaporator Heat Balance

Similarly, the evaporator heat balance was taken across the refrigerant and air sides, following the same methodology as the condenser. The heat balance is shown in Figure 6.4.

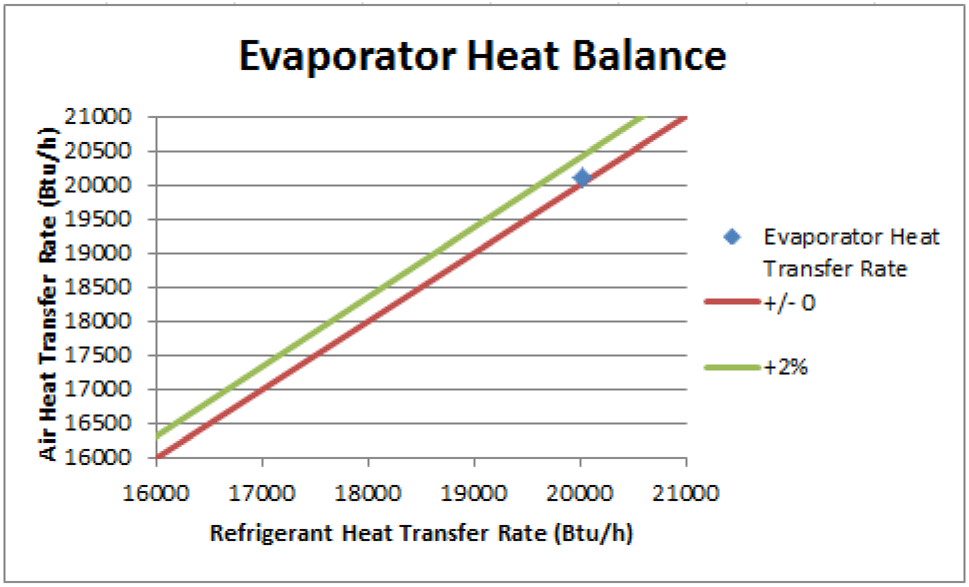


Figure 6.4 Evaporator Heat Balance

The evaporator heat balance shows an insignificant error, less than half-a-percent. This error is likely due to the calculation and rounding errors.

6.1.5 Heat Pump Power Distribution

Finally, the heat pump power distribution was considered; this is shown in Figure 6.5.

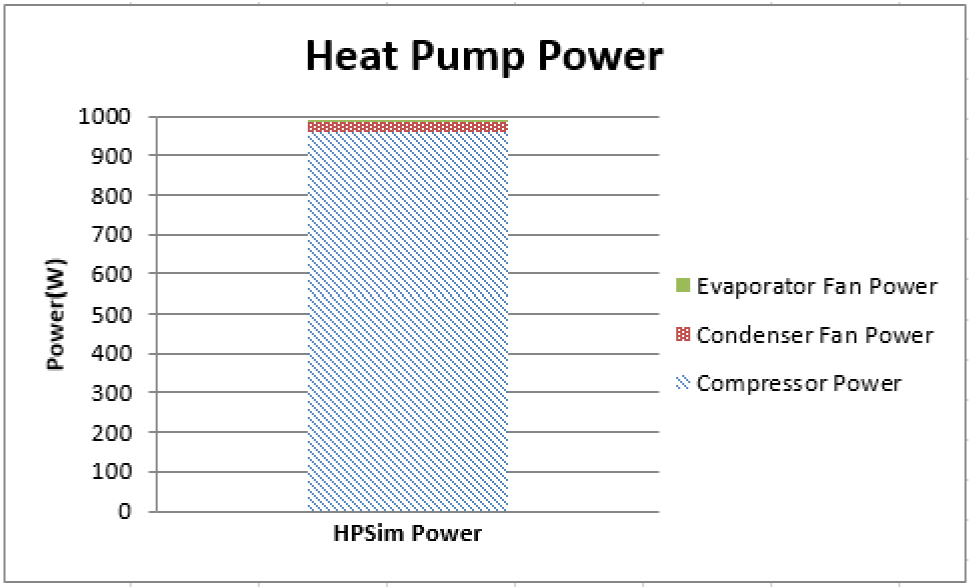


Figure 6.5 Heat Pump Power Distribution

The most significant portion of power is distributed to the compressor. In comparison, the condenser and evaporator fan powers are insignificant. The low power consumption of the fans is due to the power calculation being dependent on the air pressure drop across the fan; as the pressure drop is low, the power usage is also low. The pressure drop of the air is low as no system effect or duct length is accounted for in HPSim; there are no modeling capabilities for duct systems. This agrees with Gall's (2015) conclusion that low static pressure renders the evaporator fan power insignificant to the compressor power.

6.2 Sensitivity Analysis

After checking the heat balances, the next step to show proof of concept for the testbed is to determine the sensitivity of HPSim to adjusted variables. To this end, a number of exploratory cases were run. The sections following consider the sensitivity of HPSim to the compressor ratio and the condenser air flow rate. The air flow rate of the evaporator has a direct impact on the amount of heat transferred to the air, so it was not investigated in this section.

6.2.1 Compressor Ratio Sensitivity

The first of these looked at the impact of compressor ratio on evaporator leaving air temperature and evaporator capacity. Three different evaporator air flow rates were investigated for compressor ratio variation: one of 800 CFM, one of 1000 CFM, and one of 1200 CFM. The evaporator inlet conditions were held constant at 26.1 °C/79 °F drybulb and 8.5 °C/47 °F wetbulb. The compressor ratio ranged from 1 to its lower feasible bound of 0.15; below 0.15, the heat pump cycle was unstable and could not converge. The condenser air flow rate was held constant at 3500 CFM; the condenser air inlet conditions were constant at 21°C/69.8 °F drybulb and 17.7 °C/63.9 °F wetbulb. These conditions result in a dry coil. The first item checked is the heat balance across the evaporator, as seen in Figure 6.6.

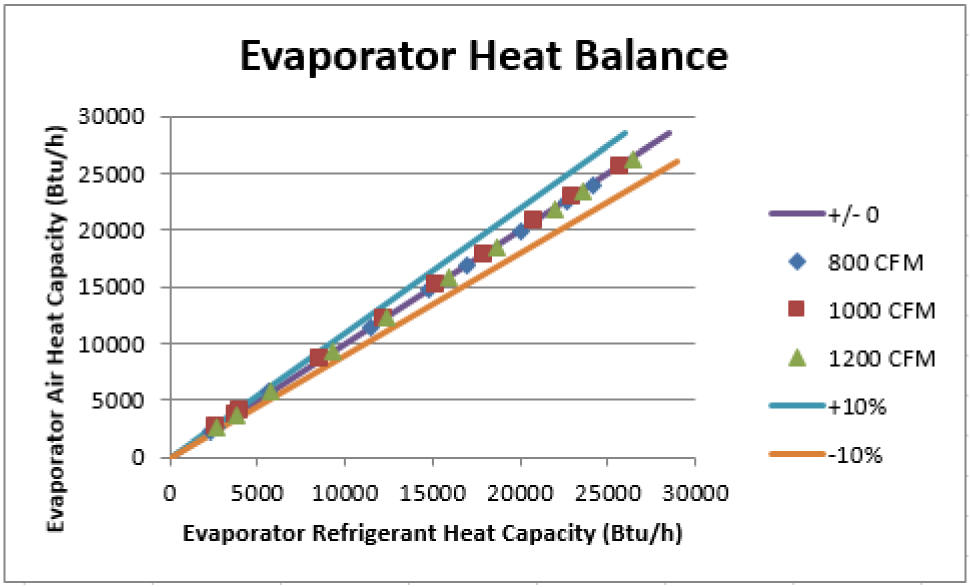


Figure 6.6 Evaporator Heat Balance

The evaporator heat balance follows a line with a slope close to 1-to-1. For each point, there is a difference of about 100 to 200 Btu/h between the air and refrigerant heat capacities, but this is insignificant (less than 2% error) for the compressor ratios from 1 to 0.4. When the compressor ratio reaches 0.3 and lower, the capacities have been greatly reduced and therefore a 100 or 200 Btu/h difference has a much larger impact on the error; even still, the error is always less than 9%. A variety of reasons for these errors is given in Section 6.1.1. The impact of the compressor ratio on evaporator gross capacity ($\dot{Q}_{evap}$) is seen in Figure 6.7.

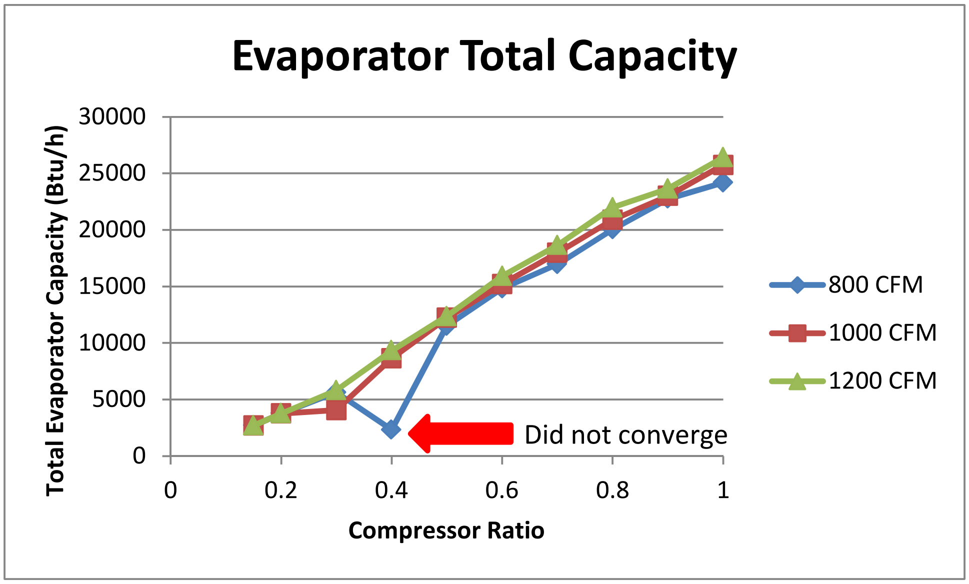


Figure 6.7 Evaporator Total Capacity at a Range of Compressor Ratios

Knowing that an increasing compressor ratio increases the refrigerant mass flow, the refrigerant heat balance,

$$\dot{Q}_{evap} = (\dot{m}_{ref} * \Delta h)_{evap} \qquad (6.2)$$

shows that as $\dot{m}_{ref}$ increases, $\dot{Q}_{evap}$ does also. This is shown clearly in Figure 6.8 except for one instance: the 800 CFM case with a compressor ratio of 0.4. The reason for this anomaly is the evaporator outlet does not converge on the refrigerant superheat; instead a two-phase refrigerant outlet was reported, reducing the leaving enthalpy drastically – and thus the refrigerant capacity. A reduced degree of superheat or even two-phase refrigerant flow leaving the evaporator greatly reduces the evaporator capacity. Other possible errors were discussed in Section 6.1.1.

Note that there is no significant difference between capacities for different evaporator air flow rates. This is due to the location of the evaporator refrigerant process on the pressure-enthalpy diagram. The refrigerant process primarily takes place in the saturated region. As the evaporator air flow rate changes, the evaporator pressure adjusts up and down on the pressure-enthalpy diagram in response. However, while the changing evaporator pressure also adjusts the enthalpy, the difference in enthalpy is small at the region of the pressure-enthalpy curve in question. This results in only a small change in the total refrigerant capacity over the evaporator.

Figure 6.8 shows the evaporator leaving air temperature for the three airflow cases.

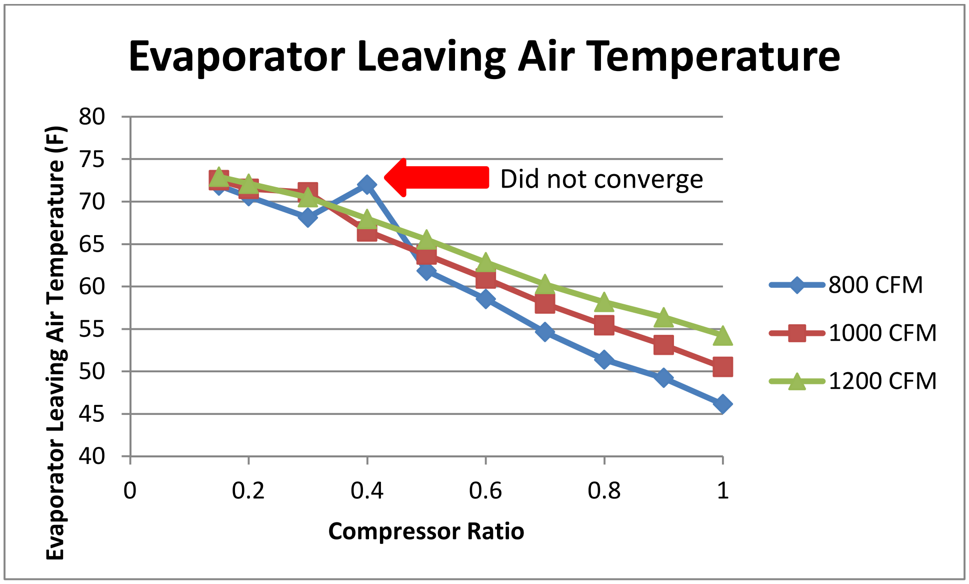


Figure 6.8 Evaporator Leaving Air Temperature at a Range of Compressor Ratios

Overall, the evaporator leaving air temperature is very sensitive to the compressor ratio with a maximum range of almost 30 °F/16.7 °C. Notice there is an outlying temperature point for a compressor ratio of 0.4 and an evaporator air flow rate of 800 CFM, the same outlying point as in Figure 6.7; this is due to nonconvergence in the calculation.

As discussed in Section 5.1.3, $\mathrm{T}_{air_{evap\ out}}$ is negatively proportional to the compressor ratio; as the compressor ratio increases, $\mathrm{T}_{air_{evap\ out}}$ decreases and vice-versa. This is clearly shown here as $\mathrm{T}_{air_{evap\ out}}$ becomes cooler as the compressor ratio increases. Although the trends are the same for all three air flow rates, the lower $\dot{m}_{air_{evap}}$ produce a lower $\mathrm{T}_{air_{evap\ out}}$ than the higher flow rates. The reason for this can be derived from the dry coil heat balance shown in Section 5.1.3, replicated below:

$$\mathrm{T}_{air_{evap\ out}} = \mathrm{T}_{air_{evap\ in}} - \frac{(\dot{m}_{ref} * \Delta h)_{evap}}{cp * \dot{m}_{air_{evap}}} \qquad (6.3)$$

As $\dot{m}_{air_{evap}}$ decreases, the $\frac{(\dot{m}_{ref} * \Delta h)_{evap}}{cp * \dot{m}_{air_{evap}}}$ term increases, resulting in a larger reduction in temperature for $\mathrm{T}_{air_{evap\ out}}$. In contrast, as $\dot{m}_{air_{evap}}$ increases, the $\frac{(\dot{m}_{ref} * \Delta h)_{evap}}{cp * \dot{m}_{air_{evap}}}$ term decreases, and $\mathrm{T}_{air_{evap\ out}}$ correspondingly increases in temperature.

Even still, the compressor ratio has a far greater impact on the temperature than the evaporator air flow rate: compare the modulating compressor ratio's temperature range (at a constant air rate) of almost 30 °F/16.7 °C with the constant compressor ratio's maximum temperature range (at a modulating air rate) of about 10 °F/5.6 °C. As such, the compressor ratio is the primary variable for GenOpt to adjust. The evaporator air flow rate can be secondarily adjusted, if necessary, by the E+/GenOpt Data Exchange Manager; the necessary condition for adjustment is compressor modulation resulting in unfeasible refrigerant cycles.

6.2.2 Condenser Air Flow Sensitivity

The condenser air flow rate ranged from 1000 to 5000 CFM; the condenser air inlet conditions were constant at 69.8 °F/21°C drybulb and 63.9 °F/17.7 °C wetbulb. Three different evaporator air flow rates were investigated for condenser air flow impact: 800 CFM, 1000 CFM, and 1200 CFM. The evaporator inlet conditions were held constant at 79 °F/26.1 °C/drybulb and 47 °F/8.5 °C wetbulb. The compressor ratio was held constant at 0.8. These conditions result in a dry coil. Because the focus of the testbed remains on the evaporator side of the heat pump cycle (due to this being the connection point between HPSim and EnergyPlus), the following discussion remains concerned with the evaporator performance at varying condenser air flow rates. Figure 6.9 considers at the evaporator heat balance for a variety of condenser air flow rates.

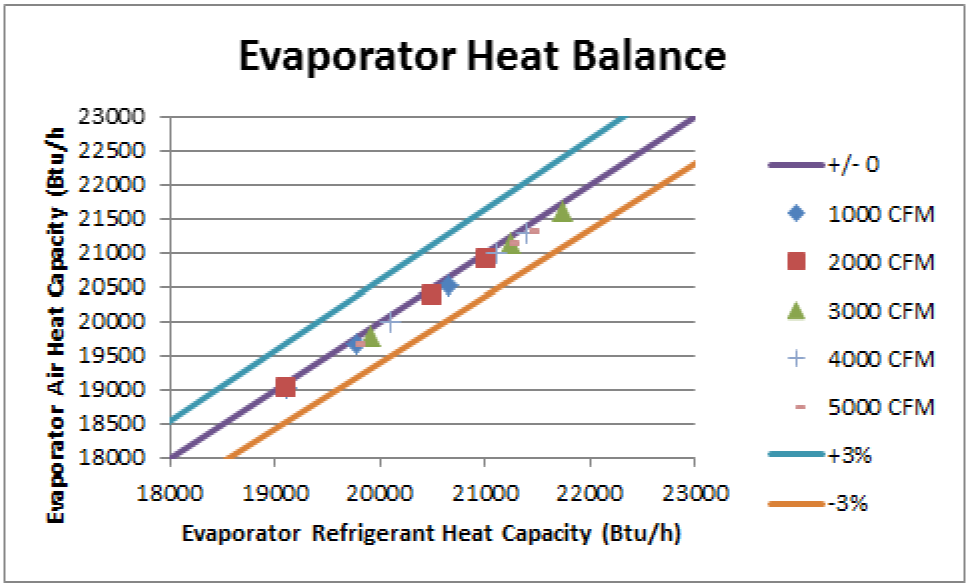


Figure 6.9 Evaporator Heat Balance at a Range of Condenser Air Flow Rates

Overall, the evaporator capacity balance is tight; all cases are within +/- 3% difference and there is no significant disagreement between the air and refrigerant heat transfer rates.

The next comparison is of the evaporator total capacity dependent on the evaporator and condenser air flow rates; this is shown in Figure 6.10. For clarity, the three lines are points at evaporator air flow rates, while the x-axis is the condenser air flow rate; the y-axis is the evaporator total capacity.

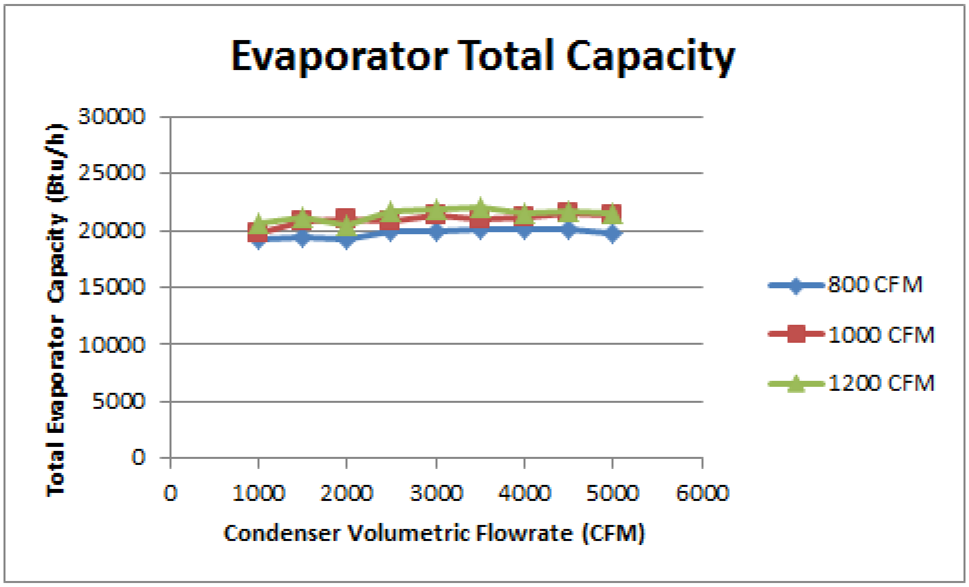


Figure 6.10 Evaporator Total Capacity at a Range of Condenser Air Flow Rates

All points above are at a constant compressor ratio of 0.8. Overall, the evaporator total capacities remain fairly constant at ranging condenser air flow rates.

Finally, the evaporator leaving air temperature was considered for a range of condenser air flow rates, as shown in Figure 6.11.

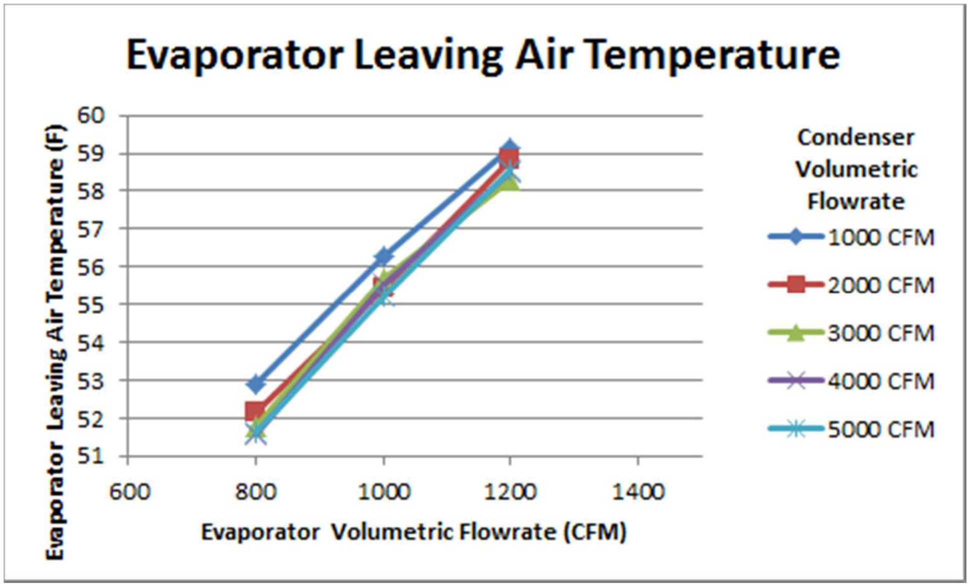


Figure 6.11 Evaporator Leaving Air Temperature at a Range of Condenser Air Flow Rates

Overall, the evaporator leaving air temperatures are only moderately sensitive to the condenser air flow rate; the largest variation is about 1.5 °F/0.8 °C at the lower evaporator air flow rate. This is reasonable, since the impact of the condenser air flow rate increases as the evaporator air flow rate decreases. With the evaporator air flow rate at 1200 CFM, the condenser air flow rate varies the evaporator leaving air temperature by only 1 °F/0.6 °C. At 1000 CFM, the evaporator leaving air temperature varies by about 1.5 °F/0.8 °C, and at 800 CFM, the temperature varies by about 2 °F/1.1 °C. These results correspond with what Gall (2015) discovered, especially as discussed in Section 5.1.2. In addition, optimizing the condenser fan required the addition of a third optimization variable with little payback. Therefore, the condenser fan was not optimized for the testbed. However, the condenser air flow rate can be adjusted by the EnergyPlus/GenOpt Data Exchange Manager if necessary.

In addition, it is worthwhile considering the effect of the condenser fan on the compressor power; this is seen in Figure 6.12. The three series shown on the chart are the evaporator air flow rates.

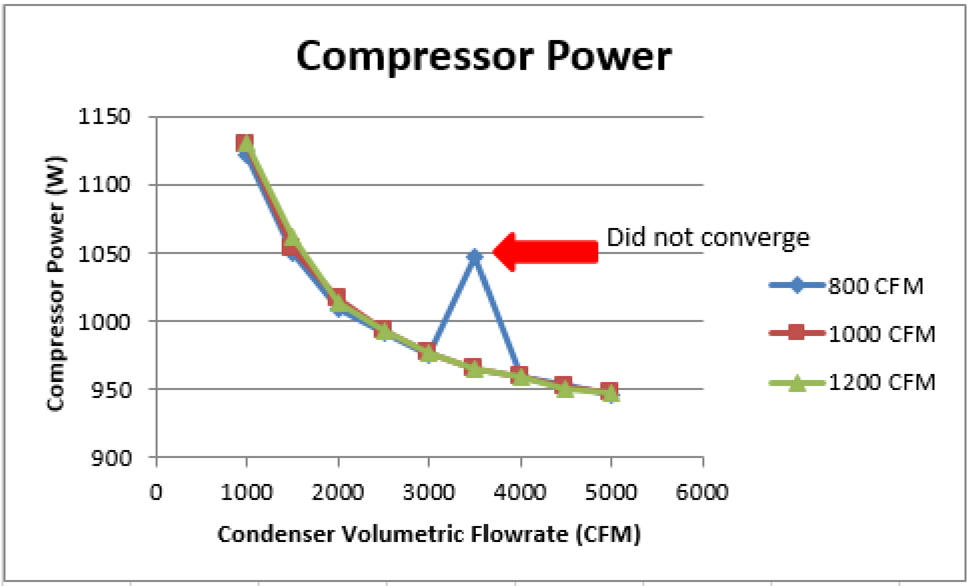


Figure 6.12 Compressor Power and Condenser Air Flow Rate

It can be generally seen that as the condenser air flow rate increases, the compressor power decreases. An overall decrease in compressor power of about 175 W is seen over the 4000 CFM increase in condenser air flow rate. The greatest rate in power decrease is at lower condenser air flow rates; as the condenser air flow rate increases between about 3000 and 5000 CFM, the rate of decline in power is more gradual. The one outlier is a point at an evaporator air flow rate of 800 CFM and a condenser air flow rate of 3500 CFM that did not converge; the compressor ratio was 0.8. The system power follows the same pattern as the compressor power, as seen in Figure 6.13.

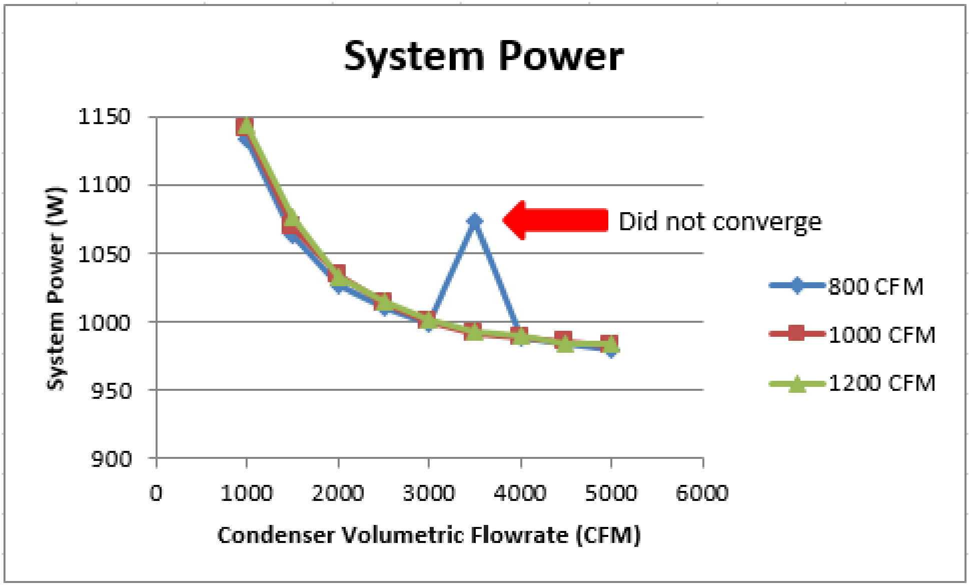


Figure 6.13 System Power and Condenser Air Flow Rate

The system power decreases with the same pattern, with a range of about 175 W over 4000 CFM of condenser air flow rate. The same nonconvergent point is an outlier for the system power, and also for the condenser refrigerant outlet pressure, as seen in Figure 6.14.

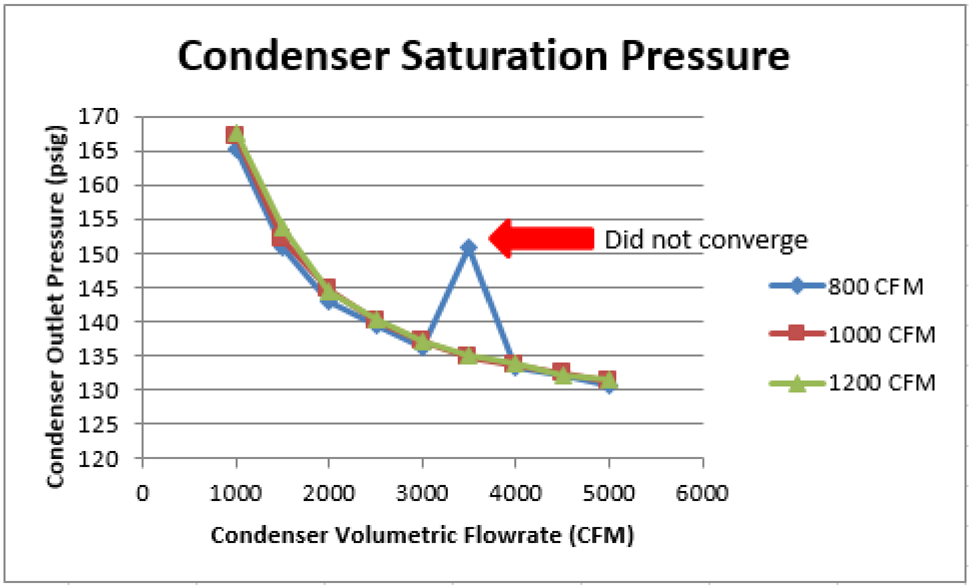


Figure 6.14 Condenser Refrigerant Outlet Pressure and Condenser Air Flow Rate

The condenser outlet pressure follows the exact same pattern as the compressor power: as the condenser air flow rate increases, the refrigerant condenser outlet pressure decreases. An overall decrease of about 35 psig is seen over a change in 4000 CFM in the condenser air flow rate. A final comparison of the compressor power and condenser outlet refrigerant pressure is seen in Figure 6.15.

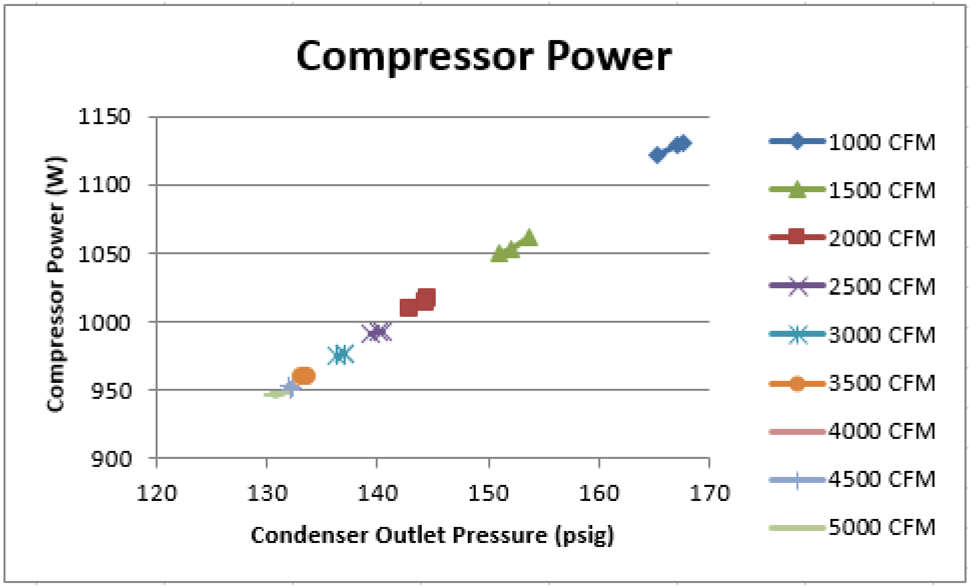


Figure 6.15 Compressor Power and Condenser Air Flow Rate

In Figure 6.15, the air flow rate series are for the condenser fan. As the condenser air flow rate increases, the compressor power and condenser outlet pressure decrease in conjunction.

6.3 Connectivity Tests

In order to demonstrate the testbed concept, connectivity of the composite programs had to be shown. To this end, the inputs and outputs of EnergyPlus and HPSim were investigated. As the two programs are at opposite ends of the testbed, if they each have the pertinent information, then the intermediate programs have the correct information as well. The information that EnergyPlus sent to HPSim was tracked, as was the information sent back from HPSim to EnergyPlus. The heat transfer rates HPSim reported and the heat transfer rates calculated from the air properties at the inlet and outlet of the coil, which EnergyPlus sent and received, were compared. These comparisons are below for the sensible capacity, latent capacity, and total capacity of the coil.

The point considered had a condenser air flow rate of 3500 CFM; the condenser air inlet conditions were 70.39 °F/21.3°C drybulb and 63.75 °F/17.6 °C wetbulb. The evaporator air flow rate was 990 CFM; the evaporator inlet conditions were 75.38 °F/24.1 °C drybulb and 65.37 °F/18.5 °C wetbulb. The compressor ratio was 1. These conditions result in a wet coil.

The sensible capacity was calculated with the air properties of EnergyPlus nodal points using

$$\dot{Q}_{sensible} = \dot{m}_{air} * cp * (T_{air_{in}} - T_{air_{out}}) \qquad (6.3)$$

where $\dot{m}_{air}$ (lbm/hr) and $T_{air_{out}}$ (°F) were HPSim outputs sent to EnergyPlus, $T_{air_{in}}$(°F) was an input from EnergyPlus to HPSim, and cp (taken as 0.2403 Btu/lbm) was the standard specific heat of air. This value is indicated in Figure 6.16 on the horizontal axis while the vertical axis shows the rate of sensible heat transfer as calculated by HPSim.

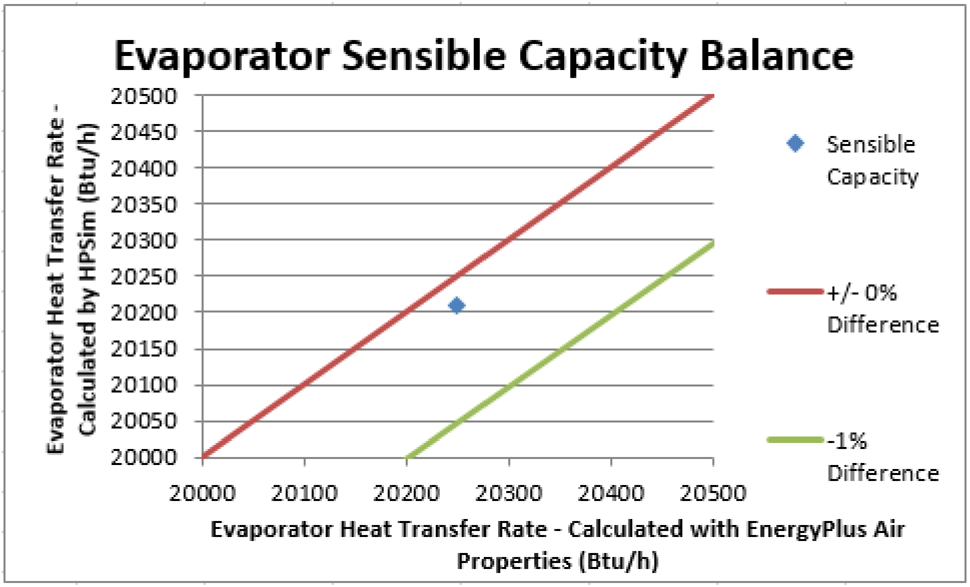


Figure 6.16 Evaporator Sensible Capacity Balance

Although the two heat transfer rates are not identical, the difference between them is 40 Btu/h – a difference of 0.2%. This difference is insignificant, and is easily explained by the considerations seen in Section 6.1.1.

The latent capacity was calculated using

$$\dot{Q}_{latent} = \dot{m}_{air} * \left(\omega_{air_{in}} - \omega_{air_{out}}\right) * h_{fg} \qquad (6.4)$$

where $\dot{Q}_{latent}$ is rate of latent heat transfer (Btu/h), $\dot{m}_{air}$ is the mass flow rate of the air (lbm/h), $\omega_{air_{in}}$is the humidity ratio of the air entering the coil (lb/lb), $\omega_{air_{out}}$ is the humidity ratio of the air leaving the coil (lb/lb), and h_{fg} is the rate of evaporation (Btu/lbm). This value and the latent capacity determined by HPSim agree less well than the sensible capacities, but with less than a 4% error. The latent capacity balance is seen in Figure 6.17.

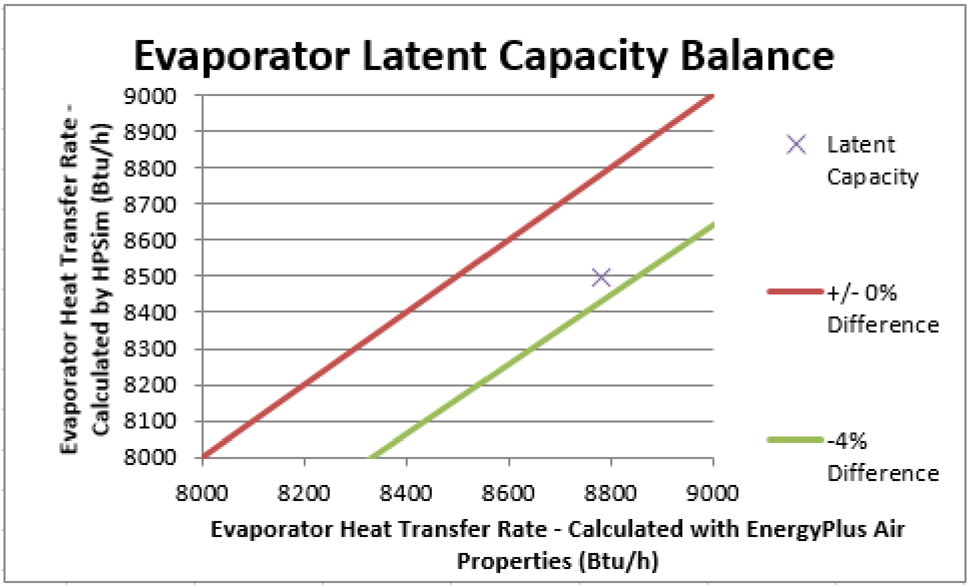


Figure 6.17 Evaporator Latent Capacity Balance

The error is likely due to minor differences in air property routines, convergence errors, and rounding. However, it is important to note that the latent capacities only agree well if both programs use the same barometric pressure, which is dependent on the elevation above sea level. Different barometric pressures can have a large impact on the air properties calculated by the psychrometric routines. For example, a difference of 1010 ft/308 m resulted in a difference in humidity ratio (lb/lb) on the magnitude of ten-thousandths (about five ten-thousandths). This difference, although small, resulted in a latent heat transfer difference of about 2,000 Btu/h/586 W (out of 8,500 Btu/h/2491 W). HPSim currently calculates air properties at sea level, while EnergyPlus calculates at the height listed in the weather file; for this testing, the weather file height was set to sea level.

Overall, the total capacity calculated by HPSim compares favorably with the total capacity calculated using EnergyPlus' values and

$$\dot{Q}_{total} = \dot{Q}_{sensible} + \dot{Q}_{latent} \qquad (6.5)$$

There is less than a 2% error between the two total capacities, as shown in Figure 6.18.

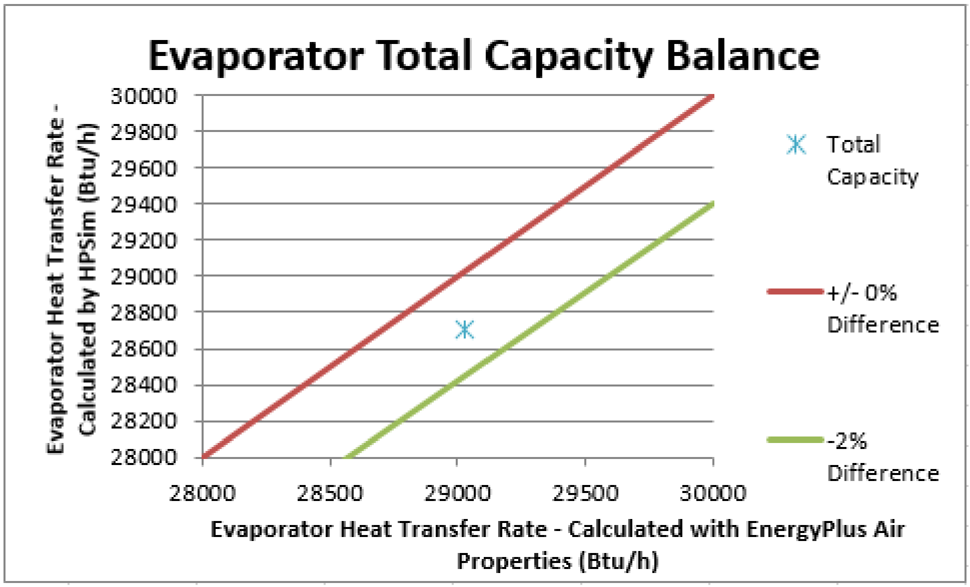


Figure 6.18 Evaporator Total Capacity Balance

This error is a combination of sensible and latent errors and therefore due to the same concerns.

Overall, the potentials for error include varying barometric pressures in the psychrometric routines, energy leaving with condensate for wet coils, convergence issues, and rounding/conversion errors. Each of these should be restricted to potential small contributions to error (with the exception of some convergence issues), but together can accumulate. Regardless, this test case shows that HPSim and EnergyPlus are properly connected and communicating.

6.4 Simulation Testbed Case Study

A two-zone model conditioned by a single heat pump was created in EnergyPlus to verify correct operation of the simulation testbed. The penalty function schema discussed in Section 5.2 was implemented in the testbed.

The run period was the full day of September 7; no sizing period design day was run. A weather file for Stillwater, Oklahoma from the EnergyPlus website was used to supply the weather data. The outdoor air temperatures and relative humidities for the day is shown in Figure 6.19.

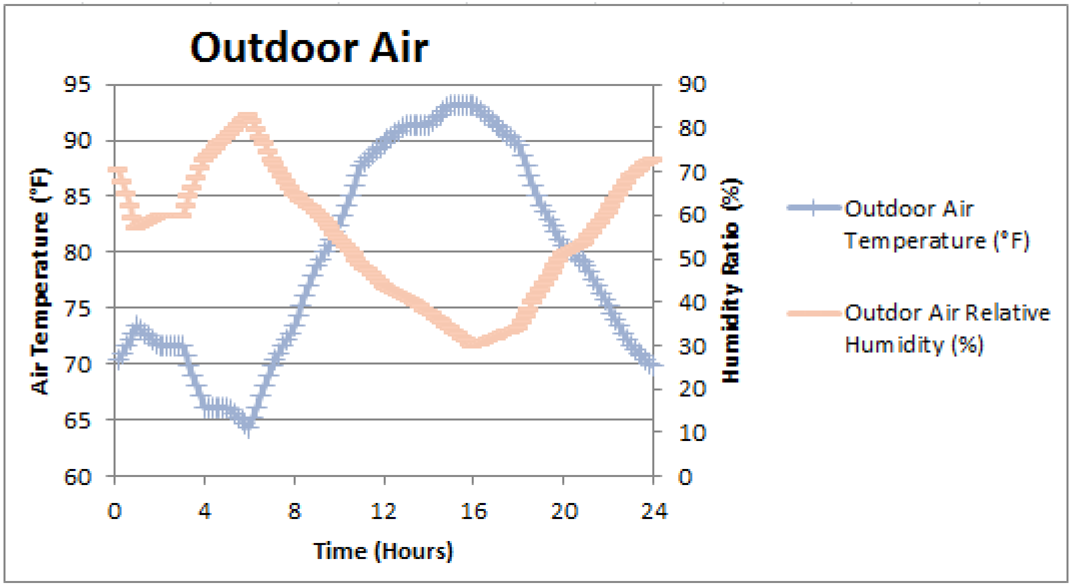


Figure 6.19 Outdoor Air Conditions for September 7 in Stillwater, Oklahoma

The exterior building wall structure was composed of 1”/2.54 cm stucco, 4”/10.16 cm brick, 1”/2.54 cm insulation, and 4”/10.16 cm gypsum board. The interior wall was composed of 3/4”/1.91 cm gypsum board, 8”/20.32 cm clay tile, and 3/4 in/1.91 cm gypsum board. The floor was constructed of 8”/20.32 cm HW concrete; the roof is composed of 1/2”/1.27 cm stone, 3/8”/0.95 cm felt and membrane, 1”/2.54 cm dense insulation and 2”/5.08 cm HW concrete. The exterior building walls were sun and wind exposed. The building was 9.84 ft/3 m tall, 31.5 ft/9.6 m long, and 31.5 ft/9.6 m wide; the building split in half along the length to form the two rooms represented by each zone.

There are two zones: an east zone and a west zone, each consisting of a single room. The zones had no internal load from 12 AM until 12 PM, and then a moderate internal load (8,000 Btu/h or 25.4 W/m^2) from 12 PM until 12 AM. For reference, an average office equipment heat gain is around 10.8 W/m^2.

The air loop contains a fan and a cooling coil. Both the EnergyPlus fan and coil models are used only in Warmup; otherwise, HPSim models both. The coil outlet node is its sensor node, its schedule is the cooling coil availability schedule, and the cooling coil object type is a single speed

DX coil. The coil has a length of 3.9 in/0.1 m and a hydraulic diameter of 3.28 ft/1 m. It has a gross rated total capacity of 36,048 Btu/h/10,565 W, a sensible heat ratio of 0.75, a gross rated cooling COP of 3, and a rated air flow of 1271 CFM/0.6 m^3/s. The fan is variable volume, with an always-on availability per the fan and coil availability schedule. It has a 70% efficiency, a 0.09 psi/600 Pa pressure rise, a maximum flow rate of 2755 CFM/1.3 m^3/s and a minimum of 424 CFM/0.2 m^3/s, and a 90% motor efficiency with the motor completely in the air stream. The fan inlet node is the Air Loop Inlet Node and the air outlet node is the Cooling Coil Air Inlet Node. The cooling coil availability schedule is 1 (always on), the cooling setpoint is 73.4 °F/23 °C, the heating setpoint is 59 °F/15 °C, the zone control type schedule is 4 (dual temperature setpoint), and the reheat temperature schedule is 140 °F/60 °C.

Each zone has a thermostat, but they both have a dual setpoint thermostat with the same cooling and heating setpoints per above; note that the heating setpoint is never used. The supply air temperature manager is controlled by the Seasonal Reset Supply Air Temp schedule on the Air Loop Outlet Node. There are also air distribution units, with their outlets being the respective zone air outlet nodes; these contain the reheat coils and dampers, and are listed under zone equipment connections. The heat pump refrigerant for HPSim was R-22.

6.4.1 Simulation Constraints

For GenOpt, a mesh size divider of 2 was used with an initial mesh size exponent of 0 and an increment of 1; the reduction step number was 1. Based on test cases, HPSim can generally operate with a compressor ratio between about 0.15 and 1. To reduce the number of points checked in the optimization and thus the runtime, GenOpt modulated the compressor ratio between 0.8 and 1. The initial point of optimization for the compressor flow rate was 0.8 and the step size was 0.1.

Cases were run for a single 24-hour day and data was reported in 10-minute increments. The simulation controls were discussed in earlier chapters, particularly Section 5.1. To demonstrate the responsiveness of the controls and penalty function to EnergyPlus, internal loads and penalty functions varied in the following cases.

For all cases considered, the y_3 penalty function component was 0. In addition, y_1 varied depending on the timestep inputs, but its maximum value was 15 W. This is insignificant compared to the y_4 penalty function component, as discussed in Section 6.4.3. This resulted in y_4 – the comfort-energy tradeoff – controlling the penalty function.

6.4.2 Simulation Baseline

The first case discussed is a baseline case with no penalty function: y_1, y_3, and y_4 were set to 0. GenOpt thus minimized true heat pump power and no controls or constraints were added to the power. The cooling setpoint for this case was 73.4 °F/23 °C with a 1.8 °F/1 °C deadband on the drybulb return air temperature. This case had no internal load from 12 AM until 12 PM, and then a moderate internal load (8,000 Btu/h or 25.4 W/m^2) from 12 PM until 12 AM. For reference, an average office equipment heat gain is around 10.8 W/m^2.

The evaporator entering air – or return air – temperatures are shown below in Figure 6.20.

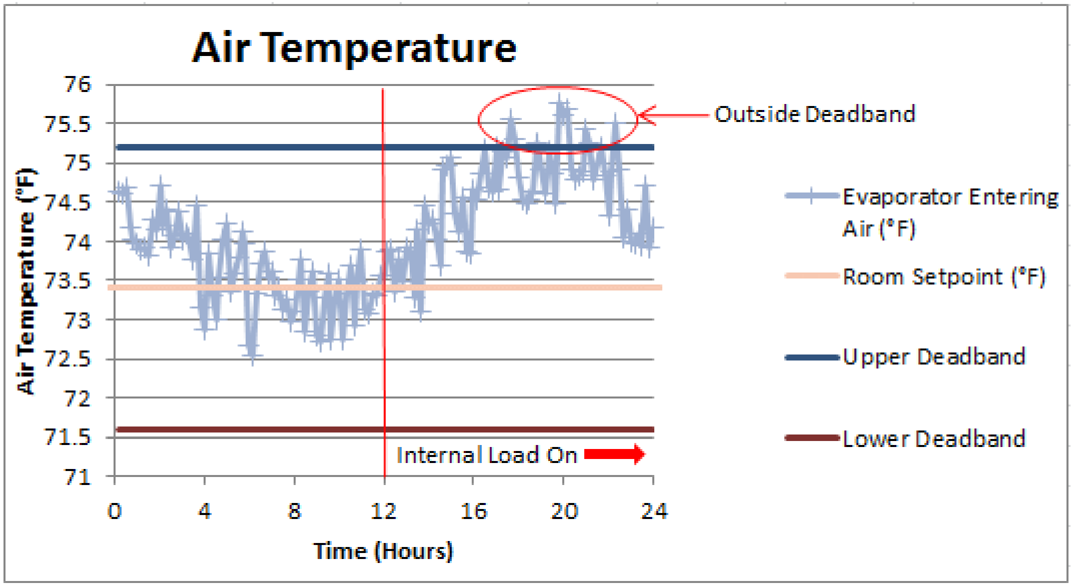


Figure 6.20 Evaporator Entering Air Temperature for September 7

The air temperature begins the day around 74-74.5 °F/23.3-23.6 °C and then decreases throughout the morning. The air temperature remains close to the setpoint, within the +/- 1.8°F/1 °C setpoint deadband in the morning. The internal load begins at 12 PM and the air temperature begins to rise correspondingly. Overall, the air temperature remains inside the setpoint deadband, except between 6 and 10 PM, when it exceeds the upper deadband nine times. The highest temperature occurs at 8 PM when the air returns at 75.8 °F/24.3 °C, or 0.6 °F/0.33 °C above the deadband. The peak cooling rate also occurs at 8 PM, as shown in Figure 6.21.

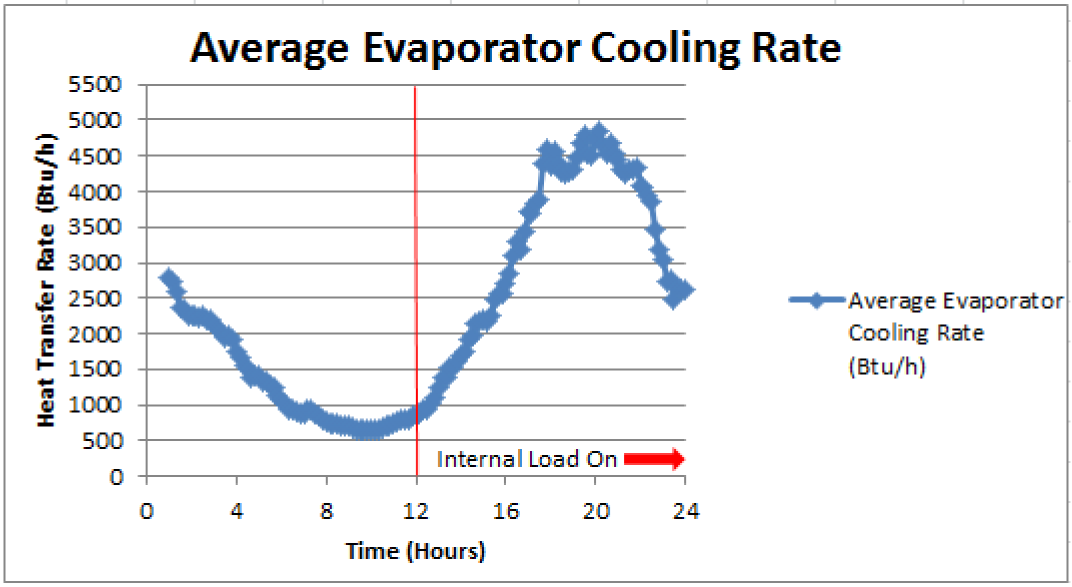


Figure 6.21 Evaporator Cooling Rate for September 7

The evaporator cooling rate is averaged over the hour. The evaporator cooling rate starts off around 2,800 Btu/h but reduces to about 700 Btu/h by 8 AM and minimizes at 625 Btu/h at 10 AM. The cooling rate remains low until 12 PM. Once the internal load begins at 12 PM, the cooling rate increases, peaking at 4,800 Btu/h around 8 PM. The highest rates of cooling occur between 7 and 9 PM; this corresponds with one of the out-of-deadband temperatures shown in Figure 6.20. The rate of cooling fluctuates but remains high until around 10 PM, when it starts to decline. The heat pump power shows a similar trend; Figure 6.22 shows the hourly average.

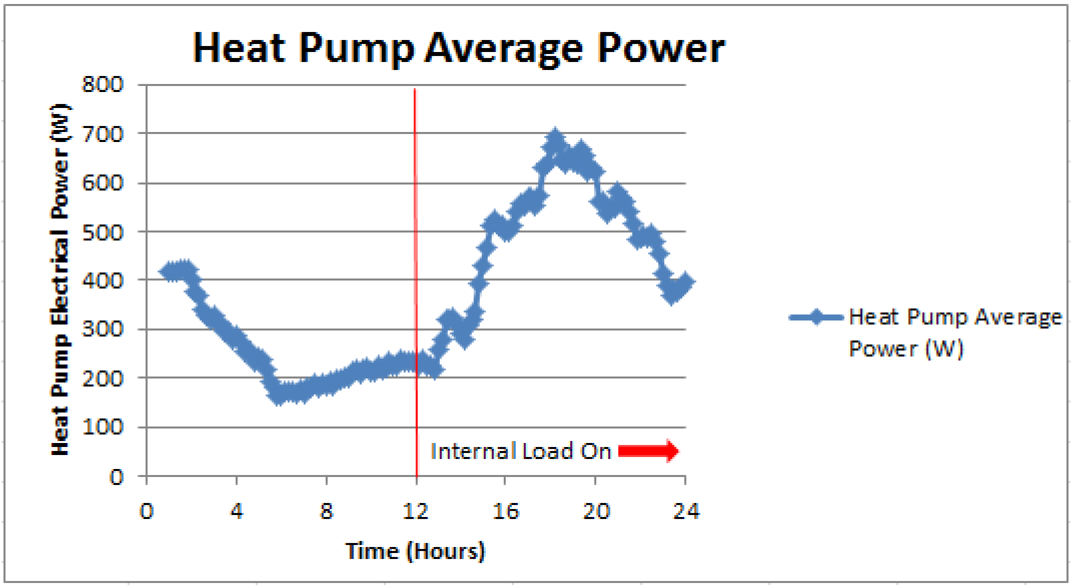


Figure 6.22 Heat Pump Average Power for September 7

The heat pump power starts off at 420 W and then decreases throughout the morning, reaching a minimum around 6 AM. After 6 AM, the power slowly increases until the internal load is introduced at 12 PM; after 12 PM, power rapidly increases. The power peaks at 690 W near 6 PM, earlier than the peak cooling rate and air temperatures. The heat pump power fluctuates throughout the evening, but remains high. It begins to significantly around 9 PM.

The compressor ratio is shown in Figure 6.23.

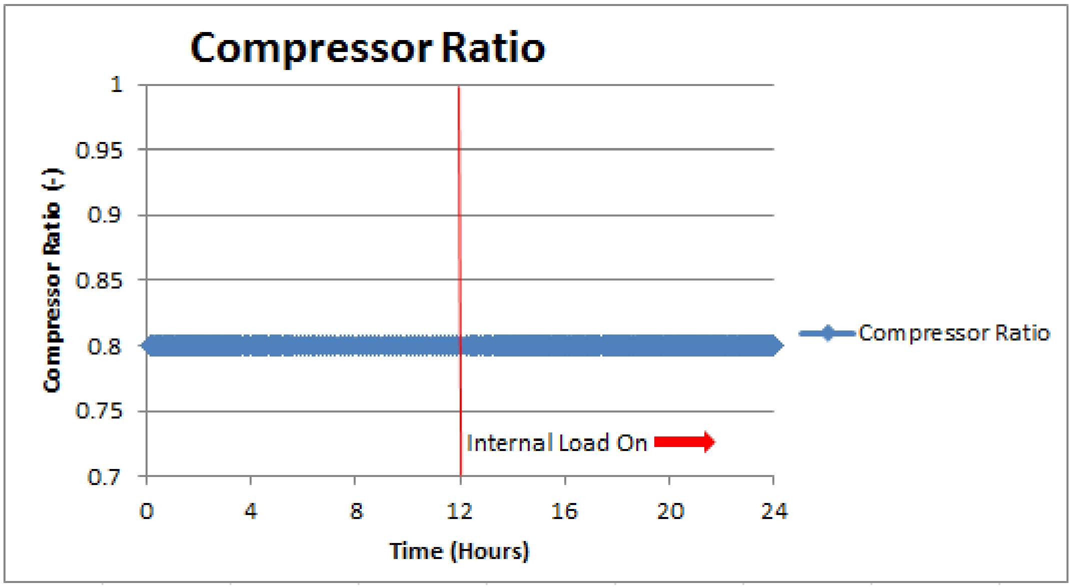


Figure 6.23 Compressor Ratio for September 7

The compressor ratio is constant for the minimum of 0.8, as anticipated. Without a penalty function, a compressor ratio of 0.8 will produce the lowest power in every instance.

6.4.3 Penalty Function Simulation Results

The following cases contain the penalty function that was neglected in the baseline. Per Section 5.2.3, y_4 is adjusted per the compressor ratio range. For a compressor ratio range of 0.8 to 1.0, the range in power is about 200 W and y_4 thus becomes

$$y_4 = 200 \frac{W}{{}^{\circ}\mathrm{F}^2} * (T_{setpoint} - T_{RA})^2 \qquad (6.6)$$

The return air temperature – or evaporator entering air temperature – over the full day is seen in Figure 6.24.

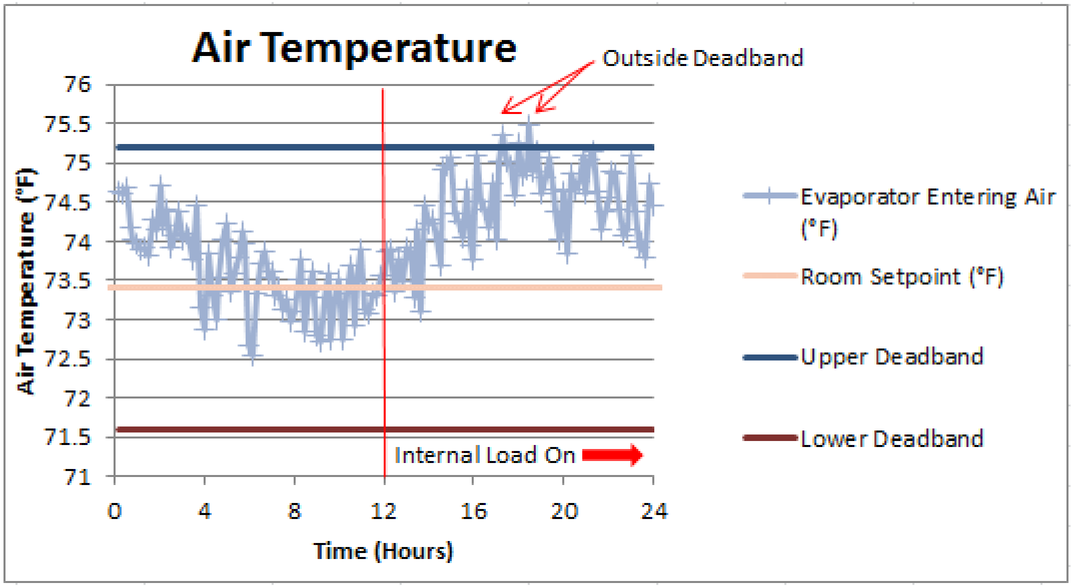


Figure 6.24 Evaporator Entering Air Temperature for September 7

Like in the baseline case, the air temperature begins the day around 74-74.5 °F/23.3-23.6 °C and gradually decreases throughout the morning. The air temperature remains close to the 73.4 °F/23 °C setpoint, within the +/- 1.8°F/1 °C deadband in the morning. When the internal load begins at noon, the temperature increases and is overall higher in the afternoon than in the morning. Regardless, the temperatures are kept within the setpoint deadband with two exceptions: a 75.4 °F/24.2 °C temperature around 6 PM and a 75.5°F/24.2 °C around 7 PM. Note that the envelope load in the morning hours is not sufficient to trigger the penalty function.

The average evaporator cooling rate for the day is seen in Figure 6.25.

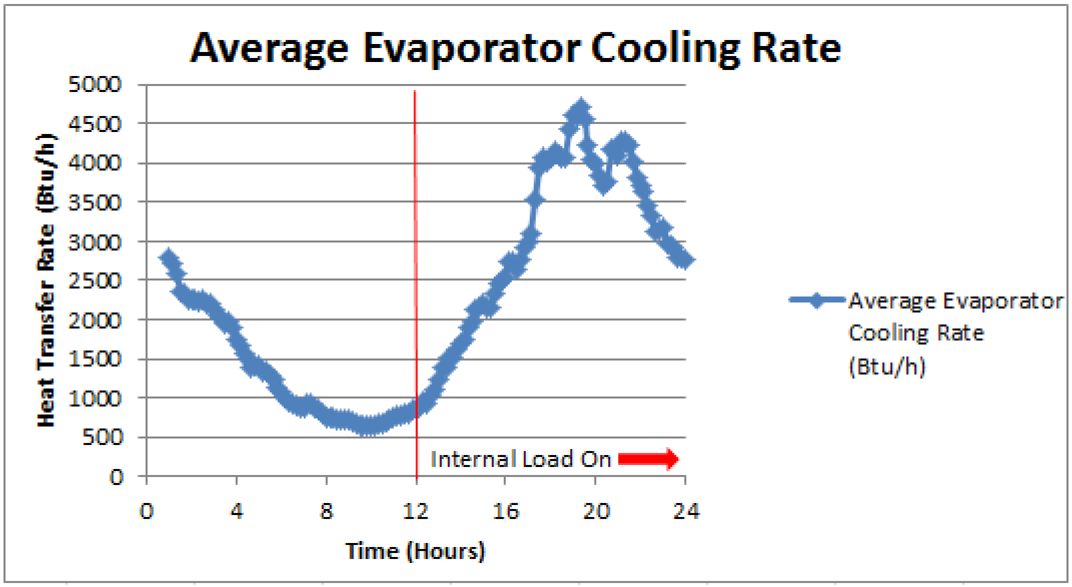


Figure 6.25 Evaporator Cooling Rate for September 7

The evaporator cooling rate starts off around 2,800 Btu/h but reduces to about 700 Btu/h by 8 AM and reaches a 625 W minimum around 10 AM. The cooling rate begins to increase around 10 AM but remains low until 12 PM. Once the internal load begins at 12 PM, the cooling rate increases, peaking around 7 PM; this corresponds with one of the out-of-deadband temperatures shown in Figure 6.21. The rate of cooling fluctuates but remains high until around 10 PM, when it starts to decline. The heat pump power shows a similar trend; Figure 6.26 shows the hourly average.

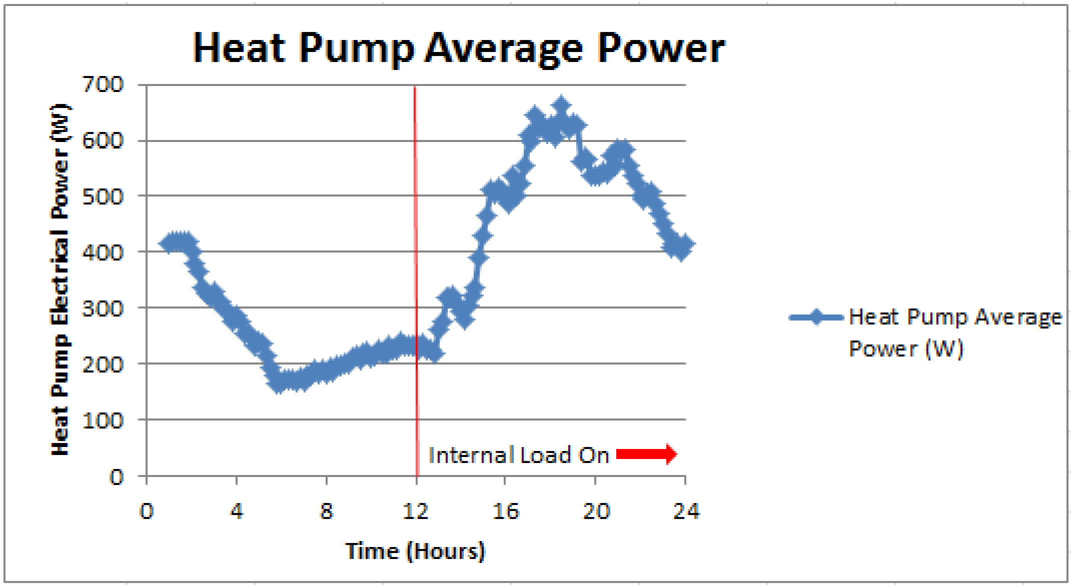


Figure 6.26 Heat Pump Average Power for September 7

As expected, the lowest power is seen at the lowest cooling rate, from about 4 AM to 8 AM. As the cooling rate and compressor ratios generally increase from 12 PM through the end of the day, so does the power. The power increases at a slightly faster rate than the cooling rate from 8 AM to 12 PM. At 12 PM the internal heat load is applied and from noon until 10 PM, the power generally increases; there are some fluctuations between 7 and 9 PM, which correspond with cooling rate fluctuations. At 10 PM, the power rate begins to decline in conjunction with the reduction in cooling rate. The daily power usage is 51.38 kW. The compressor ratio also reduces around 10 PM, as Figure 6.27 shows.

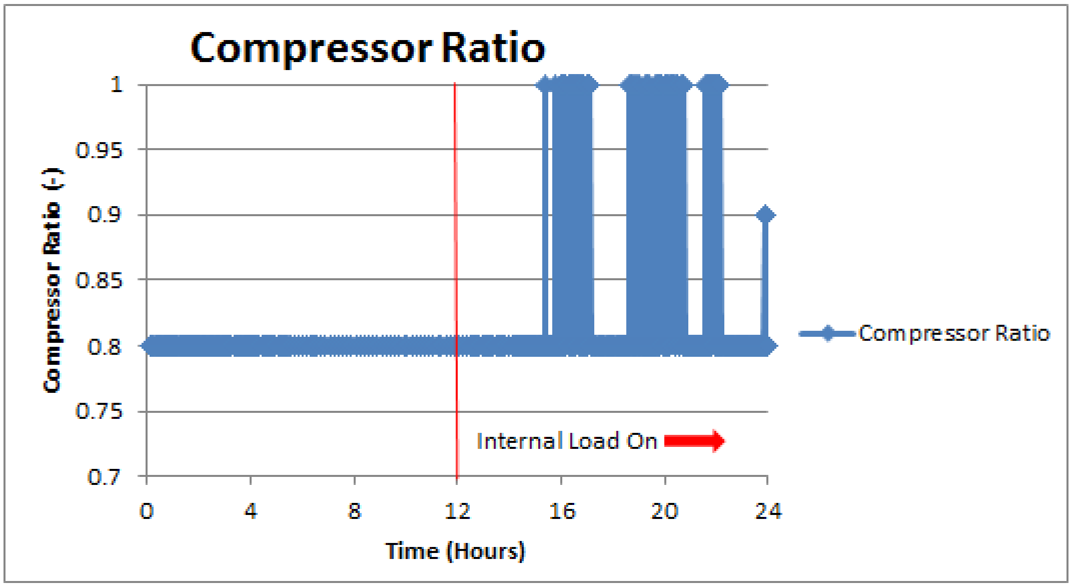


Figure 6.27 Compressor Ratio for September 7

The compressor ratio is constant at 0.8 throughout the morning and into the early afternoon. Since the minimum compressor ratio was set to 0.8, this means that the compressor is running at the minimum ratio for this entire time. After 12 PM, the internal heat load was applied to the zone, but there was no immediate increase in the compressor ratio. The compressor ratio increases to 1 at 4 PM and then fluctuates between the maximum and minimum compressor ratios for the remaining afternoon and early evening. There is only one point, at midnight, when the compressor ratio goes to a value between 0.8 and 1; the compressor ratio is set as 0.9. This shows that the penalty function is too tightly constrained for this case – at a ΔT of 1 °F/0.56 °C, y_4 results in a 200 W component. This is equivalent to the nominal change in heat pump power between compressor ratios of 0.8 and 1; the penalty function thus renders the power from a 0.8 compressor ratio as equivalent or slightly larger than the power from a compressor ratio of 1. As the ΔT in the afternoon is around 2 °F/1.11 °C in some places, this results in a y_4 component having a value of 400 W; this would have a similar result. The impact of the penalty function is seen further in Section 6.4.4.

6.4.4 Penalty Function Adjustments

Three additional values for the y_4 penalty function component were investigated. These are addressed in the following sections.

6.4.4.1 Case 1

The same behavior from Section 6.4.3 was also seen when y_4 was halved (y_4 = 100 W/°F^2) and quartered (y_4 = 50 W/°F^2). When y_4 was reduced by an order of magnitude – down to 20 W at a ΔT of 1 °F/1 °F/0.56 °C or 160 W at a ΔT of 2 °F/1.11 ° – it no longer dominated the optimization. Therefore, for the first case, y_4 had the form:

$$y_4 = 20\frac{W}{°F^2} * (T_{setpoint} - T_{RA})^2 \qquad (6.6)$$

The air temperature at this penalty function is seen in Figure 6.28.

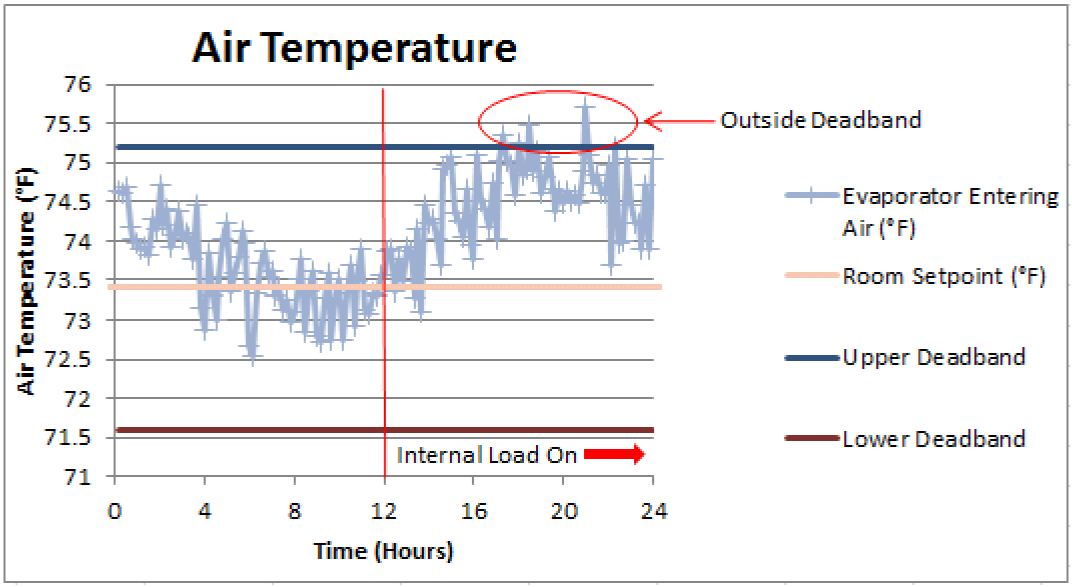


Figure 6.28 Evaporator Entering Air Temperature for September 7

As the morning envelope load is not sufficient to trigger the penalty function, the morning temperatures remain the same as the case shown in Section 6.4.3. Once the internal load is applied at noon, the temperature fluctuates in an upward direction. Air temperatures exceed the upper limit of the deadband at three instances – 6 PM, 6:30 PM, and 9 PM. The 9 PM temperature is the highest, but remains within 0.5 °F/0.28 °C of the upper deadband at 75.7 °F/24.3 °C. Therefore, the y_4 = 20 W/°F^2 results in an overall +/- 2.3°F/1.28 °C deadband. The impact of this y_4 on the heat pump power is seen in Figure 6.29.

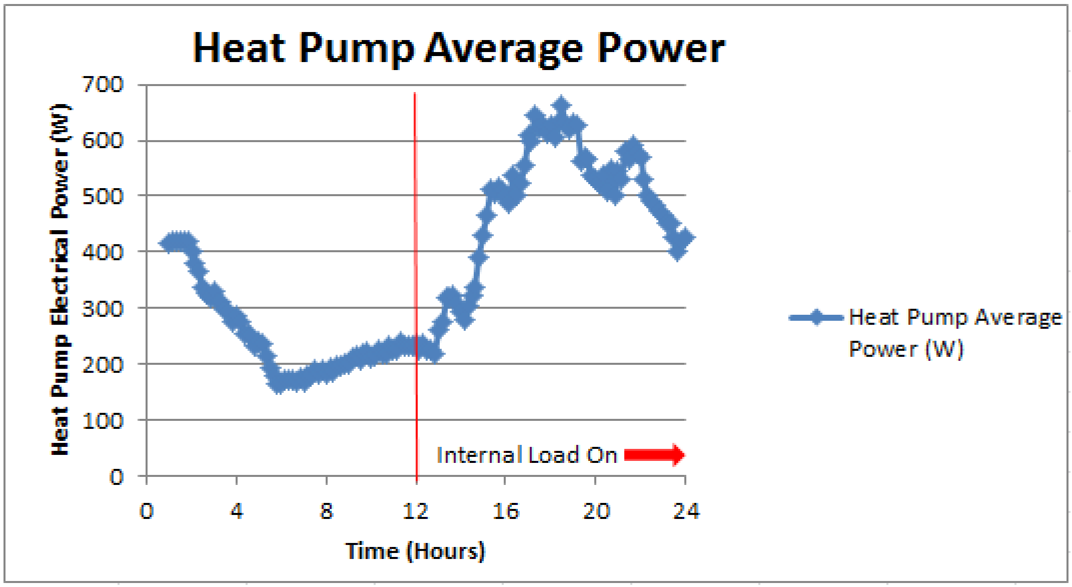


Figure 6.29 Heat Pump Average Power for September 7

The heat pump power for the morning is the same as in Section 6.4.3 due to the penalty function not being triggered. As the internal load is added at 12 PM, the heat pump power increases; it reaches a maximum of 660 W at 6:30 PM. The maximum power corresponds with a high temperature shown in Figure 6.28. The temperature decreases slightly until 8 PM, when it begins to increase again, reaching a local maximum of 585 W at 9 PM. Between 9 and 9:30 PM, the power decreases slightly but remains fairly constant; the high power rates correspond with the highest temperature shown in Figure 6.28. After 9 PM, the power decreases again, reaching a minimum of 400 W around midnight. The daily power usage is 51.24 kW, a slight reduction from the baseline case.

Finally, the compressor ratio for this case is considered in Figure 6.30.

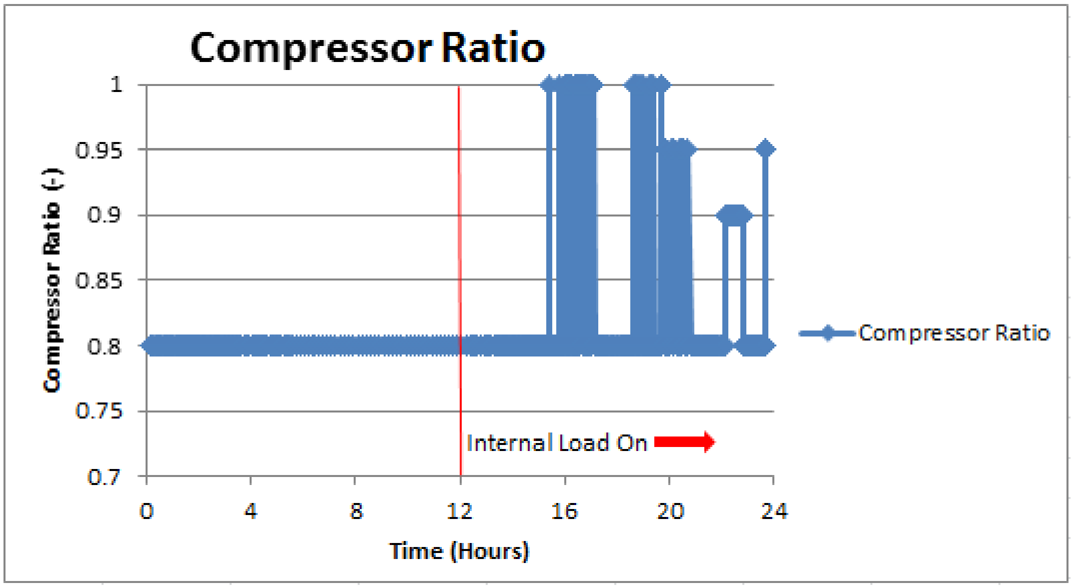


Figure 6.30 Compressor Ratio for September 7

The introduction of some intermediate compressor ratio values (0.9 and 0.95) indicate that the y_4 penalty function component is no longer completely dominating the optimization. However, the large area of compressor ratios fluctuating between the minimum and maximum, as seen between 3 and 7 PM, indicate that the y_4 penalty function component may still be too large. To investigate this, y_4 was halved again.

6.4.4.2 Case 2

The new y_4 is thus

$$y_4 = 10\frac{W}{{}^{\circ}\mathrm{F}^2} * (T_{setpoint} - T_{RA})^2 \qquad (6.7)$$

The impact of the new y_4 is seen on the air temperatures in Figure 6.31.

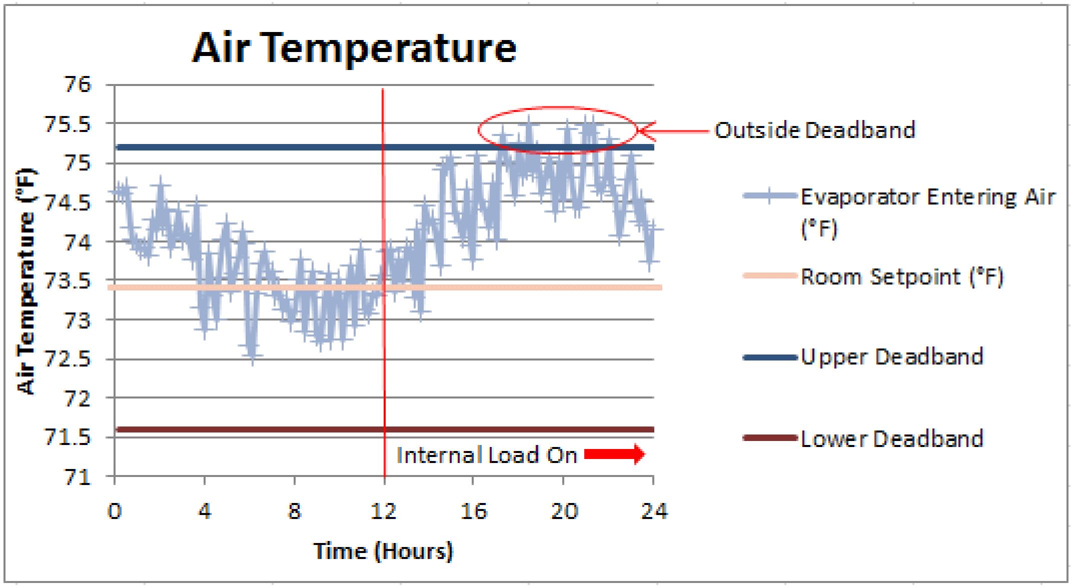


Figure 6.31 Evaporator Entering Air Temperature for September 7

As the weight of y_4 halved again, more temperatures exceed the upper deadband limit. There are eight temperatures between 6 to 10 PM that exceed the deadband; however, they remain within 0.5 °F/0.28 °C of the upper deadband at 75.7 °F/24.3 °C. Therefore, the y_4 = 10 W/°F^2 penalty function component also results in an overall +/- 2.3°F/1.28 °C deadband. This indicates that, if a +/- 2.3°F/1.28 °C deadband to the 73.4 °F/23 °C setpoint is acceptable, there is no disadvantage to the y_4 = 10 W/°F^2 penalty function component from a temperature perspective. Figure 6.32 shows a slight advantage from a power perspective.

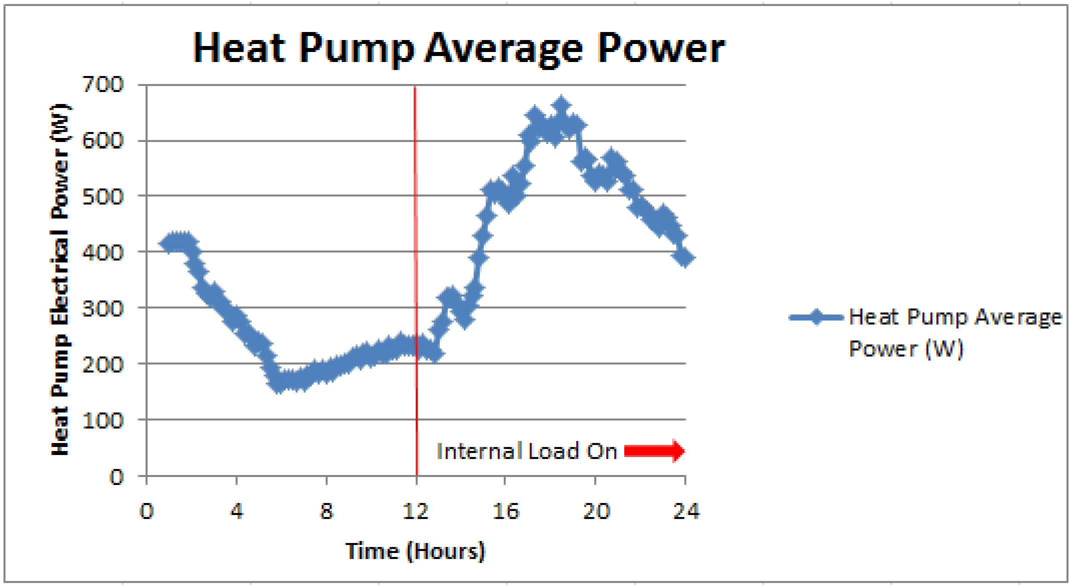


Figure 6.32 Heat Pump Average Power for September 7

The peak power is 660 W at 6:30 PM, the same as the y_4 = 20 W/°F^2 penalty function component case previously shown in Figure 6.29. However, a local maximum of 570 W occurs at 9:30 PM, a reduction of 15 W from the local maximum at that time seen in Figure 6.29. The power decreases between 9:30 PM and midnight, where it reaches a local minimum; at 12 AM, the power is at a value of 390 W, a slight reduction from the minimum seen for the previous case in Figure 6.29. The daily power usage is 50.91 kW. Although slight, an overall decrease in power is seen between the two cases.

The compressor ratios for this case are shown in Figure 6.33.

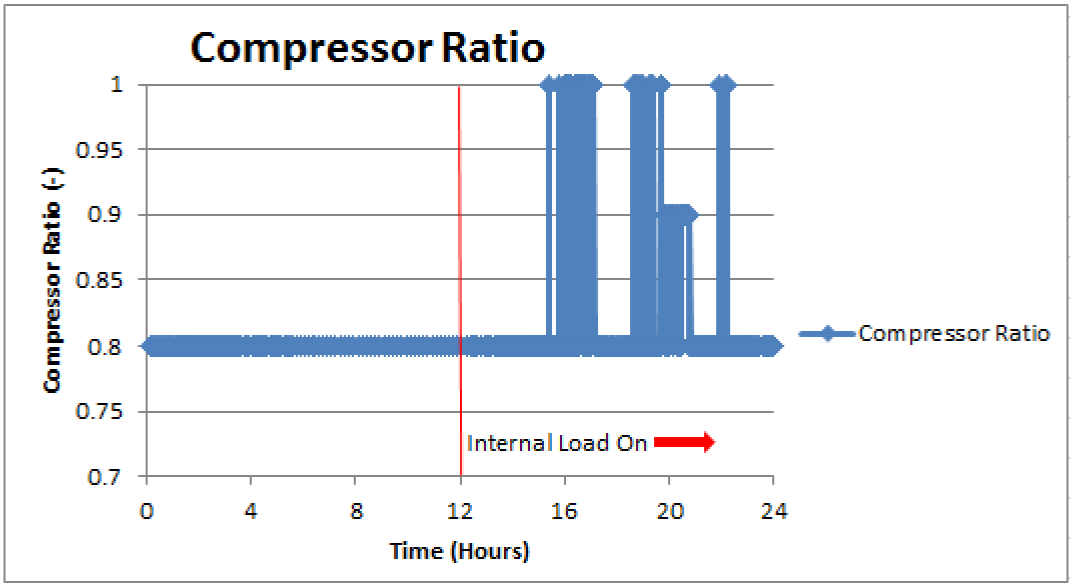


Figure 6.33 Compressor Ratio for September 7

Once again, there is a significant set of time with a compressor ratio fluctuating between 0.8 and 1: from 3:30 to 5 PM and from 6:30 to 7:30 PM. There is also a brief 0.8 to 1 fluctuation at 11 PM. However, there is also a range of 0.9 compressor ratios between 7:45 and 8:45 PM, indicating that the y_4 penalty function component does not dominate the entire time it is called. Note that the 0.9 compressor ratios occur during the period of time when the air temperature exceeds the upper limit of the setpoint deadband. One more value for the y_4 penalty function component was considered to investigate this impact.

6.4.4.3 Case 3

The y_4 penalty function component was halved one last time, resulting in:

$$y_4 = 5 \frac{W}{{}^{\circ}\mathrm{F}^2} * (T_{setpoint} - T_{RA})^2 \qquad (6.6)$$

The air temperatures for the new y_4 is seen in Figure 6.34.

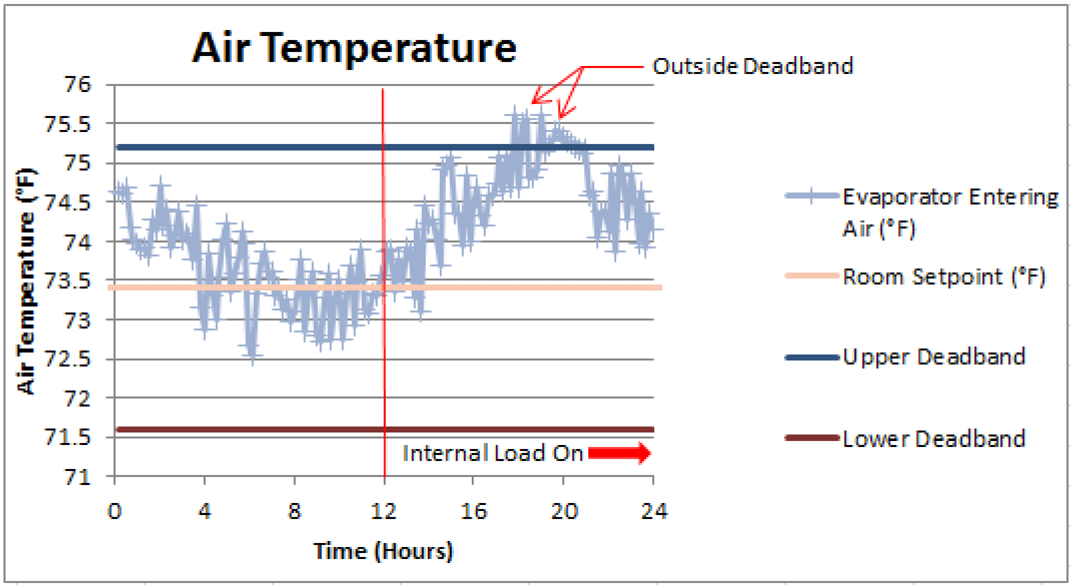


Figure 6.34 Evaporator Entering Air Temperature for September 7

The overall temperature trends follow the same pattern as seen in the previous two cases. However, there are twelve temperature points outside of the deadband between 6 and 8 PM, but they are less than 0.5 °F/0.28 °C above the upper deadband limit. The y_4 = 5 W/°F^2 penalty function component results in an effective +/- 2.3°F/1.28 °C deadband around the 73.4 °F/23 °C setpoint. If this deadband range is acceptable, the temperatures stay within it and power is reduced, as seen in Figure 6.35.

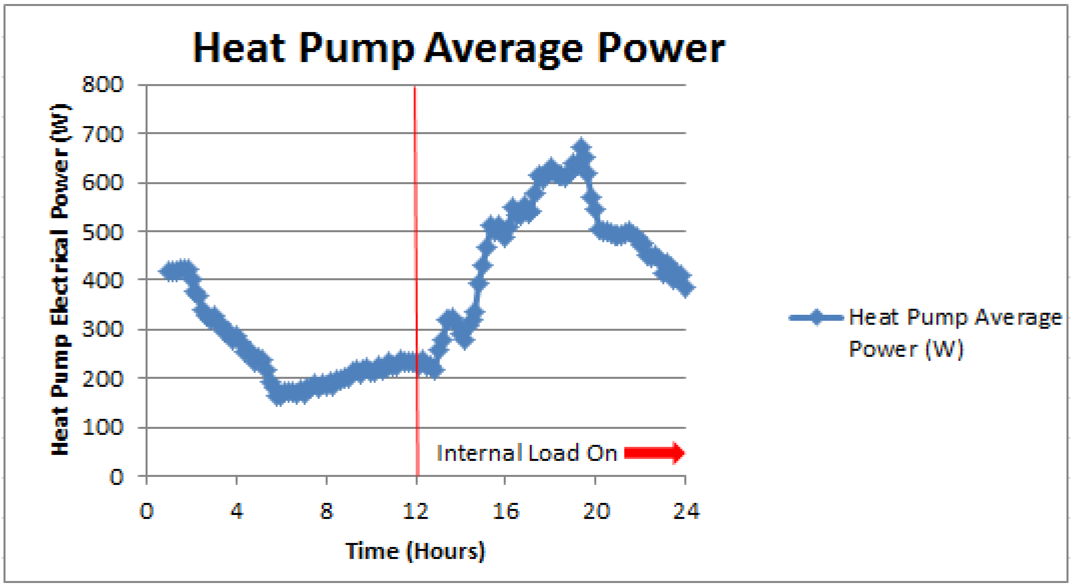


Figure 6.35 Heat Pump Average Power for September 7

The heat pump power remains fairly constant around 620 W between 5:30 and 7 PM before increasing to reach a maximum of 670 W at 7:30 PM. After 7:30 PM, there is a decrease in power until 8 PM, where it reaches a quasi-steady power level just under 500 W. At 9:30 PM, there is a small increase to reach a local maximum of 500 W. After 9:30 PM, the power decreases until it reaches 390 W at midnight. Overall, there is a reduction in power, with a daily power usage of 50.33 kW.

Lastly, Figure 6.36 shows the compressor ratios for the case where the y_4 penalty function component is equal to 5 W/°F^2 at a ΔT of 1 °F/0.56 °C.

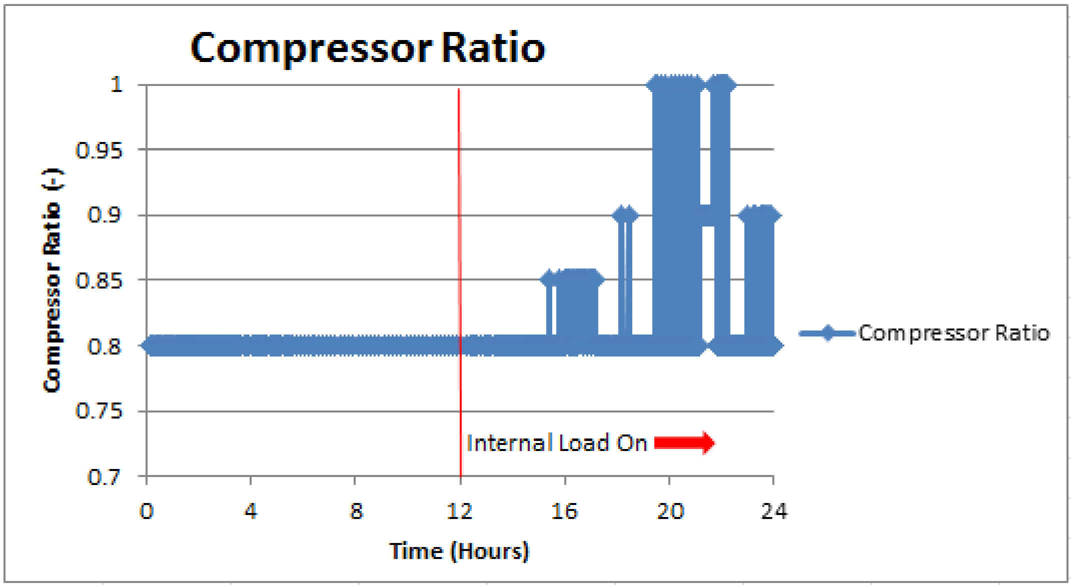


Figure 6.36 Compressor Ratio for September 7

The compressor ratio range spends time at values between 0.8 and 1 in the late afternoon and early evening. The compressor ratio ranges between 0.8 and 0.85 between 3:45 and 4:45 PM, and then reaches 0.9 twice around 6:15 PM before increasing to 1 at 7:30 PM. The compressor ratio then fluctuates between 1 and 0.8 between 7:30 and 10 PM, with a brief interval at 0.9 around 9:45 PM. The compressor ratio drops to 0.8 before returning to 0.9 between 11 PM and 12 AM. Overall, the compressor ratio spends more time at part-load conditions than previous cases, but with a reduced power consumption and little disadvantage in temperature. The compressor ratio part-load conditions indicate sensitivity to the y_4 penalty function component without it overwhelming the simulation.

To check this, Figure 6.37 shows the power from the y_4 penalty function component in comparison with the average heat pump power.

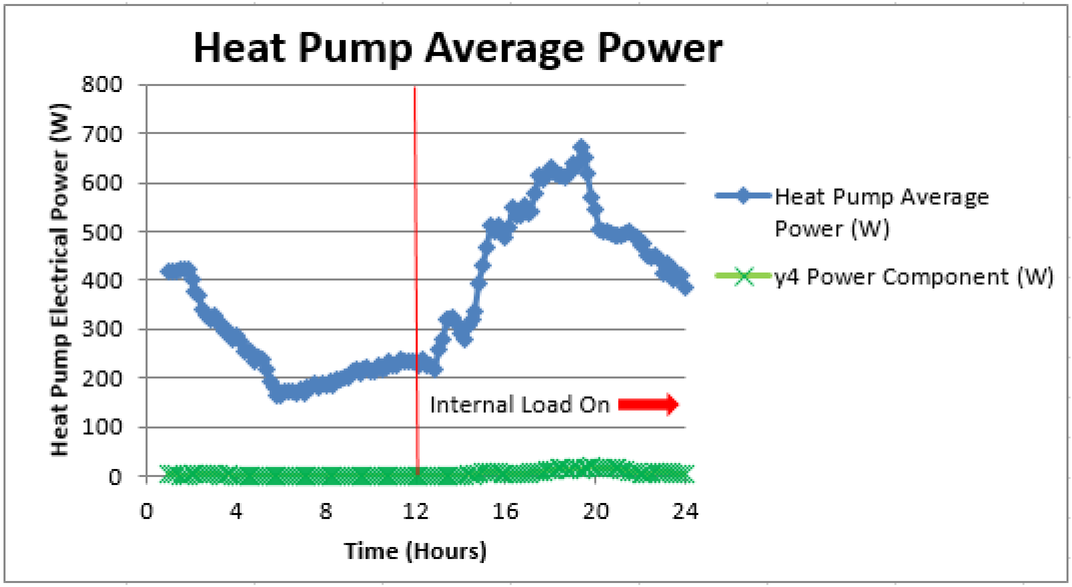


Figure 6.37 Heat Pump Power and y_4 Power Component for September 7

The y_4 penalty function component is insignificant throughout the day, with the highest y_4 power component coming at 7:45 PM. At this time, the y_4 penalty function component was 19 W out of a total reported power of 570 W, less than 4% of the total power. This indicates that the y_4 penalty function component directs the power optimization without dominating it. Figure 6.38 also shows this, showing y_4 increasing non-linearly with heat pump power but at a slower rate.

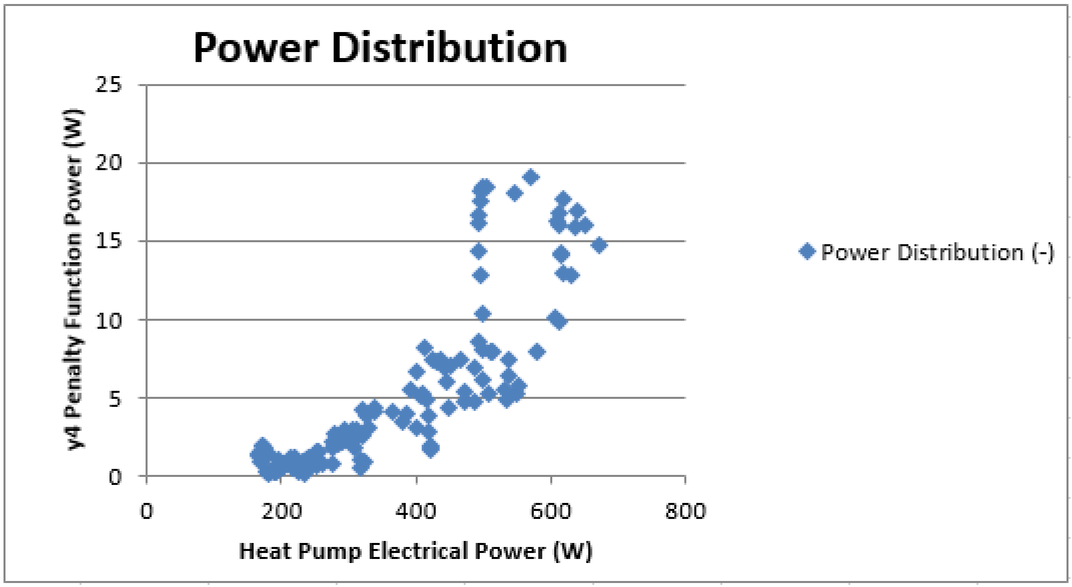


Figure 6.38 Distribution of Heat Pump Power and y_4 Power Component for September 7

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

A simulation environment with the ability to handle multiple buildings and environments has been developed to allow for testing of control strategies for variable-speed heat pump components. Its viability to control via penalty function was demonstrated, as well as the sensitivity of the simulation to the magnitude of the penalty function. This testbed will enable development of control strategies that minimize operating power while still meeting building space conditions. Although this work was done with an air-source heat pump model, the methodology developed should allow any detailed system model to be coupled for controls testing.

In addition to future work that can be pursued using this methodology, there is future work that can be done with this particular model. With some effort, HPSim could be run as a full heat pump model, allowing it to provide heating as well as cooling. This would require implementing HPSim as a heating coil model in EnergyPlus. It could also be brought in as a coil model for Air Loop Equipment; this should not be a major issue, as the call to HPSim resides inside CalcDXCoil, which is also called by the furnace module for Air Loop Equipment objects. The call to HPSim is situated in such a location as to be easily accessible by both the zone equipment routines and the air loop equipment routines. Auxiliary heat could also be added in EnergyPlus to supplement HPSim’s heating capacities. Auxiliary heat should not be set directly by HPSim, as it is an air system component; this problem should be addressed in coupling, using EnergyPlus routines to call auxiliary heat as needed.

Future work should also include fixing the error in the psychrometric properties at dry air boundary conditions, as mentioned in Section 6.1.1. The ability to account for barometric pressure (and thus elevation above sea level) should be implemented in HPSim. In addition, it would be worthwhile to confirm that differences between HPSim air side and refrigerant side heat transfer rates are either a combination of non-significant errors, as proposed in Section 6.1.1. If they are not, the source of the error should be determined and corrected. To that end, it is recommended that the convergence schema in HPSim be thoroughly investigated and potentially revised to reduce the deviation from setpoint and non-convergence issues.

Future work could include diagnostics to identify when HPSim has not converged and methodology for recovery from such an error. For example, if HPSim does not converge on superheat and the evaporator capacity drops drastically, diagnostics could identify this and then take appropriate action. Appropriate action could include interpolating between other known points to produce a reasonable approximation of what the evaporator capacity would be if HPSim had converged.

Although a common problem with solvers, the error with reporting converged results in HPSim should be corrected. The compressor model power calculation should also be investigated, and the difference between the power applied and the accompanying heat transferred to the refrigerant reconciled. The coil model calculations should be investigated to understand why adjusting the air flow rate across the coil has little impact on the coil capacity; if this can be corrected, it should. The refrigerant charge calculations should be investigated for errors, especially the condenser charge calculation routine. In addition, another input should be added to allow the fan power calculation to account for system pressure loss.

There are a few items of future work in the optimization, data manager, and penalty function areas. The GenOpt optimization algorithm could be modified so that it would consider itself converged after one or two steps where it keeps coming back to the same optimized point. The EnergyPlus/GenOpt data manager should have its temperature tolerances tightened when matching inputs. The GenOpt/HPSim data manager should include a humidity ratio component in the penalty function or some other means of controlling the zone humidity. Finally, future work should also include revising the penalty function to account for the total heat transfer rate, not just the sensible heat transfer rate.

REFERENCES

ACHR News. 2000. A Guide to Understanding Filter Drier Functions and Types. Retrieved August 17, 2016 from http://www.achrnews.com/articles/83354-a-guide-to-understanding-filter-drier-functions-and-types

ACHR News. 2001. The Role of the Suction Line Accumulator. Retrieved August 15, 2016 from http://www.achrnews.com/articles/88036-the-role-of-the-suction-line-accumulator

Bushby, S. T., N. Castro, M. A. Galler, and C. Park. 2001. Using the Virtual Cybernetic Building Testbed and FDD Shell Test for FDD Tool Development. Government Publishing Office. Retrieved April 6, 2016 from https://www.gpo.gov/fdsys/pkg/GOVPUB-C13-f27d3f3eee8cd0e396021bb214997d5a/pdf/ GOVPUB-C13-f27d3f3eee8cd0e396021bb214997d5a.pdf

Choi, J., J.T. Chung, and Y. Kim. 2004. A Generalized Correlation for Two-Phase Flow of Alternative Refrigerants through Short Tube Orifices. *International Journal of Refrigeration*. 27.

Crawley, D.B., L.K. Lawrie, C.O. Pedersen, and F.C. Winklemann. 2000. EnergyPlus: Energy Simulation Program. *ASHRAE Journal*. 42.4.

Crawley, D.B., J.W. Hand, M. Kummert, and B.T. Griffith. 2008. Contrasting the Capabilities of Building Energy Performance Simulation Programs. *Building and Environment*. 43.

EnergyPlus. 2013. Weather Data. Retrieved June 7, 2013 from http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_about.cfm

Fischer, S. K., C. K. Rice, and W.L. Jackson. 1988. The Oak Ridge Heat Pump Design Model: Mark III Version Program Documentation. Retrieved June 24, 2019 from https://info.ornl.gov/sites/publications/Files/Pub57074.pdf

Gall, J. 2015. Dynamic Modeling and Optimization Control of Air Conditioning System with Short Cycling Scroll Compressor. Doctoral Dissertation, Oklahoma State University.

Gayeski, N. T. 2010. Predictive Pre-Cooling Control for Low Lift Radiant Cooling Using Building Thermal Mass. Doctoral Dissertation, Massachusetts Institute of Technology.

Gayeski, N. T., P. R. Armstrong, and L. K. Norford. Predictive Pre-Cooling of Thermo-Active Building Systems with Low-Lift Chillers. Part I: Control Algorithm. *ASHRAE Journal* 59.1.

Hensen, Jan L. M. 1999. A Comparison of Coupled and De-Coupled Solutions for Temperature and Air Flow in a Building. *ASHRAE Transactions* 105.2.

Hensen, J. L. M. and M. Trčka. 2010. Overview of HVAC System Simulation. *Automation in Construction*. 19.2.

Iu, I S. 2007. Development of Air-to-Air Heat Pump Simulation Program with Advanced Heat Exchanger Circuitry Algorithm. Doctoral Dissertation, Oklahoma State University.

LBNL. 2016a. Building Controls Virtual Test Bed. Retrieved April 6, 2016 from http://simulationresearch.lbl.gov/bcvtb

LBNL. 2016b. GenOpt. Retrieved May 31, 2016 from http://simulationresearch.lbl.gov/GO/index.html

NIST. 2017. Virtual Cybernetic Building Testbed. Retrieved April 1, 2017 from https://www.nist.gov/laboratories/tools-instruments/virtual-cybernetic-building-testbed

Padhamanabhan, S. 2005. Modeling, Verification and Optimization of Hybrid Ground Source Heat Pump Systems in EnergyPlus. Master's Thesis, Oklahoma State University.

Padhmanabhan, S. 2011. Study of Frost Growth on Heat Exchangers Used as Outdoor Coils in Air Source Heat Pump Systems. Doctoral Dissertation, Oklahoma State University.

Payne, V. and O'Neal D.L. 2004. A Mass Flow Rate Correlation for Refrigerants and Refrigerant Mixtures Flowing through Short Tubes. *HVAC&R Research.* 10.1.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 1996. Numerical Recipes in FORTRAN 77: The Art of Scientific Computing. Vol. 1. 2nd Ed. New York: Cambridge University Press.

Trčka, M., J. Hensen, and A. Wijsman. 2005. Implementation Strategies for Distributed Modeling and Simulation of Building Systems. *Ninth International IBPSA Conference.* Montréal, Canada. August 15-18, 2005.

Trčka, M., M. Wetter, and J. L.M. Hensen. 2007. Comparison of Co-Simulation Approaches for Building and HVAC/R System Simulation. *Tenth International IBPSA Conference.* Beijing, China. July 3-5, 2007.

Trčka, M., M. Wetter, and J. L.M. Hensen. 2009. An Implementation of Co-Simulation for Performance Prediction of Innovated Integrated HVAC Systems in Buildings. *Eleventh International IBPSA Conference.* Glasgow, Scotland. July 27-30, 2009.

Trčka, M., and J. L. M. Hensen. 2010. Overview of HVAC System Simulation. *Automation in Construction*. 19.2.

Trčka, M., J. L.M. Hensen, and M. Wetter. 2010. Co-Simulation for Performance Prediction of Integrated Building and HVAC Systems – An Analysis of Solution Characteristics Using a Two-Body System. *Simulation Modelling Practice and Theory.* 18.7.

USEIA (U.S. Energy Information Administration). 2019. How Much Electricity is Used for Cooling in the United States? Retrieved March 9, 2019 from https://www.eia.gov/tools/faqs/faq.php?id=1174&t=1

Wetter, M. GenOpt – A Generic Optimization Program. *Seventh International IBPSA Conference*. Rio de Janeiro, Brasil. August 13-15, 2001.

Wetter, M. 2011. Co-Simulation of Building Energy and Control Systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation*, 4.3.

Zakula, T. 2013. Model Predictive Control for Energy Efficient Cooling and Dehumidification. Doctoral Dissertation, Massachusetts Institute of Technology.

APPENDICES

APPENDIX A

This appendix contains discussion and examples of HPSim program inputs and outputs.

The calculation (simulation or design mode) and system types (heating or cooling mode), coil and expansion device properties, type of refrigerant, and circuit properties are important inputs to HPSim. The same structure is followed throughout the entire input file; for an example, a section of the circuiting portion of the input file will be shown. The first part of the circuiting input is the section that tells how many tubes are in each circuit. In the following case, there are 20 in the first circuit.

```
IDCcktCircuiting_TubeNumbers,
    20, !Number of Tubes in Circuit 1
. . .
```

Each circuit can have a different number of tubes, or they may all have the same number. The second part of the circuiting input lists the tube sequence; this tells the program in what order the tubes are connected. The tube's numbers do not denote their connection order (e.g. 1 does not necessarily connect to 2, nor 2 to 3) but are an ordered counting of the tubes from one corner of the heat exchanger to the end. Here, 81 is the first or inlet tube to the circuit; its outlet will connect to the inlet of 82. 82's outlet will connect to 83's inlet, 83's outlet will connect to 84's inlet, and so on.

```
IDCcktCircuit1_TubeSequence,
    81, !Tube Number
    82, !Tube Number
    83, !Tube Number
    84, !Tube Number
    85, !Tube Number
    86, !Tube Number
    87, !Tube Number
    88, !Tube Number
    89, !Tube Number
    90, !Tube Number
    11, !Tube Number
    12, !Tube Number
    13, !Tube Number
    14, !Tube Number
    15, !Tube Number
    16, !Tube Number
    17, !Tube Number
    18, !Tube Number
    19, !Tube Number
    20; !Tube Number
```

Other important inputs passed through from the input file are the indoor and outdoor entering air conditions, as shown below; in the coupled version of the heat pump code these are set by EnergyPlus instead.

```
95, !field Outdoor Entering Drybulb Temperature [°F]
75, !field Outdoor Entering Wetbulb Temperature [°F]
80, !field Indoor Entering Drybulb Temperature [°F]
67, !field Indoor Entering Wetbulb Temperature [°F]
```

The short tube and capillary tube dimensions, and TXV capacity are useful outputs for sizing if the program is run in design mode; an example of the outputs are shown below.

```
Short tube diameter,                 0.069,  (in)
TXV capacity,                        0.000, (ton)
Capillary tube diameter,             0.078,  (in)
Capillary tube length,               12.000, (in)
```

The power and gross capacity requirements are useful for equipment sizing, as are the COP, SHR, and EER. Example outputs for these values are shown below.

```
     Evaporator gross capacity,          25973.715, (Btu/h)
     Condenser gross capacity,           30193.484, (Btu/h)
     Gross sensible capacity,            24973.715, (Btu/h)
     Gross latent capacity,              1000.000,  (Btu/h)
     Compressor power,                   1213.000,  (W)
     Condenser Fan power,                23.903,    (W)
     Total Power,                        1241.347,  (W)
     Refrigerant mass flow rate,         343.444,   (lbm/h)
     COP (coefficient of performance),   6.121,     (-)
     EER (energy efficiency ratio),      20.912,    (Btu/W-hr)
     SHR (sensible heat ratio),          1.000,     (-)
```

The refrigerant charge, mass flow rate, refrigerant state variables, coil capacities, and compressor power are also among the important outputs. An example of the refrigerant property outputs is shown below.

```
Location, Temperature (F), Pressure (psig), Enthalpy (Btu/lbm),
Saturation Temperature (F), Quality (%), Subcooling (R),
Superheat (R)

Evaporator inlet, 45.738, 132.029, 113.412, 45.738, 23.7, 0.00,
0.000
Evaporator outlet, 55.369, 130.899, 185.488, 45.270, 100.0, 0.00,
9.916
```

A sample set of air properties given in the output file is seen below.

```
Location, Dry bulb temperature (F), Wet bulb temperature (F),
Relative Humidity (%), Volumetric flow rate (CFM), Pressure Drop
(in-H2O)

Evaporator inlet, 80.005, 67.004, 51.1, 900.000, 0.000
Evaporator outlet, 57.046, 54.535, 85.6, 900.000, 0.030
```

VITA

Rachel Ann Spitler

Candidate for the Degree of

Master of Science

Thesis: DEVELOPMENT OF A DETAILED HEAT PUMP SIMULATION CONTROLS TESTBED

Major Field: Mechanical Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2019.

Completed the requirements for the Bachelor of Science in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in 2012.

Experience:

Summer Intern at CDM in Dallas, TX in 2010, 2011, 2012.

Research Assistant at Oklahoma State University from May 2012 to December 2014.

Engineering Intern at Cyntergy in Tulsa, OK from January 2015-present.

Professional Memberships:

ASHRAE, IBPSA.