DYNAMIC MODELING AND WAVELET-BASED MULTI-PARAMETRIC TUNING AND VALIDATION FOR HVAC SYSTEMS

A Dissertation

by

SHUANGSHUANG LIANG

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Bryan Rasmussen
Committee Members,	Jeff Haberl
	Daniel McAdams
	Michael Pate
Head of Department,	Andreas Polycarpou

August 2014

Major Subject: Mechanical Engineering

ABSTRACT

Dynamic Heating, Ventilation, and Air-Conditioning (HVAC) system models are used for the purpose of control design, fault detection and diagnosis, system analysis, design and optimization. Therefore, ensuring the accuracy and reliability of the dynamic models is important before their application. Parameter tuning and model validation is a crucial way to improve the accuracy and reliability of the dynamic models. Traditional parameter tuning and validation methods are generally time-consuming, inaccurate and can only handle a limited number of tuning parameters. This is especially true for multiple-input-multiple-output (MIMO) models due to their intrinsic complexity.

This dissertation proposes a new automatic parameter tuning and validation approach to address this problem. In this approach, a fast and accurate model is derived using linearization. Discrete-time convolution is then applied on this linearized model to generate the model outputs. These outputs and data are then processed through wavelet decomposition, and the corresponding wavelet coefficients obtained from it are used to establish the objective function. Wavelets are advantageous in capturing the dynamic information hidden in the time series. The objective function is then optimized iteratively using a hybrid method consisting of a global search genetic algorithm (GA) and a local gradient search method.

In order to prove the feasibility and robustness of the proposed approach, it is applied on different dynamic models. These models include an HVAC system model with moving boundary (MB) heat exchanger models, a heat pump model with finite

control volume (FCV) heat exchanger models, and a lumped parameter residential conditioned space model. These models generally have a large number of parameters which need tuning. The proposed method is proved to be efficient in tuning single data set, and can also tune the models using multiple experimental or field data sets with different operating conditions. The tuned parameters are further cross-validated using other data sets with different operating conditions. The results also indicate the proposed method can effectively tune the model using both static and transient data simultaneously.

DEDICATION

To my parents and twin sister

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Bryan Rasmussen, for his continuous and excellent guidance and financial support throughout the length of my research. I would also like to thank my committee members, Professor Jeff Haberl, Professor Daniel McAdams, and Professor Michael Pate for serving on my advisory committee and providing valuable advice.

Thanks to all my colleagues at Thermo-Fluids Control Laboratory for sharing laughter and enlightening conversations. I am also grateful to my friends who have helped to make my time at Texas A&M University a memorable experience. I also would like to acknowledge Emerson Climate Technologies for providing financial contribution.

Finally, thanks to my parents and twin sister, for their endless encouragement, support and love.

NOMENCLATURE

Variables	Definition
A	Area
C	Heat Capacity
C_d	Valve Discharge Coefficient
E_t	Incident Irradiance
L	Length
N	Number of Control Volumes
P	Pressure; Parameters
P_a	Atmosphere Pressure
POI	Percent of Improvement
POT	Percent of Time
Q	Heat Transfer
R	Thermal Resistance
T	Temperature
U	Overall Heat Transfer Coefficient
V	Volume
c_p	Specific Heat
d	Hydraulic Diameter; coefficients
h	Specific Enthalpy
k	Thermal Conductivity

m	Mass
$\dot{m}$	Mass Flow Rate
q	Heat Transfer
u	Internal Energy
x	Wall Thickness
α	Heat Transfer Coefficient
β	Solar Altitude
ρ	Fluid Density
λ	Normalized Length
ϕ	Relative Humidity
ε	Heat Transfer Effectiveness
ω	Compressor Speed; Specific Humidity
η	Efficiency
Δp	Pressure Drop

Subscripts/Superscripts	Definition
0	Equilibrium Point; 0 ºC
1,2	1^{st}, 2^{nd} region
CA	Circulated Air
EA	Exhaust Air
MA	Mixed Air
OA	Outside Air

RA	Return Air
SA	Supply Air
WD	Windows and Doors
a	Adiabatic; Air; Air Side; Atmosphere
c	Condenser; Ceiling; Coils
cs	Cross-Sectional
d	Difference
e	Evaporator
exf	Exfiltration
f	Floor
g	Saturated Vapor
i	Internal; Inner; Inlet
in	Internal; Inlet
inf	Infiltration
int	Intermediate
k	Compressor
o	Out; External; Outer
pf	Projected (Fenestration)
r	Room
s	Saturated; Energy Source Provided by the Humidifier or People
sp	Set Point
solar	Solar

t	Total
tot	Total
total	Total
tuned	Tuned
v	Volumetric; Vapor
vol	Volumetric
w	Wall; Water
*	Wet Condition; Updated

Abbreviations	Definition
CFM	Cubic Feet per Minute
EMCS	Energy Management Control Systems
EXV	Electronic Expansion Valve
FCV	Finite Control Volume
HP	Heat Pump
HVAC	Heating, Ventilation, and Air Conditioning
GA	Genetic Algorithm
MB	Moving Boundary
MIMO	Multiple Input Multiple Output
NTU	Number of Transfer Units
RMS	Root Mean Square
SHGC	Solar Heat Gain Coefficient

TXV	Thermostatic Expansion Valve
VCS	Vapor Compression System

Mathematical Notation	Definition
A,B,C,D	State Space Matrices
G	Impulse Response Matrix
J	Cost Function
M	First Non-Zero Value of the Input
N	First Non-Zero Value of the Impulse Response
Z	Matrix
c	Coefficients
f	Function in Nonlinear Descriptor Form
g	Output Function; Impulse Response
u	Inputs
x	States
y	Outputs
w	Wavelet Coefficients
α	Weighting Factor
φ	Haar Scale Function
ε	Relative Error
ω	Decomposed Signals
ζ	Original Signal

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGEMENTS v

NOMENCLATURE vi

TABLE OF CONTENTS xi

LIST OF FIGURES xiv

LIST OF TABLES xix

CHAPTER I INTRODUCTION AND LITERATURE REVIEW 1

Dynamic Modeling of HVAC Systems 3
Parameter Tuning and Model Validation 11

CHAPTER II DYNAMIC MODELING OF HVAC AND BUILDING SYSTEMS 17

Moving Boundary Heat Exchanger Models 18
FCV Heat Exchanger Model 20
Mass Flow Device Models 25
FCV Cooling Coil Model 25
Model Assumptions 26
Model Derivation 27
Model Verification 33
Residential Conditioned Space Model 40
Model Assumptions 40
Model Derivation 42
Model Verification 45

CHAPTER III DEVELOPMENT OF A WAVELET-BASED PARAMETER TUNING AND VALIDATION METHOD 53

Wavelets Application in Signal Processing 53
Hybrid Optimization 60

Algorithm Configuration 62
Development of Nonlinear Model 63
Model Linearization 64
Discretization and Convolution 65
Construction of Objective Function 66
Hybrid Optimization and Transient Point 66
Wavelet-Based Objective Function 68

CHAPTER IV VERIFICATION OF PROPOSED PARAMETER TUNING AND VALIDATION METHOD 69

Linear Model with Discrete-Time Convolution vs. Nonlinear Model 70
Verification Using Simulated Data 72
Experimental Test 78
Experimental Apparatus 78
Test Results 81
Heat Exchanger Model Parameter Tuning and Validation 86
HVAC Model Parameter Tuning and Validation 88
Selection of Tuning Parameters 88
Selection of Transition Point 90
Determination of Weighting Factor 95
Parameter Tuning and Validation Using Single Data Set 98
Parameter Tuning and Validation Using Multiple Data Sets with Different Operating Conditions 99
Cross Validation 105

CHAPTER V APPLICATION TO EMERSON VARIABLE SPEED HEAT PUMP MODEL 109

Emerson Heat Pump System Configuration 111
Field Data 113
Pseudo-Steady State Data Generation from Field Test Data 116
Pseudo-Steady Data 117
Expansion Valves and Compressor Parameters Generated from Pseudo-Steady Data 121
Corrections of Pseudo-Steady State Data 125
Heat Pump Model Validation Using Steady State Data 133
FCV Heat Exchanger Model 133
Complete Heat Pump Model 136
Simultaneous Parameter Tuning Using both Steady State and Transient Data 141
FCV Evaporator Model 142
FCV Condenser Model 151
Complete Heat Pump Model 159

Simultaneous Parameter Tuning Using Multiple Field Data Sets with Different Operating Conditions 163
Cross Validation 166

CHAPTER VI APPLICATION TO RESIDENTIAL CONDITIONED SPACE MODEL 169

Establishing a House Model Using EnergyPlus 169
Parameter Tuning and Validation Using Steady State Data 172
Simultaneous Parameter Tuning and Validation Using EnergyPlus Data without Heat Pump System 173
Simultaneous Parameter Tuning and Validation Using EnergyPlus Data with Heat Pump System 175

CHAPTER VII CONCLUSION 178

Future Work 179

REFERENCES 181

APPENDIX A 196

One Sample Tuning m-file in Matlab/Simulink 196

APPENDIX B 208

Input File of EnergyPlus 208

LIST OF FIGURES

Page

Figure 1.1: A simple HVAC system operating using a vapor compression cycle. 5

Figure 1.2: Typical structures of cooling coils [21]. 9

Figure 2.1: A moving boundary evaporator model with two regions. 19

Figure 2.2: A moving boundary condenser model with three regions. 21

Figure 2.3: An FCV evaporator model. 24

Figure 2.4: A four-row cooling coil. 28

Figure 2.5: An FCV cooling coil schematic. 29

Figure 2.6: Step change of inlet water mass flow rates (dry initial condition). 33

Figure 2.7: Step change of inlet air mass flow rates (dry initial condition). 35

Figure 2.8: Step change of inlet air humidity (dry initial condition). 36

Figure 2.9: Step change of inlet water mass flow (wet initial condition). 37

Figure 2.10: Step change of inlet air flow rates (wet initial condition). 38

Figure 2.11: Step change of inlet air humidity (wet initial condition). 39

Figure 2.12: Residential Conditioned Space Model. 41

Figure 2.13: Step response of supply and return air mass flow rate. 47

Figure 2.14: Step response of supply air humidity. 48

Figure 2.15: Step response of outside temperature. 49

Figure 2.16: Feedback control with room setpoint temperature change. 50

Figure 2.17: Feedback control with supply and return air mass flow rates changes. 51

Figure 2.18: Feedback control with room setpoint humidity change. 52

Figure 3.1: Common types of wavelets [77]. 57

Figure 3.2: A view of wavelet decomposition basis functions. 58

Figure 3.3: Configuration of parameter estimation and tuning method. 63

Figure 4.1: Linearized discrete-time model outputs vs. nonlinear model outputs. 71

Figure 4.2: Inputs and outputs of MB evaporator model. 73

Figure 4.3: Input change of MB evaporator model established in MATLAB/Simulink. 73

Figure 4.4: Outputs of linearized moving boundary evaporator model established in MATLAB/Simulink. 74

Figure 4.5: Reconstructed model outputs vs. original model outputs. 75

Figure 4.6: Monte-carlo simulation results. 76

Figure 4.7: Tuning results using the evaporator simulated data. 77

Figure 4.8: Experimental system. 79

Figure 4.9: Inputs of low cooling test. 82

Figure 4.10: Outputs of low cooling test. 83

Figure 4.11: Inputs of high cooling test. 84

Figure 4.12: Outputs of high cooling test. 85

Figure 4.13: Tuned results using the evaporator experimental data. 86

Figure 4.14: Objective function trajectory using only the GA. 91

Figure 4.15: Objective function trajectory for different transition points. 94

Figure 4.16: Tuning results with different weighting factors. 97

Figure 4.17: Parameter tuning and validation results using single data set. 99

Figure 4.18: Parameter tuning and validation results using multiple data sets with the same tuning parameters (data set 1). 101

Figure 4.19: Parameter tuning and validation results using multiple data sets with the same tuning parameters (data set 2). 102

Figure 4.20: Parameter tuning and validation results using multiple data sets with different tuning parameters (data set 1). 103

Figure 4.21: Parameter tuning and validation results using multiple data sets with different tuning parameters (data set 2). 104

Figure 4.22: Cross validation of the tuned solution on linearized model. 106

Figure 4.23: Cross validation of the tuned solution on nonlinear model. 107

Figure 5.1: Variable speed heat pump system. 110

Figure 5.2: Variable speed heat pump system configuration (heating mode). 112

Figure 5.3: Variable speed heat pump system configuration (cooling mode) 113

Figure 5.4: Rheem 5T field cooling data 05/30/2011. 114

Figure 5.5: Coleman 5T field cooling data 07/15/2012. 115

Figure 5.6: A pseudo-steady range for evaporator pressure. 117

Figure 5.7: Rheem 5T heating pseudo-steady state data. 118

Figure 5.8: Rheem 5T cooling pseudo-steady state data. 119

Figure 5.9: Coleman 5T cooling pseudo-steady state data. 120

Figure 5.10: TXV mass flow rate comparison using generated parameters (Rheem 5T pseudo-steady state cooling data). 121

Figure 5.11: TXV mass flow rate comparison using generated parameters (Coleman 5T pseudo-steady state cooling data). 122

Figure 5.12: EXV mass flow rate comparison using generated parameters (Rheem 5T pseudo-steady state heating data). 123

Figure 5.13: Compressor mass flow rate and power comparison using generated parameters (Rheem 5T pseudo-steady state cooling data). 124

Figure 5.14: Compressor mass flow rate and power comparison using generated parameters (Coleman 5T pseudo-steady state cooling data). 125

Figure 5.15: Heat transfer comparison for heat exchangers (Rheem 5T heating data). 127

Figure 5.16: CFM correction for heat exchangers (Rheem 5T heating data) 128

Figure 5.17: Heat transfer comparison for heat exchangers (Rheem 5T cooling data). 129

Figure 5.18: CFM correction for heat exchangers (Rheem 5T cooling data). 130

Figure 5.19: Heat transfer comparison for heat exchangers (Coleman 5T cooling data). 131

Figure 5.20: CFM correction for heat exchangers (Coleman 5T cooling data). 132

Figure 5.21: Refrigerant side heat transfer between tuned model and pseudo-steady state data. 133

Figure 5.22: Evaporator FCV model validation using Rheem 5T pseudo-steady state cooling data. 134

Figure 5.23: Evaporator FCV model validation using Coleman 5T pseudo-steady state cooling data. 135

Figure 5.24: Condenser FCV model validation using Rheem 5T pseudo-steady state heating data. 136

Figure 5.25: Condenser FCV model validation using Coleman 5T pseudo-steady state cooling data. 136

Figure 5.26: Complete HP model validation using Rheem 5T pseudo-steady state heating data. 137

Figure 5.27: Complete HP model validation using Rheem 5T pseudo-steady state cooling data. 139

Figure 5.28: Complete HP model validation using Coleman 5T pseudo-steady state cooling data. 140

Figure 5.29: Evaporator FCV model validation using Rheem 5T field cooling data 05/30/2011. 143

Figure 5.30: Evaporator FCV model validation using Rheem 5T field cooling data 08/01/2011. 145

Figure 5.31: Evaporator FCV model validation using Rheem 5T field cooling data 07/19/2011. 147

Figure 5.32: Evaporator FCV model validation using Coleman 5T field cooling data 05/17/2012. 148

Figure 5.33: Evaporator FCV model validation using Coleman 5T field cooling data 07/15/2012. 150

Figure 5.34: Condenser FCV model validation using Rheem 5T field cooling data 06/06/2011. 153

Figure 5.35: Condenser FCV model validation using Rheem 5T field cooling data 07/19/2011. 154

Figure 5.36: Condenser FCV model validation using Coleman 5T field cooling data 05/17/2012. 156

Figure 5.37: Condenser FCV model validation using Coleman 5T field cooling data 07/15/2012. 158

Figure 5.38: Complete HP model validation using Rheem 5T field cooling data 06/06/2011. 160

Figure 5.39: Complete HP model validation using Coleman 5T field cooling data 05/17/2012. 162

Figure 5.40: Evaporator FCV model multiple data validation using Rheem 5T transient cooling data 05/30/2011 and 06/26/2011. 165

Figure 5.41: Evaporator FCV model cross validation for Rheem 5T transient cooling data 07/05/2011 with tuned parameters obtained from Figure 5.40. 167

Figure 6.1: Setting up a house model in EnergyPlus. 170

Figure 6.2: View of the residential conditioned space house. 171

Figure 6.3: Residential model steady state validation. 173

Figure 6.4: Ten days weather inputs to the residential model in January. 174

Figure 6.5: Residence model validation results without heat pump. 175

Figure 6.6: Residence model validation results with heat pump (heating). 176

Figure 6.7: Ten days weather inputs to the residential model in July. 177

Figure 6.8: Residence model validation results with heat pump (cooling). 177

LIST OF TABLES

Page

Table 4.1: Computation time of nonlinear and linear model. 72

Table 4.2: Specifications of the sensors. 80

Table 4.3: Summary of parameter tuning and validation results. 87

Table 4.4: Influences of different parameters on the output error..... 89

Table 4.5: The definition of different transition points. 93

Table 4.6: Comparison of parameter tuning performed during each optimization algorithm. 108

Table 5.1: Time derivative tolerances for pseudo-steady state data generation..... 116

Table 5.2: Relative error of evaporator FCV model validation using field data..... 144

Table 5.3: Relative error of condenser FCV model validation using field data..... 152

Table 5.4: Relative error of complete HP model validation using field data. 161

Table 6.1: Inputs of EnergyPlus model (EnergyPlus configuration). 172

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

Heating, Ventilation, and Air-Conditioning (HVAC) systems are widely used in commercial, residential and military sectors for heating and cooling homes. Almost 40% of the energy consumed in these buildings is from HVAC systems in the U.S. [1]. Improving the efficiency of these systems is critical to minimizing energy and environmental costs.

Dynamic HVAC system behavior is hard to predict. Fortunately, computing technology can be used to create HVAC models and perform simulation analysis which can be used as a means to understand key variables and their effects and causes. In this way, qualitative and quantitative predictions of real HVAC systems can be obtained. Therefore, the need to construct validated dynamic models for the purpose of control design, fault detection and diagnosis, system analysis, design and optimization is highly important. This approach is known as model validation, i.e. a process to establish the validity of the model. However, Rasmussen [2] stated that "model invalidation" may be a more proper term, since the major effort in model validation is determining when the predicted outputs given by the model cannot match with the data.

Parameter tuning and model validation is the process of finding optimal HVAC system parameters such that the measured data obtained from the actual system match exactly with the predicted data from the model. It is achieved by exploring the sensitivity of the model predictions to parameter variations [2]. The reason why the parameters

need to be tuned is due to the assumptions made during modeling, during which these parameters become lumped. Developing an efficient parameter tuning and model validation method that achieves accuracy and reduces computing cost is important to obtain accurate models for modern HVAC systems.

The main goal of this dissertation is to develop a fast and automatic parameter tuning and validation method for nonlinear HVAC system models that improves model accuracy and acceptable level of computing cost. This dissertation will utilize wavelet decomposition theory, discrete-time convolution and linearization to develop an efficient parameter tuning and validation method. Linearization and discrete-time convolution can help estimate the proper parameters effectively by shortening the computation time and reducing computation costs. By establishing an objective function using wavelet coefficients, the transient information in the time series can be fully captured. The objective function can be easily adjusted to tune large data sets with multiple operating conditions.

The research goal is to improve model capability of matching outputs with experimental and field data for HVAC system models. The specific research objectives include:

- Reducing nonlinearity of HVAC models
- Establishing objective functions that can fully capture dynamics in time series for single and multiple outputs
- Simultaneously tuning static and transient data

- Simultaneously tuning multiple data with different operating conditions
- Demonstrating the proposed method by application to experimental air-conditioning system and residential heat pump system models

The remainder of this dissertation is organized as follows. Chapter I describes fundamentals and background on dynamic modeling, parameter tuning and model validation. Chapter II presents the dynamic models that will be used in this dissertation, followed by the specific description of the proposed parameter tuning and model validation method. The verification of proposed method is then demonstrated in Chapter IV. Chapter V and Chapter VI present the application of this method to Emerson heat pump systems and residential conditioned space models. Conclusions and recommendations for future work are given in Chapter VII.

Dynamic Modeling of HVAC Systems

The bulk of HVAC systems operate using a vapor compression system cycle. The challenge of modeling these vapor compression systems (VCS) is that they are actually extremely complex systems, containing components with different time scale dynamics, i.e. stiff dynamics. These components include heat exchangers, cooling coils, compressors, expansion valves, tubes, accumulators, receivers, charge compensators, etc. The dynamics in compressors and expansion valves evolve on much faster time scales than the heat exchangers. Dynamic modeling of HVAC systems has to deal with each component, and the interconnection between them. A simple HVAC system may

consist of an evaporator, a condenser, an electronic expansion valve (EXV) and a compressor, as is shown in Figure 1.1. The refrigerant flow in this system is described as follows.

- Process 1-2: Isobaric condensation. The refrigerant with high temperature and high pressure from the compressor enters the condenser and releases heat to the external fluid (typically air or water), reaching the state 2: saturated or sub-cooled liquid with high pressure and temperature.
- Process 2-3: Isenthalpic expansion. The refrigerant is then expanded in the expansion valve to become a low-temperature and low-pressure liquid. The state of refrigerant at point 3 is a two phase with low pressure and temperature.
- Process 3-4: Isobaric evaporation. The two phase fluid at 3 enters the evaporator, where it absorbs the heat from the external fluid and evaporates to a superheated or saturated vapor with low temperature and pressure.
- Process 4-1: Isentropic compression. The vapor with low pressure and low temperature is then compressed to a high pressure and temperature vapor and enters the condenser to repeat the cycle.

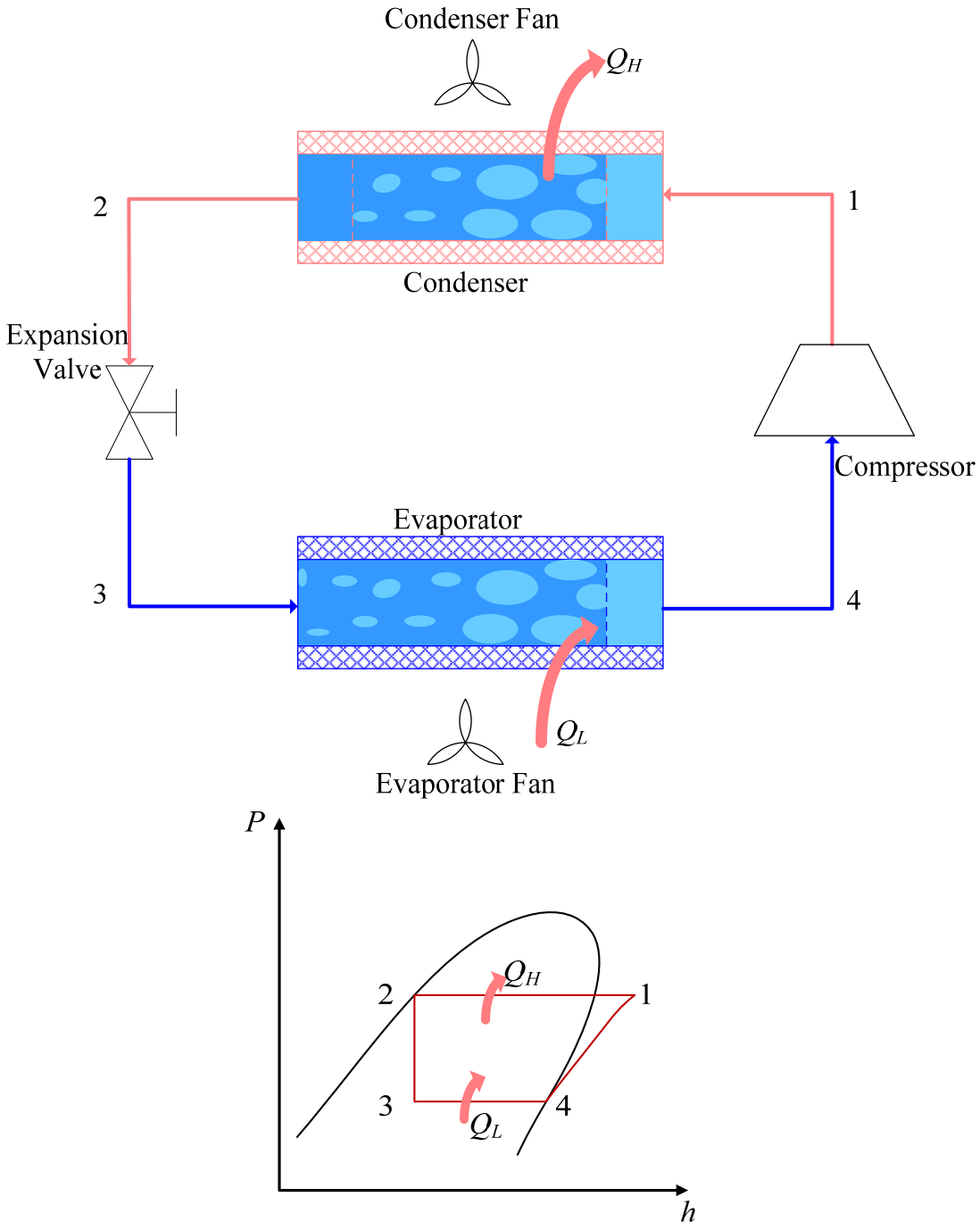


Figure 1.1: A simple HVAC system operating using a vapor compression cycle.

Dynamic modeling of HVAC systems largely focuses on heat exchangers, due to the following two reasons. First, the dynamics in heat exchangers are more complicated and hard to predict due to the phase change of the refrigerant. Second, dynamics in compressors and expansion valves are relatively much faster than heat exchangers. Thus modeling of these mass flow devices is simple: using instant algebraic relationships is sufficient [2].

The core of modeling heat exchangers lies in how to model two-phase dynamics. Three methods are often used in modeling these heat exchangers: lumped parameter method, moving boundary (MB) method and finite control volume (FCV) method [3]. In lumped parameter method, lumped parameters are assumed for the whole system or the fluid phases. The moving boundary method is essentially a type of lumped parameter method. In this method, each fluid region has its lumped parameters, and a series of time-varying boundaries with dynamic features for each region are assumed. In the finite control volume method, the heat exchanger is discretized into many regions to include more details about thermo-physical gradients and distributed parameters, thus increasing the accuracy of the model and computation cost at the same time. In all these methods, the governing differential equations are first obtained. For moving boundary method, these partial differential equations are obtained for fluid flow in a tube [4]; they are then integrated along the length of the heat exchanger to obtain ordinary differential equations [5][6]. For the finite control volume method, the governing equations for each region are obtained by discretizing the heat exchanger into many control volumes, or by discretizing the governing partial differential equations using finite difference method.

The moving boundary approach was originated from Wedekind and Stoecker [7]'s work, in which they used the term "transient point". In 1997, Narayanan et al. [8] developed a lumped parameter evaporator model, in which they included pressure drop and heat flux variations along the axis. Willatzen [9] used the moving boundary approach for dynamic modeling of heat exchangers.

In 1998, Mithraratne et al. [10] established a distributed parameter model of simulating the dynamics of an evaporator with a thermostatic expansion valve (TXV). They also considered the effect of the axial heat conduction in the pipes. Tummescheit and Eborn [11] discussed the modeling of a thermo-hydraulic model using lumped parameter and distributed parameter methods using commercial software known as Modelica.

In 2002, Bendapudi [12] presented a detailed literature review of notable prior efforts in dynamic modeling of vapor compression systems. He then compared and validated the moving boundary and finite volume methods in 2004 [13]. In 2008, he [14] comprehensively compared FCV technique and moving boundary technique. According to him, moving boundary dynamic models have the advantages of computation speed and cost. They can give results that are very close to that of FCV models. However, FCV models are more capable of capturing the nonlinear dynamics which are the key characteristics of nonlinear systems. This dissertation will investigate the effectiveness of proposed parameter tuning and validation on both moving boundary and FCV heat exchanger models.

Hemami and Dunn [15] used FCV approach to establish the evaporator and condenser model, and semi-empirical approach to develop the compressor and expansion valve model. They only performed steady state model validation to reach an error of less than 15%. Gupta [16] conducted extensive modeling using FCV method. Dhar [17] modeled a refrigeration system using moving boundary method. He validated this model using experimental data with different operating conditions. Chi and Didion [18] also established and validated their lumped parameter vapor compression system model using a 4-ton heat pump. In Gruhle and Isermann [19]'s work, they used the lumped parameter method to establish the vapor compression system model. Wedekind and Bhatt [20] successfully validated their evaporator and condenser model using transient data.

In dynamic modeling of HVAC components, the cooling coil is always a big challenge. Air cooling coils are widely used in HVAC systems to lower air temperature for human comfort. These finned coils are typically serpentine, and refrigerant or chilled water flows inside the tubes to absorb heat from the air flowing outside. Modern cooling coils often employ sophisticated geometry structure to maximize the heat transfer efficiency. Specifically, cooling coils usually have multiple rows and extensive areas of fins. These geometries severely influence the physical parameters, and heat and mass transfer. Figure 1.2 indicates some typical structures of cooling coils produced by Super Coils [21].

Extensive research has been done to develop the empirical correlations of cooling coil heat transfer coefficients as a function of pressure, temperature, mass flow rate and geometry. Wattelet [22] reviewed the previous study in developing heat transfer

coefficients in horizontal-tube heat exchangers. Much literatures focuses on the single-phase heat transfer of refrigerant flow in the tubes. The most widely known single-phase heat transfer correlation in a pipe is given by Dittus and Boelter [23], which is shown below.

Figure 1.2: Typical structures of cooling coils [21].

$$Nu = 0.023 \mathrm{Re}^{0.8} \mathrm{Pr}^{0.4} \tag{1.1}$$

Other common correlations include Gnielinski correlation [24]:

$$Nu = \frac{\frac{f}{2}(\mathrm{Re}-1000)\,\mathrm{Pr}}{1+12.7\sqrt{\frac{f}{2}}(\mathrm{Pr}^{2/3}-1)}\left[1+(\frac{D_i}{L})^{2/3}\right](\frac{\mathrm{Pr}}{\mathrm{Pr}_w})^{0.11} \tag{1.2}$$

and Sieder-Tate equation [25]:

$$Nu = 0.023\,\mathrm{Re}^{0.8}\,\mathrm{Pr}^{1/3}(\frac{\eta_m}{\eta_{wall}})^{0.14} \tag{1.3}$$

Two-phase heat transfer coefficient research can be dated back to as early as 1960s. There are numerous correlations. Kandlikar [26] summarized some well-known correlations. Chen [27]'s correlation is very simple but can yield large error under some circumstances, which have been indicated by Jallouk [28] and Mohr and Runge [29]. In 1990, Kandlikar[30] proposed a more general correlation for two-phase heat transfer coefficient inside both horizontal and vertical tubes.

In this dissertation, the convective heat transfer inside chilled water cooling coils adopts Gnielinski's method [24]. The heat transfer coefficient varies along the tube, and it can be captured by an FCV chilled water cooling coil model.

Cooling coils often undergo repeated dynamic changes in inlet conditions, such as inlet fluid temperature or mass flow rate. Capturing the dynamic characteristics of cooling coils is important to develop accurate models for controller design, fault detection and diagnostic. In light of the physical properties of air, it is highly possible that air flowing outside condenses on the tubes and fins. Condensation can severely influence convective heat transfer since both latent and sensible heats need to be

considered. Because of this complexity, little research has been done on dynamic modeling of cooling coils compared with heat exchangers.

McCullagh et al. [31] performed dynamic modeling of cooling coils assuming even tube and fin temperature distribution for each row. They used finite difference method to predict the dynamic responses of the cooling coils. Their steady state simulation results match well with experimental data for wet condition, the exception being when air mixing between rows was poor; and the dynamic responses were not validated. Clark [32] modeled the dynamics of cooling coils (four rows and above) simply by using a time constant determined by heat capacity as a function of the coil material and overall heat transfer coefficient. These simulation results match with experimental data only under dry condition. Additionally, a finite control volume method was used to model cooling coils by Chow [33]. Their results partially match with that from other cooling coil models. However, no model validation was conducted for both transient and steady state conditions.

In this dissertation, efforts are focused upon dynamic modeling of cooling coils under condensation. Finite control volume method can accurately predict dynamic response under both wet and dry conditions, since it includes more details of cooling coils.

Parameter Tuning and Model Validation

Before the application of the dynamic model, it needs to be validated to increase the model's predictive capabilities, investigate limitations, and extend its appropriateness

to handle a wide range of critical problems. Establishing accurate dynamic models is critical for control design, fault detection and diagnosis, system analysis, design and optimization. Accurate models can help with design optimization and effectively predict system behavior that may be unobtainable through experiments. The method to realize this is referred to as model validation, i.e. a process to establish the validity of the model.

Several methods are frequently used in model validation. These methods are generally divided into two categories: qualitative and quantitative [2]. Qualitative method simply means the visual norm; if the predicted outputs look similar to the data, then the model is acceptable. However, this method is inefficient in updating the model based on experimental data. Another method is quantitative, such as statistical, residual-based, parametric, and robust method. Rasmussen [2] discussed the characteristics of these methods. The method developed in this dissertation is a parametric model validation method.

Simply speaking, model validation determines whether a model is an accurate representation of the real system or not. This process is achieved by model calibration, an iterative comparison between the model and real system's behavior. Their discrepancies and the gained insights are then utilized to modify the model toward the representation of the real system. The iteration continues until the model accuracy meets the acceptable values, such as the relative error tolerance between the data and model outputs. The modification is usually achieved through parameter tuning and validation. It can provide a good understanding of the model's limitations, predictability and adaptability to a range of complex problems.

Parameter tuning and validation is an instrumental method to improve model accuracy by minimizing the difference between the experimental and predicted data and, providing the corresponding optimal parameters. However, it is a tedious and time-consuming iterative process, thus the primary challenge under these circumstances is to develop effective parameter tuning approaches that can boost increasing model predictive accuracy as well as ensuring the efficiency of model analysis.

Among those dynamic models, nonlinear models are generally more complex. Several challenges exist for parameter tuning and validation of nonlinear models. First, nonlinear models are generally computationally intense, which greatly prolongs the computation time since repeated simulations of the models are required during estimation and tuning. Furthermore, preserving the characteristics of singularly perturbed systems in these nonlinear models requires small simulation step sizes. Small step size also extends the simulation time of parameter tuning. Moreover, these models are typically complicated multiple-input-multiple-output (MIMO) systems with numerous parameters. They are frequently seen in thermal systems such as HVAC systems, aerospace systems such as jet engines, and biological systems such as cellular networks. The predicted outputs of these models usually contain various units, which brings up the interesting problem of how to properly handle them in the objective function. Additionally, for the large experimental data sets with different operating conditions, the tuned parameters obtained by one data set may not be feasible for another data set with different operating conditions.

In this dissertation, a new parameter tuning and validation approach is proposed and specifically described with a case study applied on a simple experimental air-conditioning system model, a real residential heat pump system model from Emerson Climate Technologies, and a residential conditioned space system model. This new approach is ideal in tuning physics-based MIMO models that can be linearized and discretized. It requires that a nonlinear function used to describe the model be smooth, continuous and differentiable. The discretization step should be small enough to produce an accurate model as well as avoid numerical problems. Moreover, this approach can simultaneously tune many parameters using multiple experimental data sets with different operating conditions, which is a challenge frequently seen in parameter tuning of MIMO system models. This new approach also adopts an efficient way to handle the problem caused by multiple signal domains.

In the past and current studies, manual tuning based on trial and error is mainly used in this area to minimize the difference between model prediction and experimental measurements, such as the study by Keir et al. [34]. They used the ideas of parameter sensitivity to enhance the model generation process. Sensitivity methods were specifically developed to analyze the influence of parameters on the dynamic response of a model [35]. Trajectory sensitivity functions are one of these methods [36][37]. Keir et al. [34] concluded that although parameter trajectory sensitivity functions are highly efficient in tuning their models, automatic tuning methods should be the mainstream method in the future of parameter tuning and validation study. As an iterative and tedious process of model validation itself, parameter tuning and validation is more

difficult and time-consuming to yield satisfactory results if performed manually, which is simply based on subjective engineering judgment, experiences and expert knowledge. The reason behind this lies in the fact that the tuning parameters may have contradictory or coupling effects on behavior of the model, e.g., improving the prediction accuracy of one output while decreasing the prediction accuracy of other outputs. Additionally, the process can take weeks or months.

The second problem of HVAC system model tuning and validation is that these systems usually contain numerous measurable physical parameters or immeasurable lumped parameters, posing another challenge to parameter tuning and validation of HVAC system models. In Rabehl's research, not only did they manually tune their models, but they also admitted that the number of tuning parameters is limited [38]. This problem can also be found in Helvoirt's study, in which the identification is limited for a few specified parameters [39]. Additionally, although automatic tuning is encouraged in the current research on parameter tuning and validation, the choice of algorithm for automatic tuning is difficult due to the possibility of severely prolonging computation time. In studying Energy Management Control Systems (EMCS), Nassif et al. [40] used genetic algorithms (GA) to automatically tune their component models online. EMCS refers to an optimization package installed in HVAC systems. Genetic algorithms are shown to be efficient in tuning their models, but require a large amount of computation time.

Another challenge of automatic tuning is the tedious computation as the nonlinear model needs to be simulated repeatedly. This dissertation proposes a

hierarchal method in which the local linear approximations for the nonlinear model are used. Each component model is established and linearized, and they are combined to form a complete linear system model. The outputs of the model are further predicted by using discrete-time convolution. The linear, discretized model largely reduces the computation time as it can adopt the same sample time as that of the data. This characteristic is extremely advantageous over the nonlinear model which generally requires a small step size in order to properly simulate.

CHAPTER II

DYNAMIC MODELING OF HVAC AND BUILDING SYSTEMS

This chapter discusses the dynamic modeling of HVAC systems, specifically on moving boundary and finite control volume heat exchanger models, mass flow device models, an FCV cooling coil model and a residential conditioned space model. As the simplified models, HVAC models with moving boundary heat exchanger model will be used to test and verify the proposed parameter tuning and model validation method. Moving boundary method captures the simplicity of lumped parameter models as well as the dynamics of multi-phase fluids. Despite the complexity of typical heat exchanger geometries, the MB approach assumes the heat exchanger can be modeled as one-dimensional fluid flow through a horizontal tube, with same mass, surface areas, volumes, etc.

Second, finite control volume heat exchanger models will be introduced. As mentioned in Chapter I, finite control volume method can predict the salient dynamics of multiple fluid phase heat exchangers more accurately than moving boundary method, but requires high computation cost. This method will be used to construct a heat pump model.

A cooling coil model is then presented. This model is part of Emerson heat pump model that will be validated using the proposed parameter tuning method. As previously discussed in the introduction section, condensation modeling of cooling coils is a great challenge in calculating heat transfer along the tubes. A simplified approach using a

lumped fin efficiency and effective-NTU method will be adopted to develop a cooling coil model that includes the effect of condensation.

Finally, a lumped parameter residential conditioned space model is presented. This model maintains the computational simplicity of lumped parameter methods, and is capable of predicting room temperatures, humilities, heating and cooling load throughout the year. Real weather data from typical meteorological databases will be used. All the models were implemented in MATLAB/Simulink.

Moving Boundary Heat Exchanger Models

The moving boundary evaporator model is modeled assuming two regions: a two-phase flow region and a superheat region, as is shown in Figure 2.1. Other assumptions include one-dimensional flow in a long tube, negligible axial refrigerant conduction, and negligible pressure drop. The equations for conservation of refrigerant mass, refrigerant energy and wall energy are given in Equation (2.1)~Equation(2.6). Equation (2.7) and (2.8) give the definition of heat transfer used in these conservation equations. The detailed derivations can be found in Rasmussen's Ph.D. thesis [41].

$$\frac{d\rho_1}{dP_e} A_{e,cs} L_1 \dot{P}_e + (\rho_1 - \rho_g) A_{e,cs} \dot{L}_1 = \dot{m}_{in} - \dot{m}_{int} \tag{2.1}$$

$$\frac{d\rho_2}{dP_e} A_{e,cs} L_2 \dot{P}_e + \frac{d\rho_2}{dh_{out}} A_{e,cs} L_2 \dot{h}_{out} + (\rho_g - \rho_2) A_{e,cs} \dot{L}_1 = \dot{m}_{int} - \dot{m}_{out} \tag{2.2}$$

$$(\frac{d\rho_1 h_1}{dP_e} - 1) A_{e,cs} L_1 \dot{P}_e + (\rho_1 h_1 - \rho_g h_g) A_{e,cs} \dot{L}_1 = \dot{m}_{in} h_{in} - \dot{m}_{int} h_g + q_{i,1} \tag{2.3}$$

$$(\frac{d\rho_2 h_2}{dP_e}-1)A_{e,cs}L_2\dot{P}_e+\frac{d\rho_2 h_2}{dh_{out}}A_{e,cs}L_2\dot{h}_{out}+(\rho_g h_g-\rho_2 h_2)A_{e,cs}\dot{L}_1=\dot{m}_{int}h_g-\dot{m}_{out}h_{out}+q_{i,2} \tag{2.4}$$

$$(mC_p)_w(\frac{T_{w,1}-T_{w,int}}{L_{e,total}}\dot{L}_1+\lambda_1\dot{T}_{w,1})=q_{o,1}-q_{i,1} \tag{2.5}$$

$$(mC_p)_w(\frac{T_{w,int}-T_{w,2}}{L_{e,total}}\dot{L}_1+\lambda_1\dot{T}_{w,2})=q_{o,2}-q_{i,2} \tag{2.6}$$

$$q_{i,2}=\alpha_{i,1}A_i\lambda_1(T_{w,1}-T_{r,1}) \tag{2.7}$$

$$q_{o,2}=\alpha_o A_o\lambda_2(T_a-T_{w,2}) \tag{2.8}$$

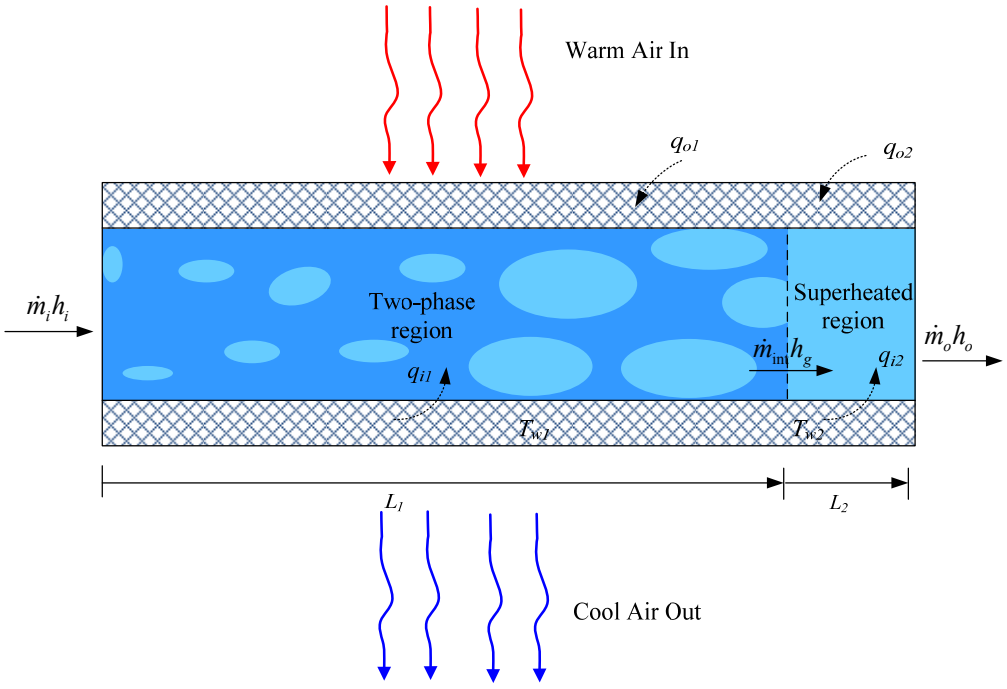


Figure 2.1: A moving boundary evaporator model with two regions.

The nonlinear model is then reorganized to the nonlinear descriptor form:

$$Z(x,u)\dot{x} = f(x,u) \tag{2.9}$$

In this form, *x* represents the dynamics in the system, *u* represents the external inputs. *f(x,u)* is the mass or energy flows that drive the dynamic responses. $Z(x,u)$ is a matrix transformation representing the easily measurable variables that facilitate modeling and simulation. For the moving boundary evaporator model obtained above, the states $x = [P_e \;\; L_1 \;\; h_{out} \;\; T_{w,1} \;\; T_{w,2}]^T$.

Similar governing equations are obtained for the condenser. Figure 2.2 shows the moving boundary condenser model with three control volumes: superheated, two-phase, and subcooled regions. Other assumptions are the same as that of the evaporator. These moving boundary models were already established in Thermo-Fluids Control Laboratory and will be used in this dissertation.

FCV Heat Exchanger Model

The finite control volume heat exchanger model is modeled by discretizing the heat exchanger model into many control volumes. Each control volume has its own physical parameters such as internal surface area, external surface area, volume, etc., and thermal properties such as pressure, enthalpy, and temperatures. The governing equations are obtained by applying energy conservation and mass conservation laws to each control volume. Thus, each control volume is treated like a lumped region. However, by increasing the number of control volumes of the heat exchanger, this

method is far more accurate and robust compared to lumped parameter method and moving boundary method.

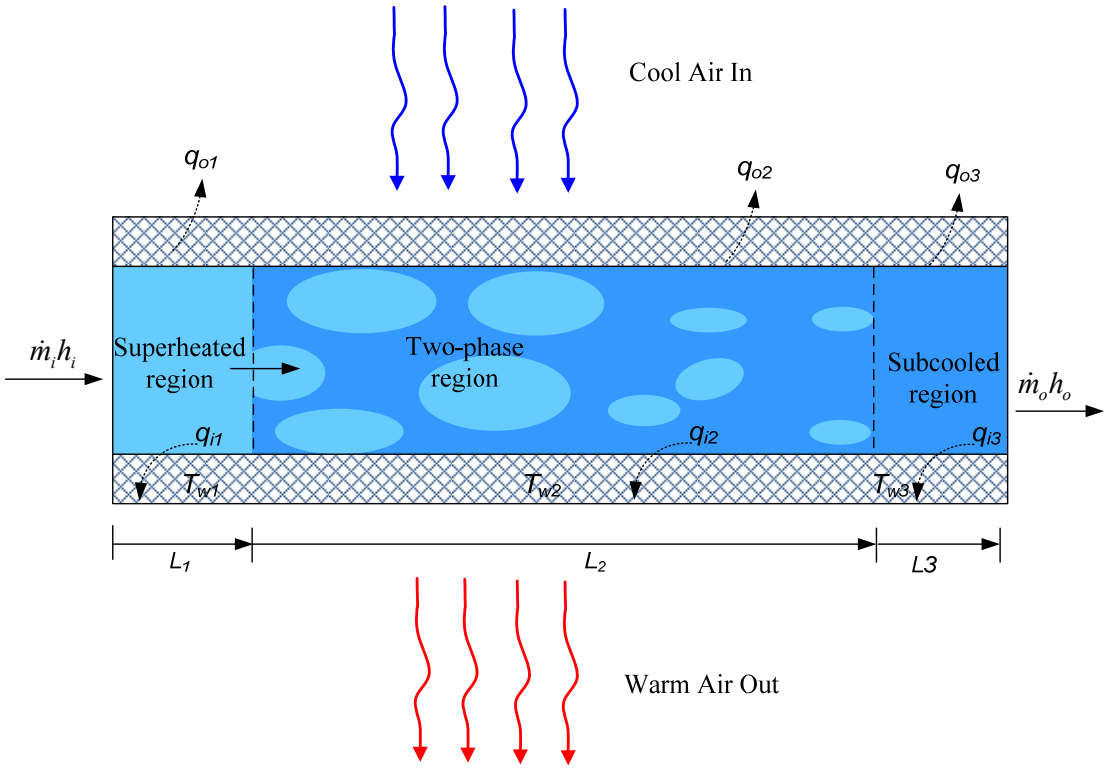


Figure 2.2: A moving boundary condenser model with three regions.

Unlike moving boundary method, where a time-varying parameter, such as mean void fraction, is assumed for the transition between two phase and single-phase region, the FCV method assumes that the fluid gradually transitions from two-phase to single-phase. The outlet enthalpy of each control volume determines the fluid state in that region. For example, if the outlet enthalpy in one control volume is equal or less than the saturated vapor enthalpy at the evaporator pressure, the state of the fluid in that region is

two-phase. If the outlet enthalpy of the fluid in a region is greater than the saturated vapor enthalpy, the state of the fluid in that region is single-phase. The control volume in which the final transition between two-phase and single-phase occurs is assumed to be in a two-phase state. The error caused by this assumption can be minimized by increasing the number of control volumes.

Figure 2.3 shows an FCV heat exchanger model. Assuming that the heat exchanger is a long, thin and horizontal tube, the refrigerant flow is one-dimensional flow, axial refrigerant conduction is negligible, and pressure drop along the heat exchanger tube due to friction and momentum change is negligible, the governing equations are derived as follows. Equations (2.10) and (2.11) are the equations for conservation of refrigerant energy. Equations (2.12) and (2.13) are the equations for conservation of mass. Equations (2.14) and (2.15) are the equations for conservation of wall energy.

Equation (2.16) and (2.17) give the definition of heat transfer used in the conservation equations. The refrigerant energy is given in Equation (2.18), and its time derivative is expanded in Equation (2.19). The refrigerant mass in a control volume is given in Equation(2.18), and its time derivative is expanded in Equation (2.21). The wall energy can be expressed in terms of the thermal capacitance and wall temperature, thus the time derivative can be expressed as in Equation (2.22).

$$\dot{U} = \dot{H}_{in} - \dot{H}_{out} + \dot{Q}_{w} \tag{2.10}$$

$$\begin{bmatrix} \dot{U}_1 \\ \vdots \\ \dot{U}_k \\ \vdots \\ \dot{U}_n \end{bmatrix} = \begin{bmatrix} \dot{m}_{in} h_{in} - \dot{m}_1 h_1 + q_{i,1} \\ \vdots \\ \dot{m}_{k-1} h_{k-1} - \dot{m}_k h_k + q_{i,k} \\ \vdots \\ \dot{m}_{n-1} h_{n-1} - \dot{m}_{out} h_n + q_{i,n} \end{bmatrix} \tag{2.11}$$

$$\begin{bmatrix} \dot{m}_1 \\ \vdots \\ \dot{m}_k \\ \vdots \\ \dot{m}_n \end{bmatrix} = \begin{bmatrix} \dot{m}_{in} - \dot{m}_1 \\ \vdots \\ \dot{m}_{k-1} - \dot{m}_k \\ \vdots \\ \dot{m}_{n-1} - \dot{m}_{out} \end{bmatrix} \tag{2.12}$$

$$\dot{m} = \dot{m}_{in} - \dot{m}_{out} \tag{2.13}$$

$$\dot{E}_w = \dot{Q}_a - \dot{Q}_w \tag{2.14}$$

$$\begin{bmatrix} \dot{E}_{w,1} \\ \vdots \\ \dot{E}_{w,k} \\ \vdots \\ \dot{E}_{w,n} \end{bmatrix} = \begin{bmatrix} q_{o,1} - q_{i,1} \\ \vdots \\ q_{o,k} - q_{i,k} \\ \vdots \\ q_{o,n} - q_{i,n} \end{bmatrix} \tag{2.15}$$

$$q_{i,k} = \alpha_{i,k} A_{i,k} (T_{w,k} - T_{r,k}) \tag{2.16}$$

$$q_{o,k} = \alpha_{o,k} A_{o,k} (T_{a,k} - T_{w,k}) \tag{2.17}$$

$$U_k = m_k u_k \tag{2.18}$$

$$\dot{U}_k = \mathrm{V}_k [(\frac{\partial \rho_k}{\partial P}\Big|_{h_k} \dot{P} + \frac{\partial \rho_k}{\partial h_k}\Big|_P \dot{h}_k) u_k + (\frac{\partial u_k}{\partial P}\Big|_{h_k} \dot{P} + \frac{\partial u_k}{\partial h_k}\Big|_P \dot{h}_k) \rho_k] \tag{2.19}$$

$$m_k = \rho_k V_k \tag{2.20}$$

$$\dot{m}_k = [\frac{\partial \rho_k}{\partial P}\Big|_{h_k} \dot{P} + \frac{\partial \rho_k}{\partial h_k}\Big|_P \dot{h}_k] V_k \tag{2.21}$$

$$\dot{E}_{w,k} = (mC_p)_w \dot{T}_{w,k} \tag{2.22}$$

The governing equations of FCV heat exchangers can also be expressed in the nonlinear descriptor form described in Equation (2.9).

The model obtained above is highly nonlinear, thus it is also linearized to facilitate parameter tuning and model validation. The linearization process is similar to that to be described in Equations (3.10) ~ (3.12), except that each control volume is linearized individually. These FCV models were already established in Thermo-Fluids Control Laboratory and will be used in this dissertation.

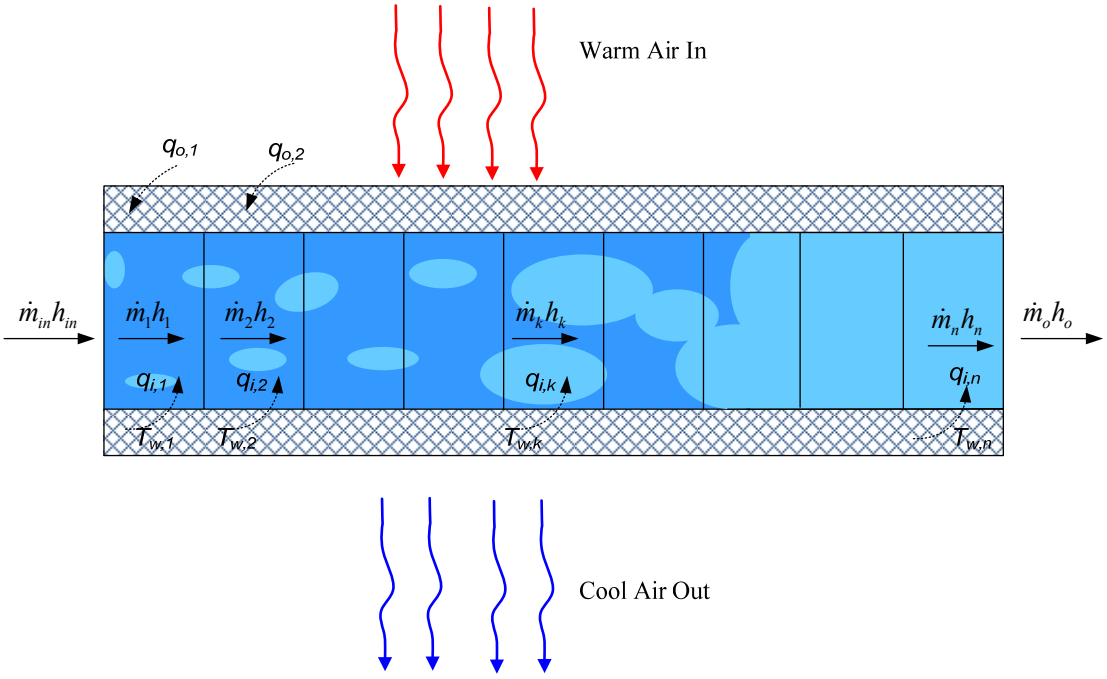


Figure 2.3: An FCV evaporator model.

Mass Flow Device Models

As described in Chapter I, dynamics in mass flow devices, i.e. compressors and expansion valves, are relatively much faster than heat exchangers. Modeling of these mass flow devices is based on static algebraic relationships. For the compressor, the modeling assumptions include two semi-empirical maps for volumetric and isentropic efficiency. It is also assumed an adiabatic process and a positive displacement system. The equation for conservation of mass is:

$$\dot{m}_v = C_d \sqrt{\rho_v (P_{v,in} - P_{v,out})} \tag{2.23}$$

Similarly for the electronic expansion valve, the modeling assumptions include a semi-empirical map for the discharge coefficient, isenthalpic process and standard orifice flow. The mass flow rate equation for conservation of mass is:

$$\dot{m}_k = \omega_k V_k \rho_k \eta_v \tag{2.24}$$

For the thermostatic expansion valve, the mass flow rate is expressed by

$$\dot{m}_v = (v_1 + v_2 (P_b - P_e)) \sqrt{\rho_v (P_c - P_e)} \tag{2.25}$$

where v_1 and v_2 are the TXV coefficients, P_b is bulb temperature, P_c is condenser temperature and P_e is evaporator temperature.

FCV Cooling Coil Model

In the cooling coils system, air flows along the fins outside the tubes and is cooled down by water flowing inside the tubes. Chilled water cooling coils often operate under either steady-state or dynamic conditions at different times. For steady-state

conditions, cooling coils are easy to model since the inlet and outlet parameters are constants. However, cooling coils often undergo changes in outdoor conditions, load requirements, set point temperatures, etc. The dynamic changes caused by these situations make it hard to capture the cooling coils' behavior. Additionally, because of the physical properties of the air, it is highly possible that air flowing outside condenses on the tubes or fins. Dehumidification may influence the convective heat transfer since both latent and sensible heats need to be considered.

Cooling coils are usually counter-cross flow finned tube heat exchangers with different arrangements. This type of geometry is very complicated and inconvenient to obtain a physical model. Thus, a simplified counter-flow arrangement model was used according to Zhou [42]. Co-flow and counter flow cooling coils were also modeled. For co-flow, air flows parallel to the water flow in the same direction. For counter-flow, air flows opposite and parallel to the water flow direction.

Model Assumptions

Model development is usually based on certain assumptions. The dynamic models developed for chilled water cooling coils include the following assumptions:

- Constant specific heats and densities for water, air and tube material
- Humid air is an ideal gas
- Water is incompressible
- Steady flows for water and air

- Negligible heat conduction in the direction of the fluid flow
- Negligible energy storage in the air
- Air-side heat transfer can be determined using the effectiveness-NTU ethod
- Lewis number for heat and mass transfer is 1 for wet condition
- The effect of condensate water on coil dynamics and overall heat capacitance is negligible under wet operating conditions
- Temperature distribution within fins in the fin height direction follows the steady-state profile, which can be calculated by the fin efficiency parameter
- The velocities of air and water are uniform
- Negligible heat conduction along the fin
- Uniform fin temperature over a control volume

Model Derivation

Generally, chilled water cooling coils have finned tubes with complicated arrangements. Figure 2.4 shows a typical four-row counter cross-flow serpentine cooling coils structure. Figure 2.5 shows one classical serpentine circuiting arrangement of a cooling coil, and describes the control volume used in the model. A single cross-flow finned tube can be considered as a basic structure of a cooling coil. As discussed before, a simplified model based on a pure counter-flow arrangement has been established. Under this assumption, the air flow direction is parallel and opposite to water flow direction.

Different types of cooling coils were modeled using finite control volume method. These models include the consideration of: (1) structure; (2) co-flow, counter flow or counter cross-flow; (3) number of rows; (4) number of tubes; (5) number of control volume in each tube.

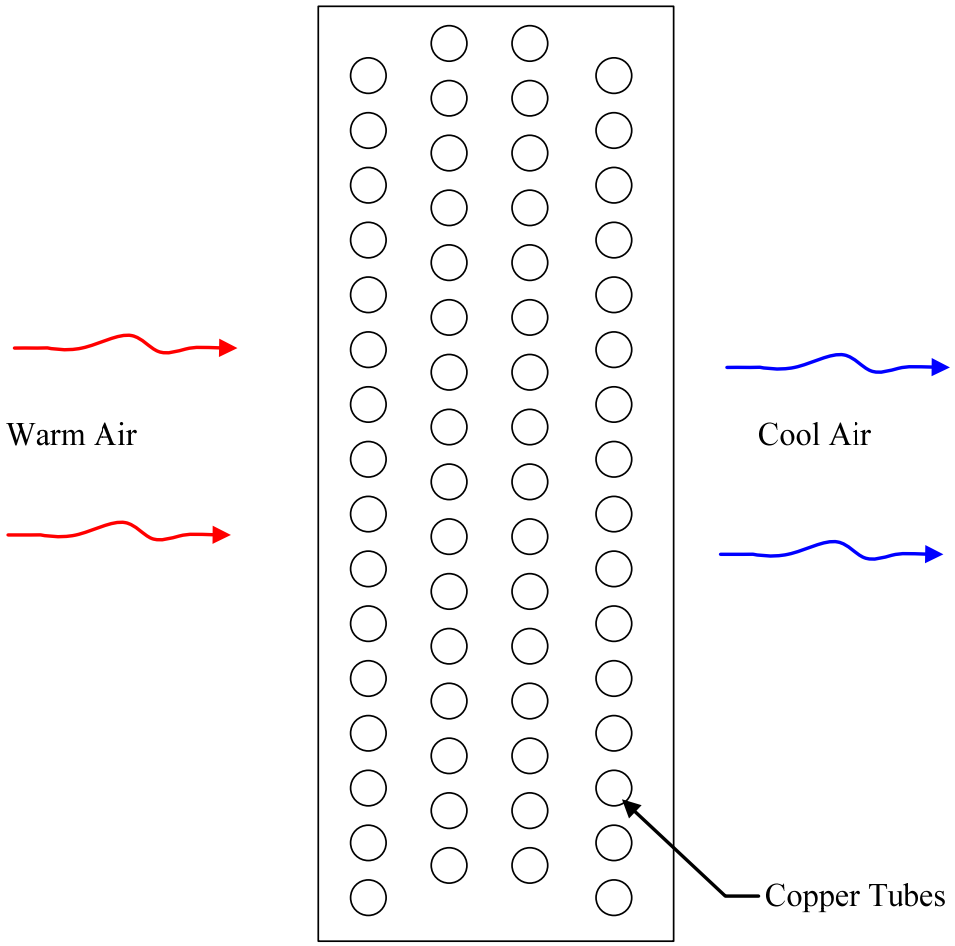


Figure 2.4: A four-row cooling coil.

The governing equations for each control volume are described as follows:

$$C_w \dot{T}_{w,j} = m_{w,in} c_{p,w} (T_{w,j-1} - T_{w,j}) - \frac{1}{R_{w,j}} (T_{w,j} - T_{c,j}) \tag{2.26}$$

$$C_c \dot{T}_{c,j} = \frac{1}{R_{a,j}} (T_{a,j} - T_{c,j}) + \frac{1}{R_{w,j}} (T_{w,j} - T_{c,j}) \tag{2.27}$$

$$C_c \dot{T}_{c,j} = \frac{1}{R^*_{a,j}} (h_{a,j} - h_{s,c,j}) + \frac{1}{R_{w,j}} (T_{w,j} - T_{c,j}) \quad . \tag{2.28}$$

Equation (2.26) represents conservation of energy for water while Equation (2.27) and (2.28) represent coil energy conservation under dry and wet condition, respectively.

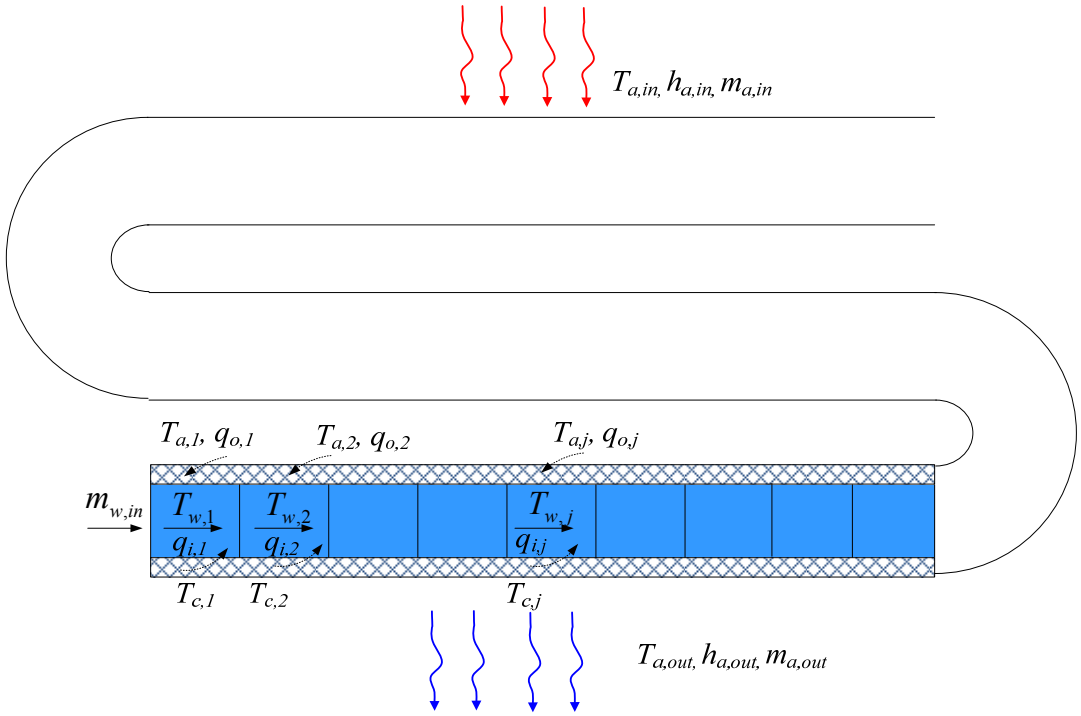


Figure 2.5: An FCV cooling coil schematic.

For dry conditions, the governing equations for each region can be combined and organized into the nonlinear descriptor form.

$$\begin{bmatrix} C_w\dot{T}_{w,1} \\ \vdots \\ C_w\dot{T}_{w,j} \\ \vdots \\ C_w\dot{T}_{w,n} \\ C_c\dot{T}_{c,1} \\ \vdots \\ C_c\dot{T}_{c,j} \\ \vdots \\ C_c\dot{T}_{c,n} \end{bmatrix} = \begin{bmatrix} Z_{11} & 0 \\ 0 & Z_{22} \end{bmatrix} \begin{bmatrix} \dot{T}_{w,1} \\ \vdots \\ \dot{T}_{w,j} \\ \vdots \\ \dot{T}_{w,n} \\ \dot{T}_{c,1} \\ \vdots \\ \dot{T}_{c,j} \\ \vdots \\ \dot{T}_{c,n} \end{bmatrix} = \begin{bmatrix} m_{w,in}c_{p,w}(T_{w,in} - T_{w,1}) - (T_{w,1} - T_{c,1}) / R_{w,1} \\ \vdots \\ m_{w,in}c_{p,w}(T_{w,j-1} - T_{w,j}) - (T_{w,j} - T_{c,j}) / R_{w,j} \\ \vdots \\ m_{w,in}c_{p,w}(T_{w,n-1} - T_{w,n}) - (T_{w,n} - T_{c,n}) / R_{w,n} \\ (T_{a,1} - T_{c,1})R_{a,1} + (T_{w,1} - T_{c,1}) / R_{w,1} \\ \vdots \\ (T_{a,j} - T_{c,j}) / R_{a,j} + (T_{w,j} - T_{c,j}) / R_{w,j} \\ \vdots \\ (T_{a,n} - T_{c,n}) / R_{a,n} + (T_{w,n} - T_{c,n}) / R_{w,n} \end{bmatrix} \tag{2.29}$$

For wet conditions, similar equations are obtained:

$$\begin{bmatrix} C_w\dot{T}_{w,1} \\ \vdots \\ C_w\dot{T}_{w,j} \\ \vdots \\ C_w\dot{T}_{w,n} \\ C_c\dot{T}_{c,1} \\ \vdots \\ C_c\dot{T}_{c,j} \\ \vdots \\ C_c\dot{T}_{c,n} \end{bmatrix} = \begin{bmatrix} Z_{11} & 0 \\ 0 & Z_{22} \end{bmatrix} \begin{bmatrix} \dot{T}_{w,1} \\ \vdots \\ \dot{T}_{w,j} \\ \vdots \\ \dot{T}_{w,n} \\ \dot{T}_{c,1} \\ \vdots \\ \dot{T}_{c,j} \\ \vdots \\ \dot{T}_{c,n} \end{bmatrix} = \begin{bmatrix} m_{w,in}c_{p,w}(T_{w,in} - T_{w,1}) - (T_{w,1} - T_{c,1}) / R_{w,1} \\ \vdots \\ m_{w,in}c_{p,w}(T_{w,j-1} - T_{w,j}) - (T_{w,j} - T_{c,j}) / R_{w,j} \\ \vdots \\ m_{w,in}c_{p,w}(T_{w,n-1} - T_{w,n}) - (T_{w,n} - T_{c,n}) / R_{w,n} \\ (h_{a,1} - h_{s,c,1}) / R^*_{a,1} + (T_{w,1} - T_{c,1}) / R_{w,1} \\ \vdots \\ (h_{a,j} - h_{s,c,j}) / R^*_{a,j} + (T_{w,j} - T_{c,j}) / R_{w,j} \\ \vdots \\ (h_{a,n} - h_{s,c,n}) / R^*_{a,n} + (T_{w,n} - T_{c,n}) / R_{w,n} \end{bmatrix} \tag{2.30}$$

The water-side thermal resistance can be calculated as

$$R_{w,j} = \frac{N}{\alpha_{w,j} A_{w,tot}} \tag{2.31}$$

The air-side thermal resistance under dry condition is calculated using the effectiveness-NTU method:

$$R_{a,j} = \frac{1}{\varepsilon_{a,j} m_{a,in} c_{p,a}} \tag{2.32}$$

$$\varepsilon_{a,j} = 1 - e^{-NTU_{a,j}} \tag{2.33}$$

$$NTU_{a,j} = \frac{\eta_{a,j} \alpha_{a,j} A_{a,tot}}{m_{a,in} c_{p,a} N} \tag{2.34}$$

The outlet air temperature for the current region, which is also the local inlet air temperature for the corresponding control volume on the next row, is given by

$$T_{a,out} = T_{a,j} + \varepsilon_a (T_{c,j} - T_{a,j}) \tag{2.35}$$

The air-side thermal resistance under wet condition is also calculated by effectiveness-NTU method.

$$R^*_{a,j} = \frac{1}{\varepsilon^*_{a,j} m_{a,in}} \tag{2.36}$$

$$\varepsilon^*_{a,j} = 1 - e^{-NTU^*_{a,j}} \tag{2.37}$$

$$NTU^*_{a.j} = \frac{\eta^*_{a.j} \alpha^*_{a,j} A_{a,tot}}{m_{a,in} c_{p,a} N} \tag{2.38}$$

The outlet air enthalpy, which is also the local inlet air enthalpy for the corresponding control volume on the next row, is given by

$$h_{a,out} = h_{a,j} + \varepsilon^*_{a,j} (h_{s,c,j} - h_{a,j}) \tag{2.39}$$

The outlet air humidity, which is also the local inlet air humidity for the corresponding control volume on the next row, is given by [43]

$$\omega_{out} = \omega_{s,j} + (\omega_i - \omega_{s,j}) \exp(-\frac{\alpha^*_{a,j} A_{a,tot} \varepsilon^*_{a,j}}{m_{a,in} c_{p,a}}) \tag{2.40}$$

where $\omega_{s,j}$ is calculated by

$$\omega_{s,j} = 0.622 \frac{P_{g,j}}{P_a - P_{g,j}} \tag{2.41}$$

where $P_{g,j}$ is the saturation vapor pressure and is given by

$$P_{g,j} = 6.11 \cdot 10^{\frac{7.5T_{c,j}}{237.7+T_{c,j}}} \tag{2.42}$$

The relative humidity is calculated by

$$\phi = \frac{\omega P_a}{(0.622 + \omega) P_g} \tag{2.43}$$

The air enthalpies can be approximated by

$$h_{a,j} = c_{p,a} T_{a.j} + \omega_i (h_{v,0} + c_{p,v} T_{a,j}) \tag{2.44}$$

It is also used to calculate the outlet air temperature under wet condition.

The dew point temperature is calculated by

$$T_{dp} = \frac{-430.22 + 237.7 * \ln(10\phi P_{g,j})}{-\ln(10\phi P_{g,j}) + 19.08} \tag{2.45}$$

Dry Initial Condition

This section presents some verification results when the initial steady state of the cooling coils is under dry condition. Figure 2.6 shows the simulation results when the inlet water mass flow decreases from 0.5 kg/s to 0.3 kg/s at t = 5 s.

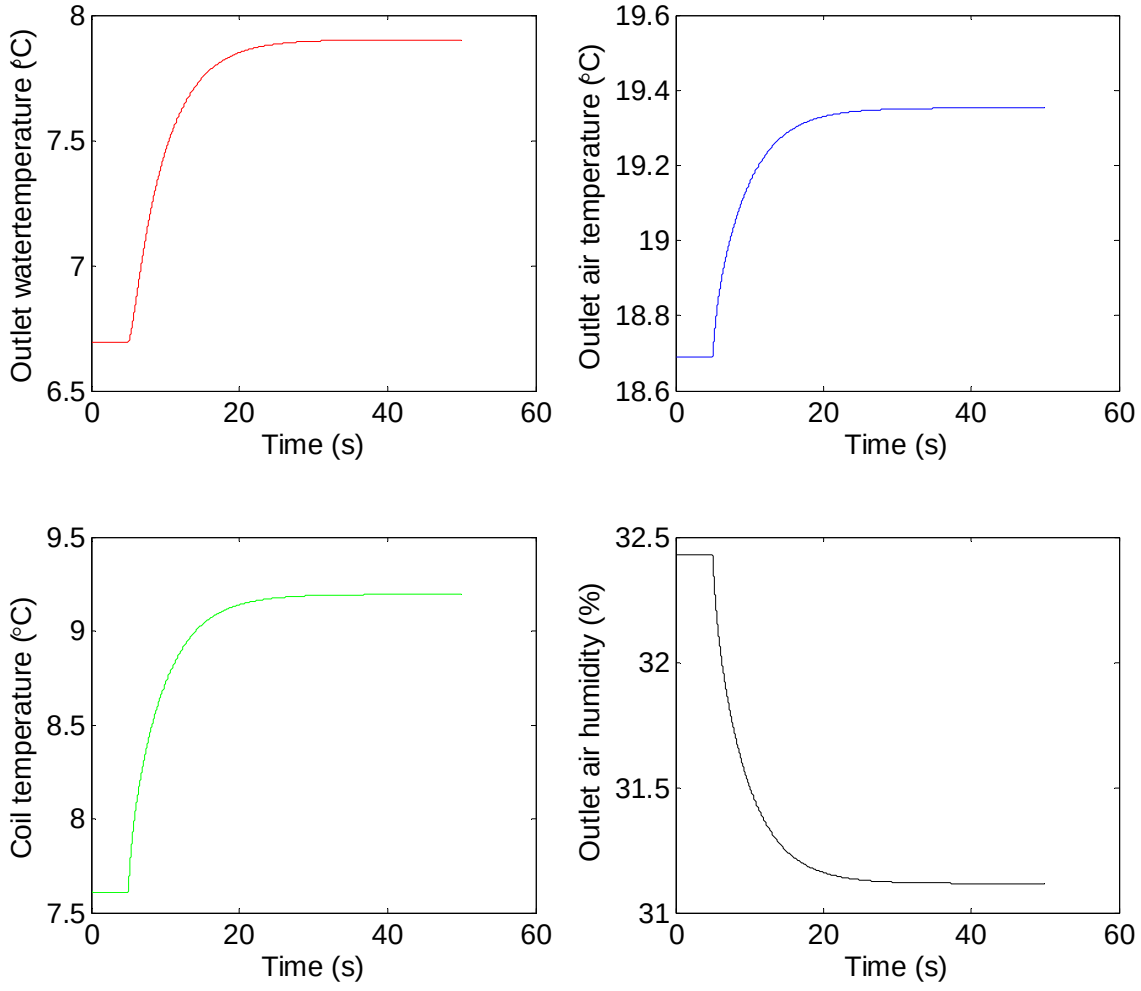


Figure 2.6: Step change of inlet water mass flow rates (dry initial condition).

As the water mass flow rate decreases and other operating conditions do not change, less heat can be transferred from air to water, thus the outlet air temperature increases. The outlet water temperature also increases because more heat is provided to less water. This also explains the coil temperature change. The outlet relative humidity decreases because air temperature increases and there is no condensation.

Figure 2.7 shows the simulation results when the inlet air mass flow increases from 0.6 kg/s to 1.2 kg/s at t=10 s. As the air mass flow rate increases, more air needs to be cooled down. Since the operating conditions of water do not change, the outlet air temperature increases. The outlet water temperature also increases because more air increases the heat transfer. This also explains the coil temperature change. The outlet relative humidity decreases because air temperature increases and there is no condensation.

Figure 2.8 shows the simulation results when the inlet air humidity increases from 20% to 60% at t=5s. As the air humidity increases, condensation occurs and the heat transfer coefficient increases, thus more heat energy can be transferred to the water. Since the operating conditions of water do not change, the outlet air temperature decreases. The outlet water temperature increases due to the increased heat transfer. This also explains the coil temperature change. The outlet relative humidity increases to 100% because of the condensation.

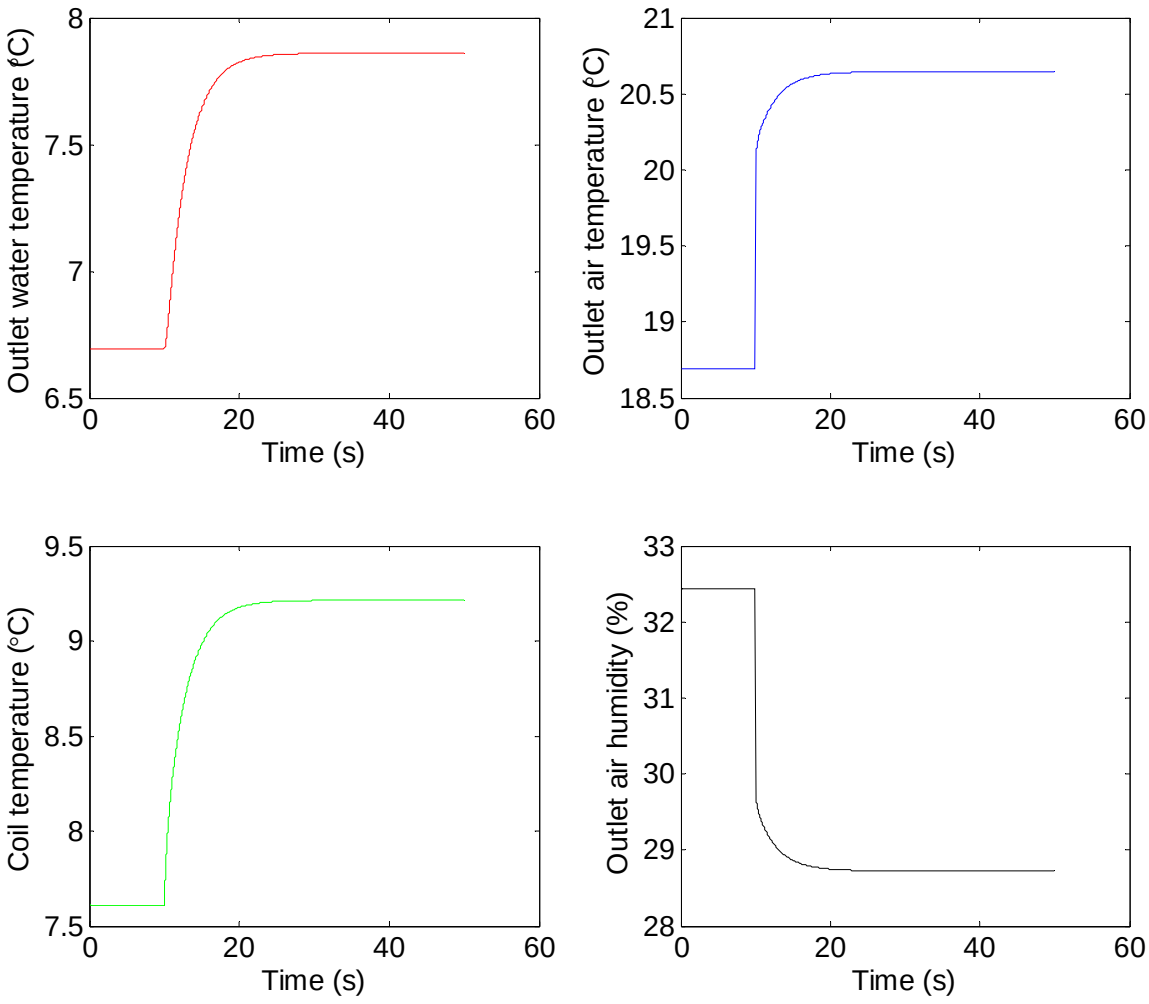


Figure 2.7: Step change of inlet air mass flow rates (dry initial condition).

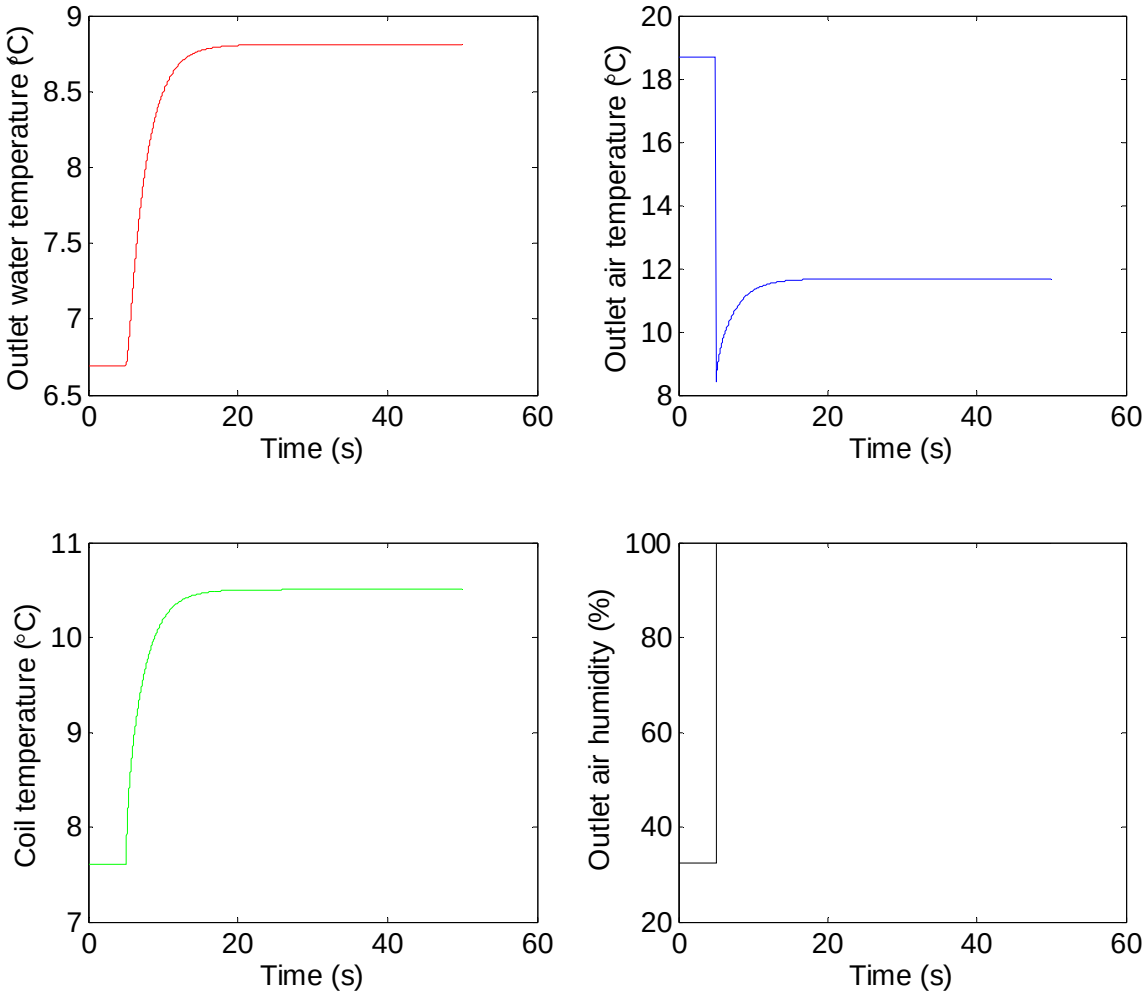


Figure 2.8: Step change of inlet air humidity (dry initial condition).

Wet Initial Condition

This section presents some verification results when the initial steady state of the cooling coils is under wet condition. Figure 2.9 shows the simulation results when the inlet water mass flow rate decreases from 0.5 kg/s to 0.3 kg/s at t=5s. These figures show a similar trend to that of the dry condition. As the water mass flow rate decreases and other operating conditions do not change, less heat can be transferred from air to water,

thus the outlet air temperature increases. The outlet water temperature also increases because more heat is provided to less water. This also explains the coil temperature change. The outlet relative humidity is 100% because condensation occurs.

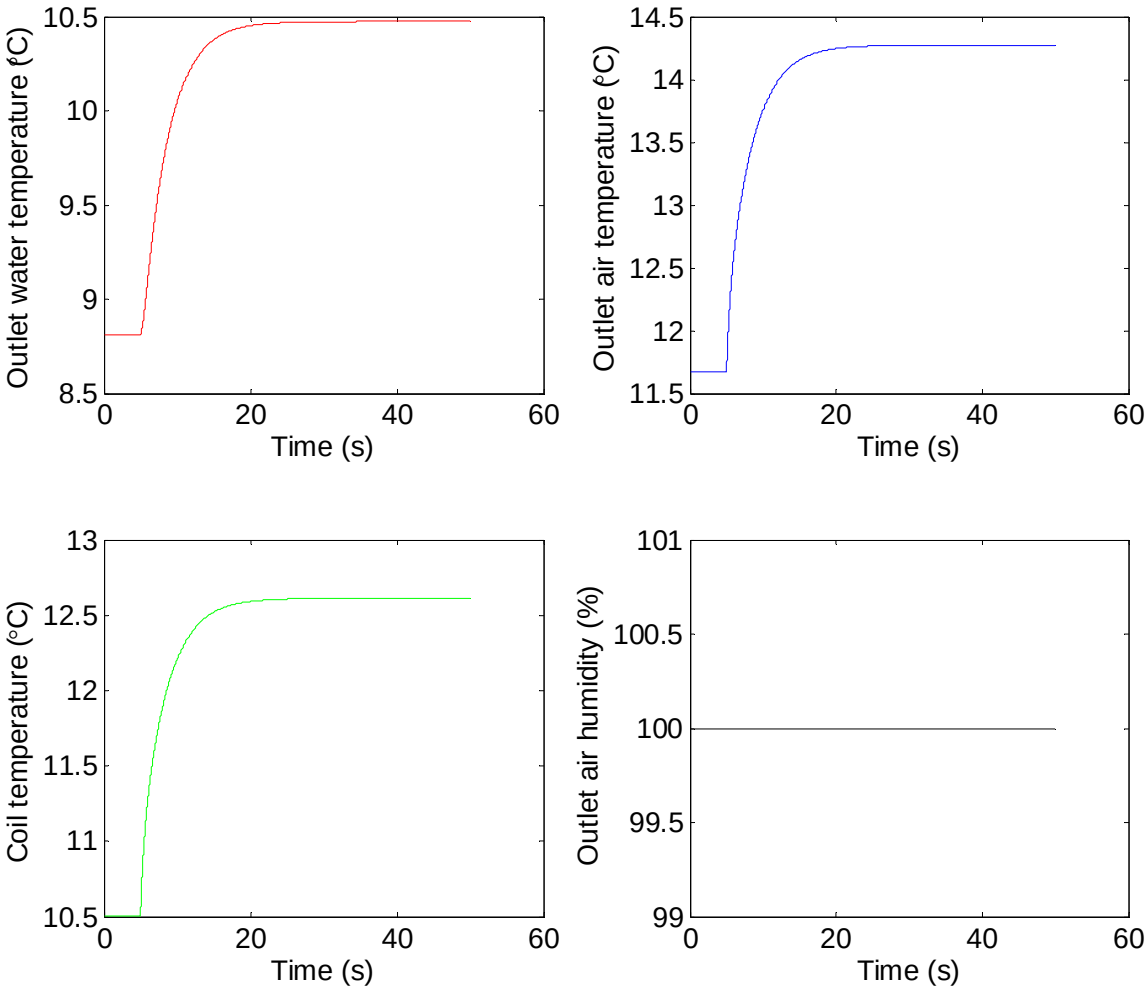


Figure 2.9: Step change of inlet water mass flow (wet initial condition).

Figure 2.10 shows the simulation results when the inlet air mass flow increases from 0.6 kg/s to 1.2 kg/s at t=10s. These figures show a similar trend to that of the dry

condition. As the air mass flow rate increases, more air needs to be cooled down. Since the operating conditions of water do not change, the outlet air temperature increases. The outlet water temperature also increases because more air increases the heat transfer. This also explains the coil temperature change. The outlet relative humidity is 100% because condensation occurs.

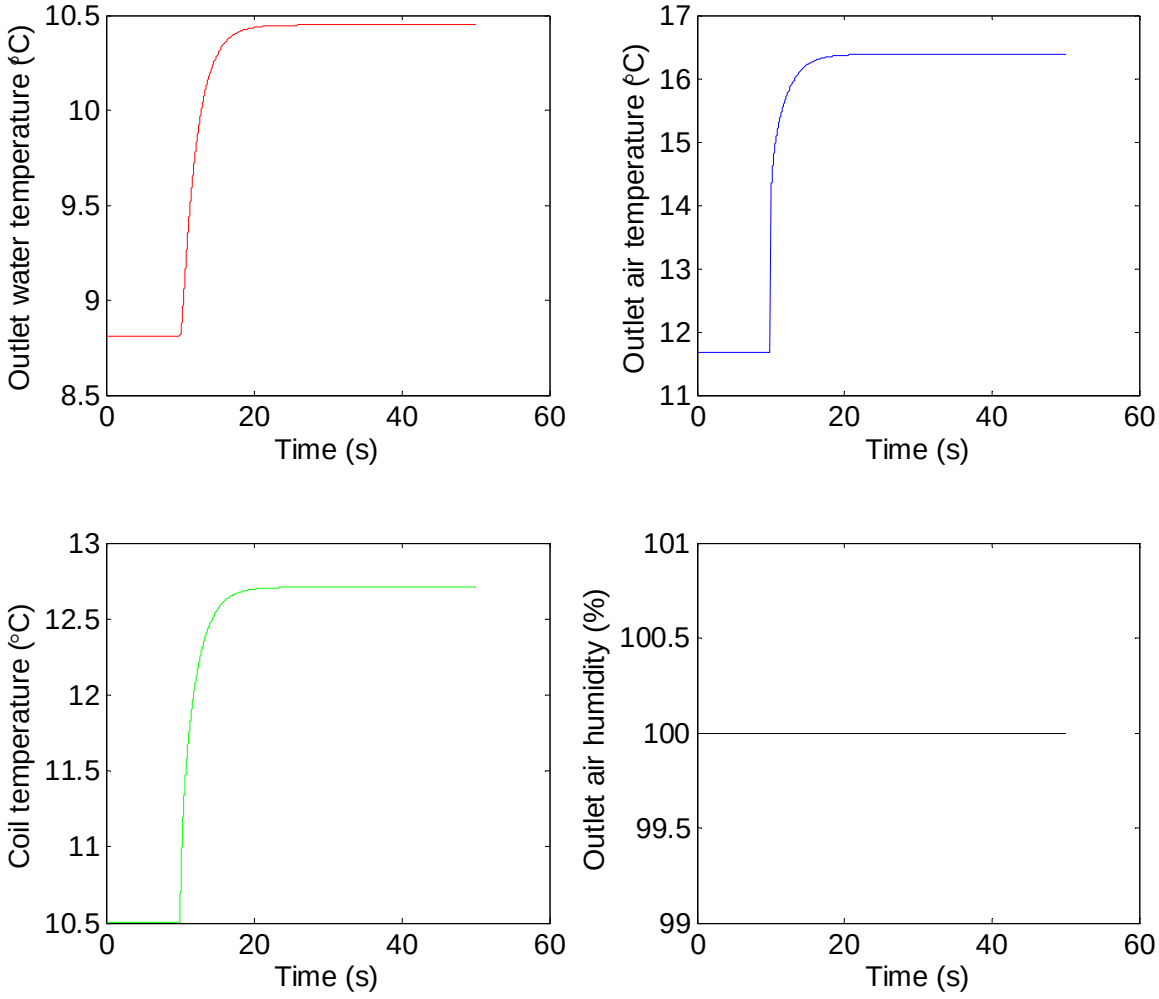


Figure 2.10: Step change of inlet air flow rates (wet initial condition).

Figure 2.11 shows the simulation results when the inlet air humidity decreases from 60% to 20% at t=5s. As the humidity decreases, condensation disappears and the heat transfer coefficient decreases, thus less heat can be transferred to the water. Since the operating conditions of water do not change, the outlet air temperature increases. The outlet water temperature decreases due to the decreased heat transfer. This also explains the coil temperature change. The outlet relative humidity decreases to 32.4%.

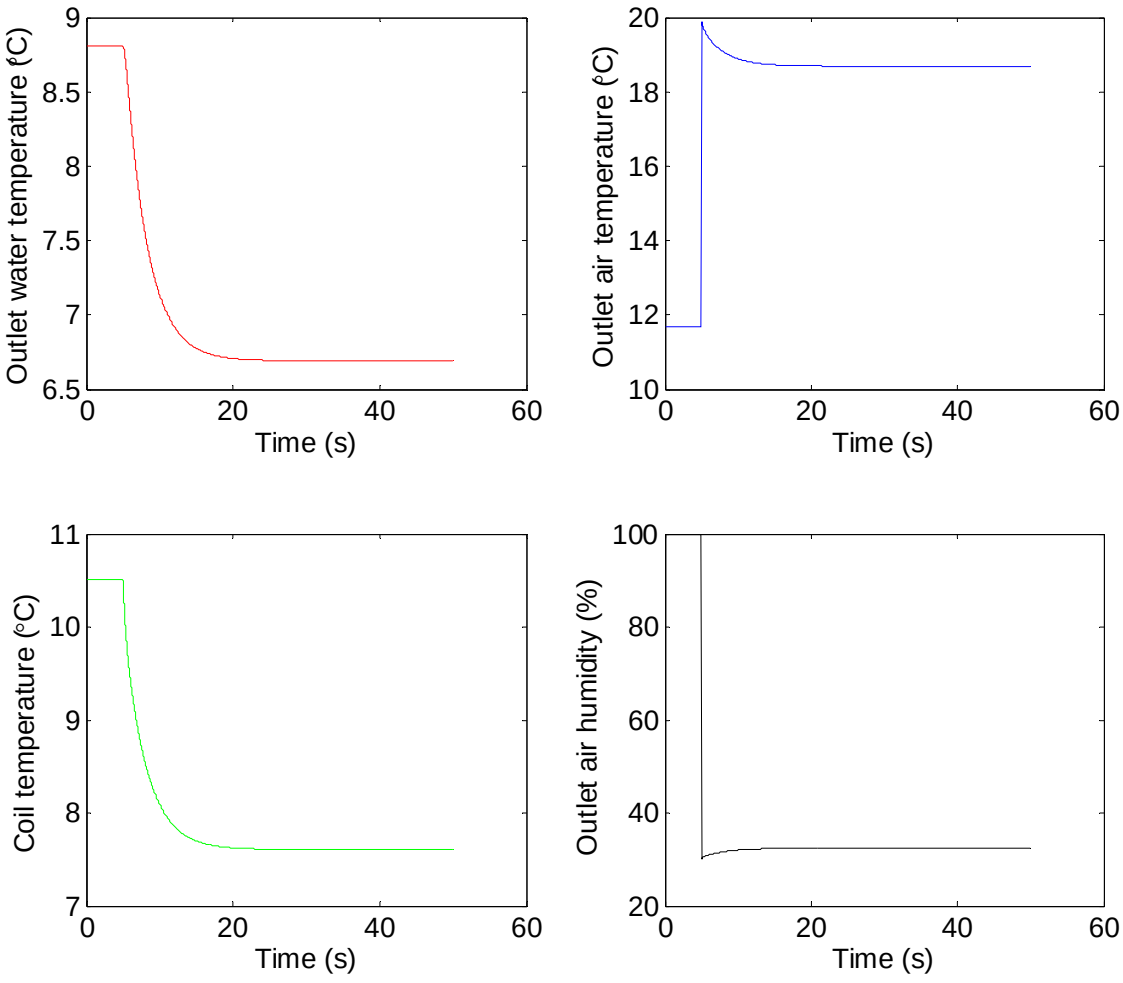


Figure 2.11: Step change of inlet air humidity (wet initial condition).

Residential Conditioned Space Model

The modeling approach of residential house aims to predict seasonal and daily energy use, as well as other parameters, while maintaining computational simplicity of lumped parameter models. This model includes a simplified representation of seasonal cooling and heating loads on houses, including yearly weather data from typical meteorological databases. This approach assumes a sealed room with closed doors and windows, and the air temperature and humidity are uniformly distributed in the entire room. The boundaries between the wall and indoor air, the ceiling and indoor air, the wall and the outdoor air, the indoor air and the floor, are also assumed. This model will be integrated with the finite control volume variable speed heat pump model developed for Emerson.

Model Assumptions

Figure 2.12 shows a simplified residential conditioned space model. The conservation of energy equations for indoor air, wall and ceiling can be applied and the governing equations can be derived. The derivation approach uses several modeling assumptions. These assumptions have been commonly used in past modeling efforts and are stated below:

- Air is an ideal gas with a constant specific heat.
- Room water vapor is an ideal gas.
- The room is assumed to be a cuboid with closed doors and windows.

- The floor/soil temperature is a constant.
- The room temperature and humidity is evenly distributed.

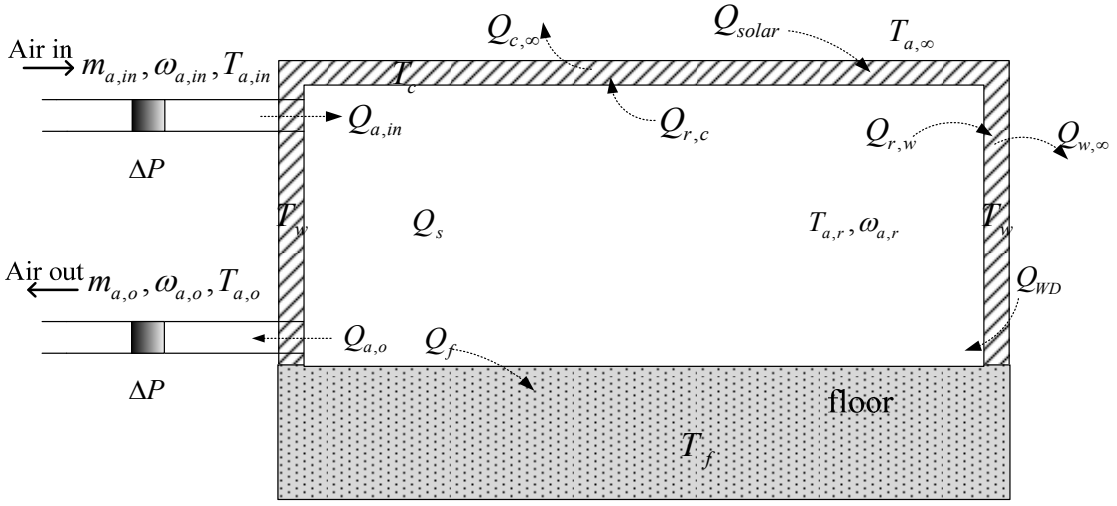


Figure 2.12: Residential Conditioned Space Model.

- The wall is lumped into a single layer; its temperature is evenly distributed, but can change with time.
- The ceiling is lumped into one layer; its temperature is evenly distributed, but can change with time.
- The heat exchange of the room with the environment includes: (1) heat convection between the wall and the outdoor air; (2) heat convection between the floor and the room air; (3) fenestration energy flow, consisting of energy flow caused by indoor-outdoor temperature difference (thermal energy flow), and instantaneous energy flow caused by solar radiation (solar energy flow);

(4) heat convection between the ceiling and the outdoor air;(5) solar heat gain through the ceiling.

- The internal heat source of the room includes the heat/cooling supplied by the heat pump, occupancy of people and equipment such as lighting and computers.
- The boundary conditions for this room include: heat convection between (1) the wall and the room air; (2) the floor and the indoor air.
- The pressure drop along the duct is a function of the air mass flow rate.
- There is no air temperature drop along the duct.

Model Derivation

The conservation equations of room air mass, water vapor mass, room air energy, wall energy and ceiling energy are derived as follows:

Conservation of room air mass

$$\frac{dm_{air}}{dt} = \dot{m}_{a,SA} + \dot{m}_{a,inf} - \dot{m}_{a,RA} - \dot{m}_{a,exf} \tag{2.46}$$

Conservation of room water vapor mass

$$\frac{dm_{v}}{dt} = \dot{m}_{v,SA} + \dot{m}_{v,inf} - \dot{m}_{v,RA} - \dot{m}_{v,exf} + \dot{m}_{v,s} \tag{2.47}$$

which can be expanded as

$$\frac{d(\omega_{a,r} m_{air})}{dt} = \omega_{a,SA}\dot{m}_{a,SA} + \omega_{a,OA}\dot{m}_{a,inf} - \omega_{a,RA}\dot{m}_{a,RA} - \omega_{a,RA}\dot{m}_{a,exf} + \dot{m}_{v,s} \tag{2.48}$$

Conservation of room air energy

$$\frac{d(m_v u_v)}{dt} + \frac{c_{v,air} d(m_{air} T_{a,r})}{dt} = Q_s + Q_{a,SA} + Q_{a,inf} - Q_{r,w} - Q_{r,c} + Q_{WD} - Q_{a,RA} - Q_{a,exf} - Q_f \quad (2.49)$$

where

$$Q_{a,SA} = \dot{m}_{a,SA} h_{a,SA} \quad (2.50)$$

$$Q_{a,RA} = \dot{m}_{a,RA} h_{a,RA} \quad (2.51)$$

$$Q_{a,inf} = \dot{m}_{a,inf} h_{a,inf} \quad (2.52)$$

$$Q_{a,exf} = \dot{m}_{a,exf} h_{a,exf} \quad (2.53)$$

$$Q_{r,w} = U_{r,w} A_w (T_{a.r} - T_w) \quad (2.54)$$

$$Q_{r,c} = U_{r,c} A_c (T_{a.r} - T_c) \quad (2.55)$$

$$Q_{WD} = U_{WD} A_{pf} (T_{a,OA} - T_{a,r}) + (\mathrm{SHGC}) A_{pf} E_t \quad (2.56)$$

$$Q_f = \alpha_{r,f} A_f (T_{a,r} - T_f) \quad (2.57)$$

$$u_v = u_{v,0} + c_{v,v} T_{a,r} \quad (2.58)$$

$$A_{pf} = A_{WD} \cos(\beta) \quad (2.59)$$

Conservation of wall energy

$$\frac{c_{p,w} m_w dT_w}{dt} = Q_{r,w} - Q_{w,OA} \quad (2.60)$$

where

$$Q_{w,OA} = U_{w,OA} A_w (T_w - T_{a,OA}) \quad (2.61)$$

Conservation of ceiling energy

$$\frac{c_{p,c} m_c dT_c}{dt} = Q_{r,c} + Q_{solar} - Q_{c,OA} \quad (2.62)$$

where

$$Q_{c,OA} = U_{c,OA} A_c (T_c - T_{a,OA}) \tag{2.63}$$

The enthalpy of the humid air is calculated by:

$$h_{a,SA} = c_{p,air} T_{a,SA} + \omega_{a,SA} (h_{v,0} + c_{p,v} T_{a,SA}) \tag{2.64}$$

$$h_{a,RA} = c_{p,air} T_{a,RA} + \omega_{a,RA} (h_{v,0} + c_{p,v} T_{a,RA}) \tag{2.65}$$

$$h_{a,inf} = c_{p,air} T_{a,OA} + \omega_{a,OA} (h_{v,0} + c_{p,v} T_{a,OA}) \tag{2.66}$$

$$h_{a,exf} = h_{a,RA} \tag{2.67}$$

Calculations of mixture air:

$$\dot{m}_{a,OA} = \dot{m}_{a,RA} - \dot{m}_{a,CA} \tag{2.68}$$

$$\omega_{a,MA} \dot{m}_{a,MA} = \dot{m}_{a,CA} \omega_{a,RA} + \dot{m}_{a,OA} \omega_{a,OA} \tag{2.69}$$

$$h_{a,MA} \dot{m}_{a,MA} = \dot{m}_{a,CA} h_{a,RA} + \dot{m}_{a,OA} h_{a,OA} \tag{2.70}$$

$$h_{a,MA} = c_{p,air} T_{a,MA} + \omega_{a,MA} (h_{v,0} + c_{p,v} T_{a,MA}) \tag{2.71}$$

where

$$\dot{m}_{a,MA} = \dot{m}_{a,SA} \tag{2.72}$$

The overall heat transfer coefficient is expressed by:

$$\frac{1}{UA} = \Sigma \frac{1}{\alpha A} + \Sigma R \tag{2.73}$$

where R is the thermal resistance and is expressed by

$$R = \frac{x}{kA} \tag{2.74}$$

The correlations of the air humidity are listed in the following equations:

$$P_g = 6.11 \cdot 10^{(\frac{7.5T_{a,r}}{237.7+T_{a,r}})} \tag{2.75}$$

$$P_v = \phi P_g \tag{2.76}$$

$$\phi = \frac{\omega P_a}{(0.622+\omega)P_g} \tag{2.77}$$

$$\omega = \frac{0.622P_v}{P_a - P_v} \tag{2.78}$$

The temperature difference between the actual room temperature and the thermostat temperature set point is:

$$T_d = T_{a,r} - T_{sp} \tag{2.79}$$

The return air temperature is:

$$T_{a,RA} = T_{a,r} \tag{2.80}$$

The air humidity of return air is:

$$\varphi_{a,RA} = \varphi_{a,r} \tag{2.81}$$

Model Verification

To illustrate the capabilities of the simulation model, several dynamic simulation studies were performed. The figures present the results of each of some open loop an close loop simulation studies.

Open Loop

In Figure 2.13, the input change is the step changes of the supply and return air mass flow rates, shown in the first figure. As the supply and return air mass flow rate suddenly increase from 0.5 kg/s to 0.7 kg/s, the room temperature gradually increases, as more heat is transferred to the room. The heating load suddenly increases due to the increase of supply air mass flow rates, but it then decreases because the room temperature gradually increases. When the supply and return air mass flow rates return to the initial value, all the corresponding outputs return to the initial steady state condition.

Figure 2.14 shows step responses of supply air humidity, shown in the first figure. As the supply air humidity suddenly increase from 20.5% to 40%, the room temperature gradually increases. This is due to the increase of total inlet air enthalpy, since it consists of water vapor (increases due to the input) and dry air enthalpy. As the room temperature increases, the heating load decreases accordingly. All the outputs reach a new steady state at about 40 sec.

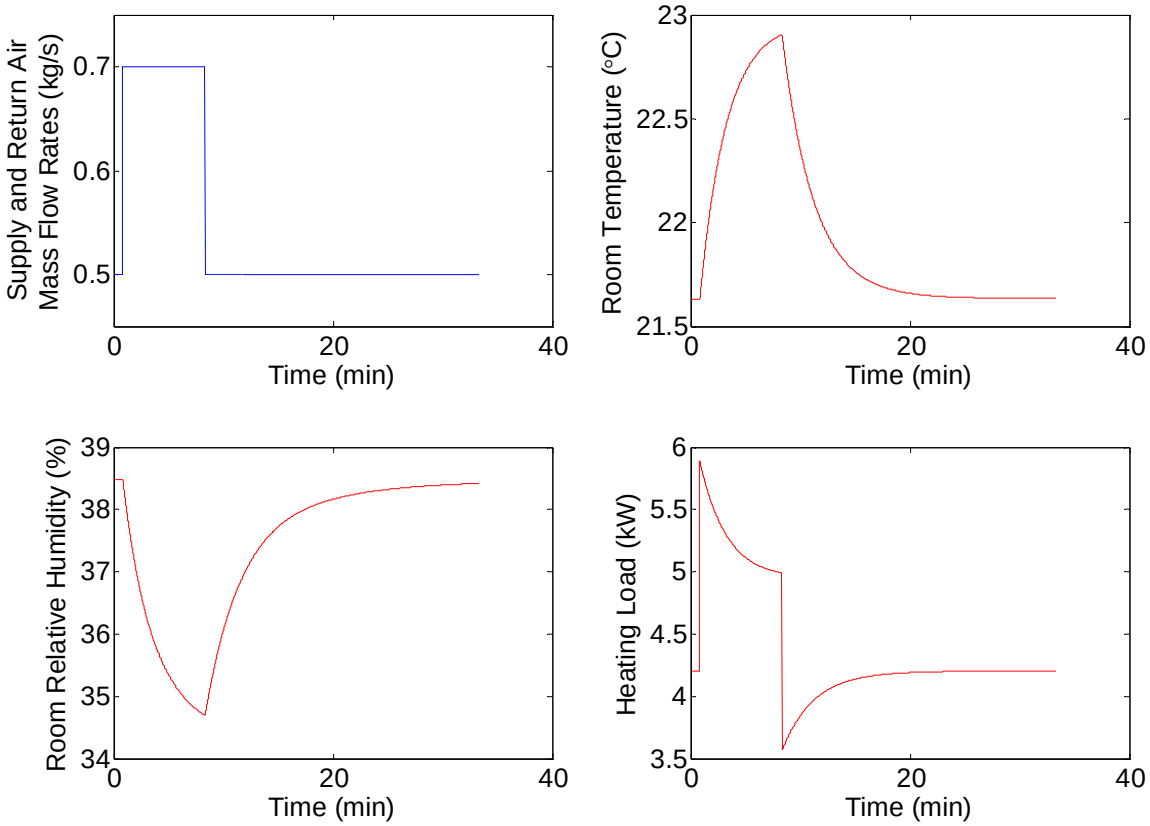


Figure 2.13: Step response of supply and return air mass flow rate.

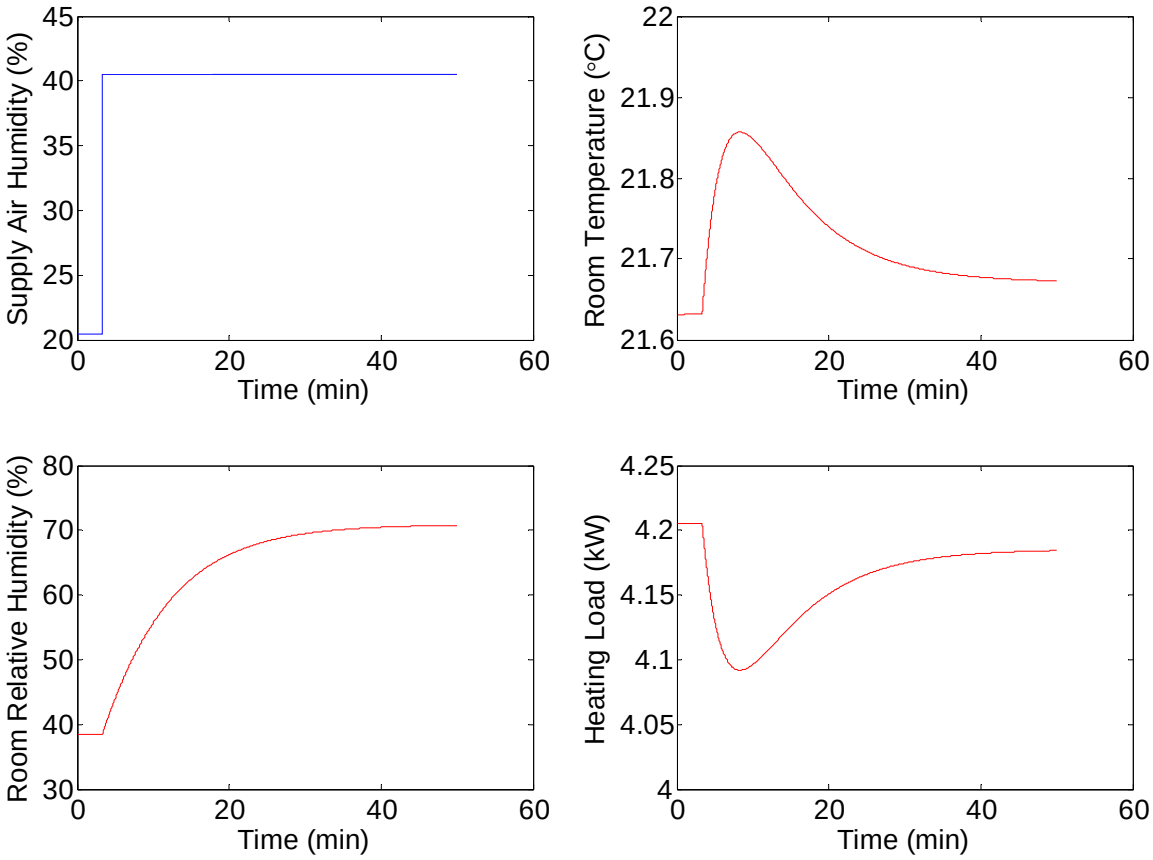


Figure 2.14: Step response of supply air humidity.

In Figure 2.15, the input change is the step changes of outside air temperature, shown in the first figure. As the outside air temperature suddenly decreases from 0 ºC to -10 ºC, the room temperature gradually decreases, as more heat is lost from the room to the outside. The heating load, which is proportional to the positive temperature difference between the supply air and room air, thus increases. The room relative humidity increases because the decrease of room temperature.

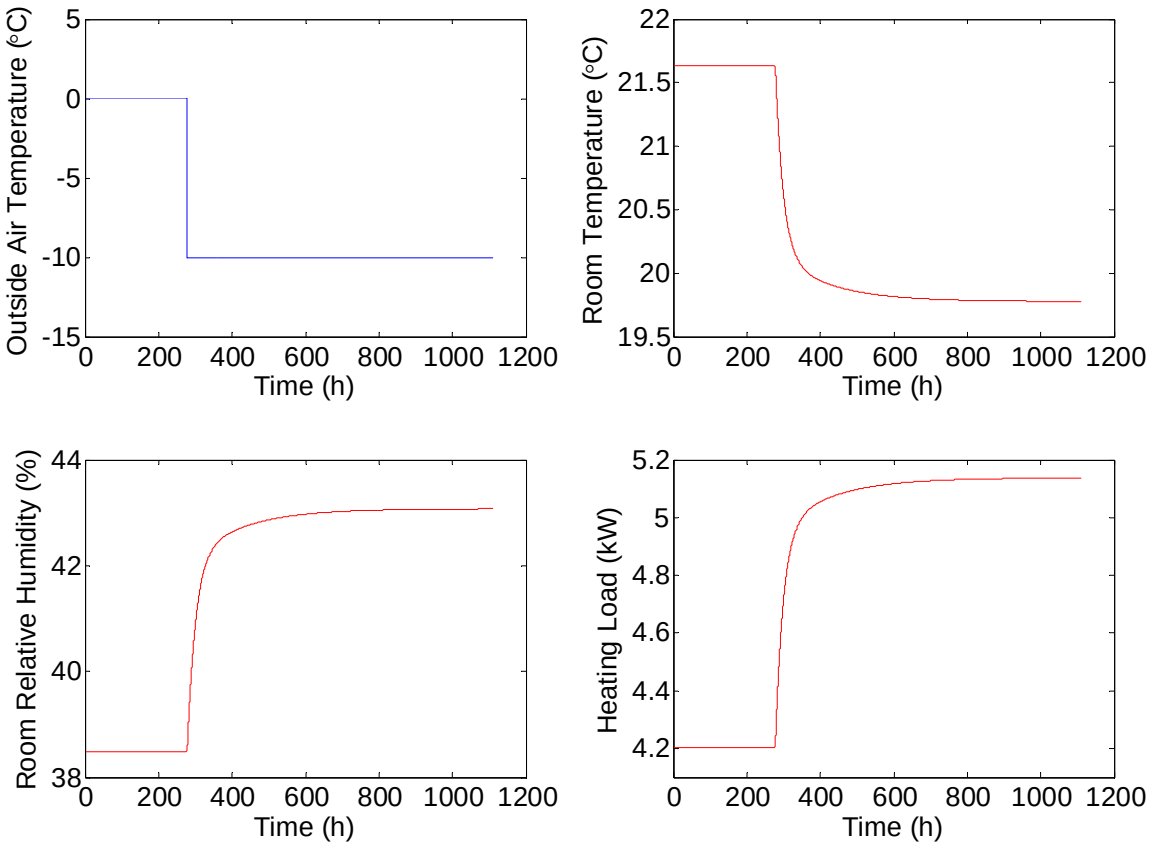


Figure 2.15: Step response of outside temperature.

Closed Loop

In Figure 2.16, the setpoint room temperature increases from 20 ºC to 26 ºC. The controlled inputs are supply air temperature and humidity. The controller is a PI controller, and it eventually drives the room temperature to 26 ºC. The inputs at the beginning of the simulation change because the initial steady state room temperature and humidity are not at the setpoint values, and the PI controller regulates the controlled inputs, i.e. supply air temperature and humidity, to reach the setpoint values. The third figure shows the setpoint temperature and room temperature. After the setpoint

temperature changes, the controller tries to match the room temperature with the new setpoint temperature. Other outputs, such as room humidity and heating load, change with this and reach a new steady state.

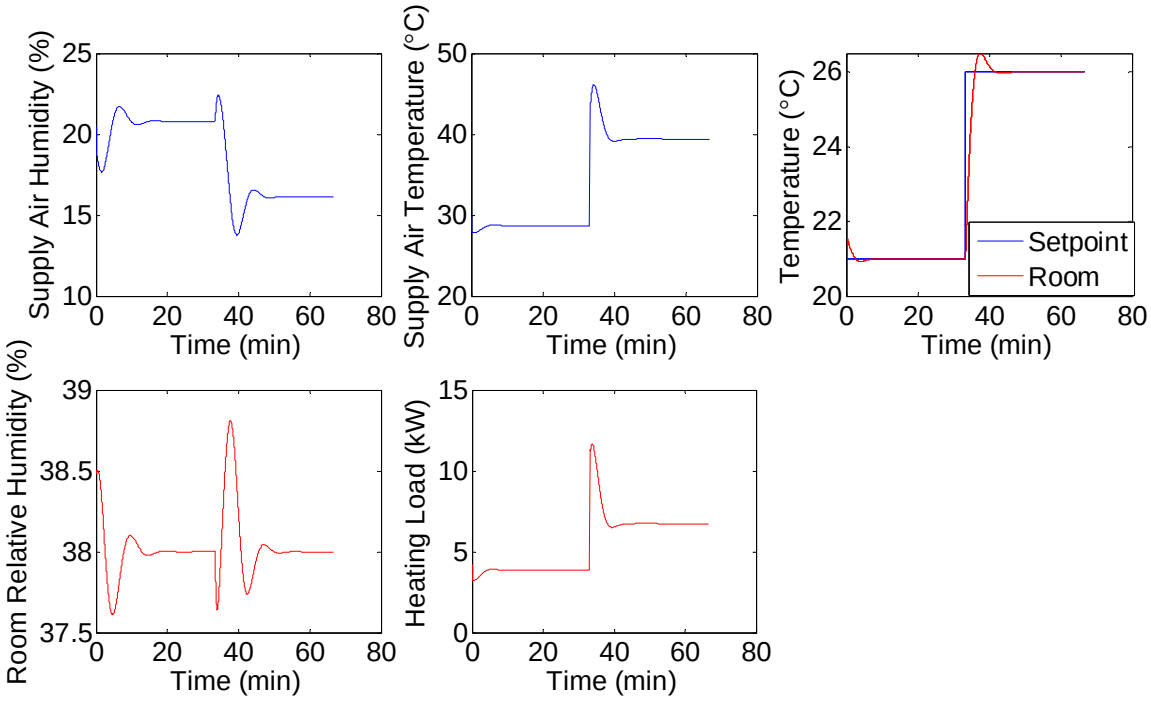


Figure 2.16: Feedback control with room setpoint temperature change.

In Figure 2.17, the supply and return air mass flow rates increases from 0.5 kg/s to 0.6 kg/s. The controlled inputs are supply air temperature and humidity. The controller is a PI controller which drives the room temperature, room humidity and heating load to reach a new steady state. The inputs at the beginning of the simulation change because the initial steady state room temperature and humidity are not at the setpoint values, and the PI controller regulates the controlled inputs, i.e. supply air

temperature and humidity, to reach the setpoint values. This is also why after the supply and return air mass flow rates change, the controller tries to regulate the inputs, i.e. supply air temperature and humidity, to reach the set point values again. The heating load changes with this and reach a steady state.

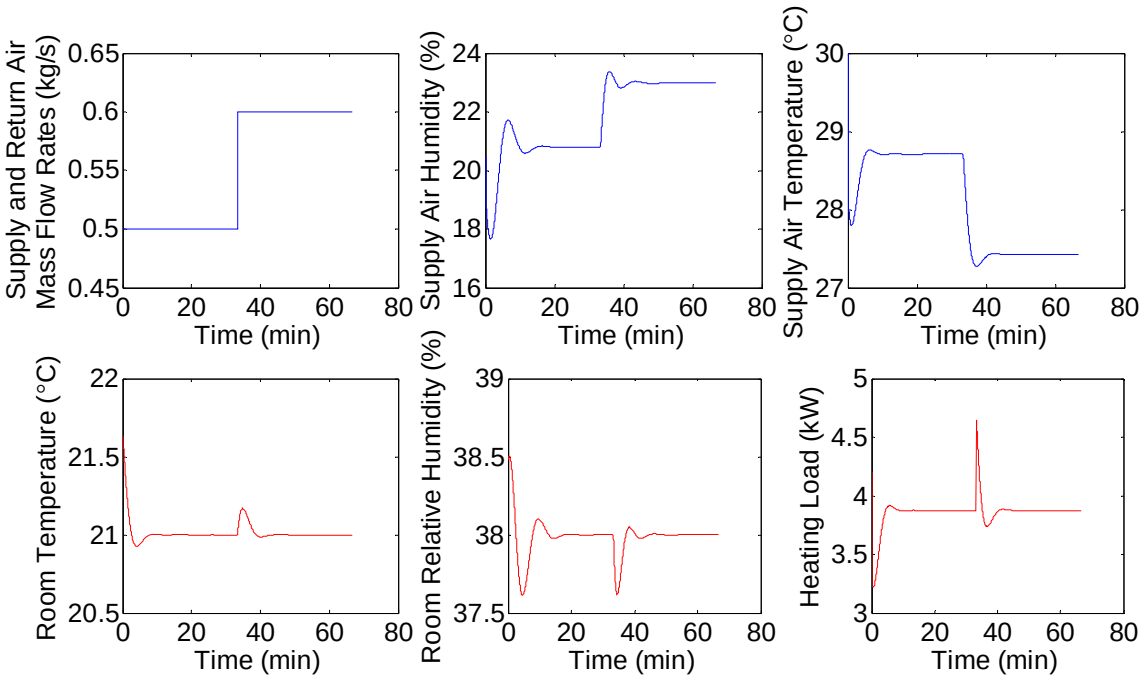


Figure 2.17: Feedback control with supply and return air mass flow rates changes.

In Figure 2.18, the setpoint room humidity increases from 38% to 48%. The controlled inputs are supply air temperature and humidity. The controller is a PI controller which enables the room temperature, room humidity and heating load to reach a new steady state. The inputs at the beginning of the simulation change because the initial steady state room temperature and humidity are not at the setpoint values, and the

PI controller regulates the controlled inputs, i.e. supply air temperature and humidity, to reach the setpoint values. The third figure shows the setpoint humidity and room humidity. After the setpoint humidity changes, the controller tries to match the room humidity with the new setpoint humidity. Other outputs, such as room temperature and heating load, change with this and reach a steady state.

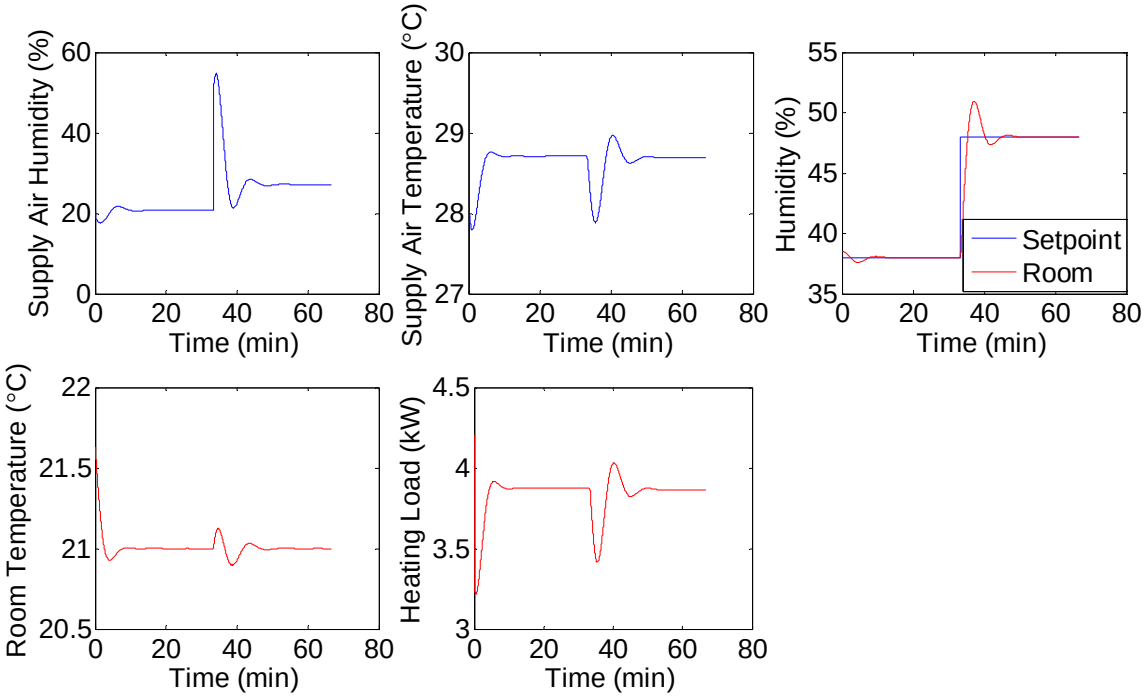


Figure 2.18: Feedback control with room setpoint humidity change.

CHAPTER III

DEVELOPMENT OF A WAVELET-BASED PARAMETER TUNING AND VALIDATION METHOD

This chapter discusses the configuration of the proposed wavelet-based multi-parametric tuning and validation algorithm. First, the application of wavelet theory in signal processing will be introduced. Wavelet analysis is highly instrumental to represent and segment waveforms in signal processing. By using it to establish an objective function, it can greatly capture the dynamics in time series in transient data.

Second, a hybrid optimization algorithm will be introduced as a way to improve model validation efficiency. Hybrid methods are a well-known topic in numerical methods [99]. The key concept is referred to as "synergy", meaning mutually advantageous combination of several approaches.

Finally, the configuration of the proposed parameter tuning and validation method will be explained and discussed in detail for each step. This includes the hybrid algorithm and objective function constructed in this dissertation.

Wavelets Application in Signal Processing

Automatic tuning is used in this study to avoid the problem caused by contrary influences of different parameters on the outputs during manual tuning, which has been mentioned in Chapter I. The core of automatic tuning is defining a metric (i.e. objective function) that minimizes the difference between the experimental and predicted data.

Traditional metrics include direct comparisons of experimental and predicted data using the sum of the root mean square (RMS) difference and the norm of predicted residuals (e.g. least square error). These direct comparison can be found in Rabehl et al. [38] and Nassif et al. [40]'s study. The dynamic characteristics of the model may be lost during comparison because direct comparison is a point-to-point method. These traditional metrics are susceptible to white noise and become inaccurate if the gradients of the parameters are high. Additionally, the multiple outputs with various units cause the problem of how to weight their influences in the objective function.

Wavelets, which improve the fidelity of the dynamic responses, can readily solve these problems and are utilized in this study to define the metric. The use of wavelets in parameter tuning is novel since it compares the aperiodic transient functions, considering wavelets are featured with compact support and arbitrary shift in time. It can handle multiple output signal domains, even from different data sets. Wavelets are less sensitive to white noise, and retain comprehensive information. This feature is extremely important in processing dynamic signals of HVAC systems, since the outputs often contain complicated dynamic information.

Wavelet analysis is highly instrumental to represent and segment waveforms in signal processing. A similar approach to wavelet analysis is the well-known conventional Fourier analysis, but it is incapable of depicting evolutionary spectral features of transient processes and preserving the time dependence. Due to the averaging over the signal duration, the spectral analysis of transient responses cannot capture the local dynamics. Wavelet transform, however, is an effective tool in allowing frequency

and time localization beyond Fourier analysis. It uses a series of local orthogonal basis functions, and preserves local transient features beyond the infinite harmonic basis functions. This is usually achieved through the multi-resolution representation of the dynamic process.

In 1982, a French engineer named Jean Morlet, studied seismological data for an oil company and first proposed the concept of wavelet analysis to find the optimal balance between time and frequency resolution [44][45]. Wavelet has been considered an extension of Haar and Gabor's ideas [46]. This concept was later fixed by Meyer [47], Mallat [48] and Daubechies [49]. Sweldens [50] then further developed this theory. More recently, new ideas about wavelets were published by Candes [51], Do and Vetterli [52], and Donoho and Duncan [53].

Wavelets are efficient in solving difficult issues in a variety of engineering fields, such as ocean and wind engineering. In 1990, Ledueq [54] used wavelet analysis to study the hydraulic noise of a centrifugal pump, which may be the first use of wavelet application in diagnostics. Wang and McFadden [55] proved that wavelet is capable of studying incipient mechanical failure. They also performed further research about wavelet applications [56][57]. A more comprehensive study of wavelet application was found in Newland's [58]~[61] study. He listed several application examples about the use of wavelets in vibration signal analysis. He also identified the ridge and phase of transient signals by using wavelet transforms [62]. Gurley and Kareem [63] comprehensively used wavelet transforms to analyses transient data in earthquake, wind and ocean engineering.

The ability to capture time-frequency features qualifies the use of wavelets in transient signal analysis. Chancey and Flowers [64] found a relationship between the transient signals and the absolute value of the wavelet coefficients when they used harmonic wavelets to identify transient characteristics. Kang and Birtwhistle [65] also proposed a wavelet-based approach to capture the vibration transients of the power transformer on-load tap-changer. Wang [66] detected the transients from different mechanical systems using wavelet coefficients. His approach can be applied in impact signals from faulty machinery parts, spikes in motor current caused by undesirable load variations, and earthquake waves. Gaberson [67] also used wavelet transform to locate the position and magnitude of transient responses in machinery signals. Chen et al. [68][69] performed wavelet decomposition and selected wavelet coefficients to analyze a refinery fluid catalytic cracking process. Lin and Qu [70] used wavelet theory to extract the impulse responses from vibration signals. Yen and Lin [71] decomposed the transient signals using the wavelet packet transform, and the wavelet coefficients were used as fault features with the aid of a statistics-based criterion. Goumas [72] studied the transient signals given by washing machines through wavelet coefficients. Similar research was also done on washing machines by Stavrakaki et al. [73]. Lu and Hsu [74] concluded that the changes in the wavelet coefficients of transient signals were highly sensitive to their transient signals. Liu et al. [75] introduced a wavelet packet-based approach for fault diagnostics. They used wavelet coefficients to capture single features. Dhar et al. [76] used an artificial neural network with wavelets to predict hourly heating

and cooling energy in commercial buildings. Figure 3.1 shows some common types of wavelets [77].

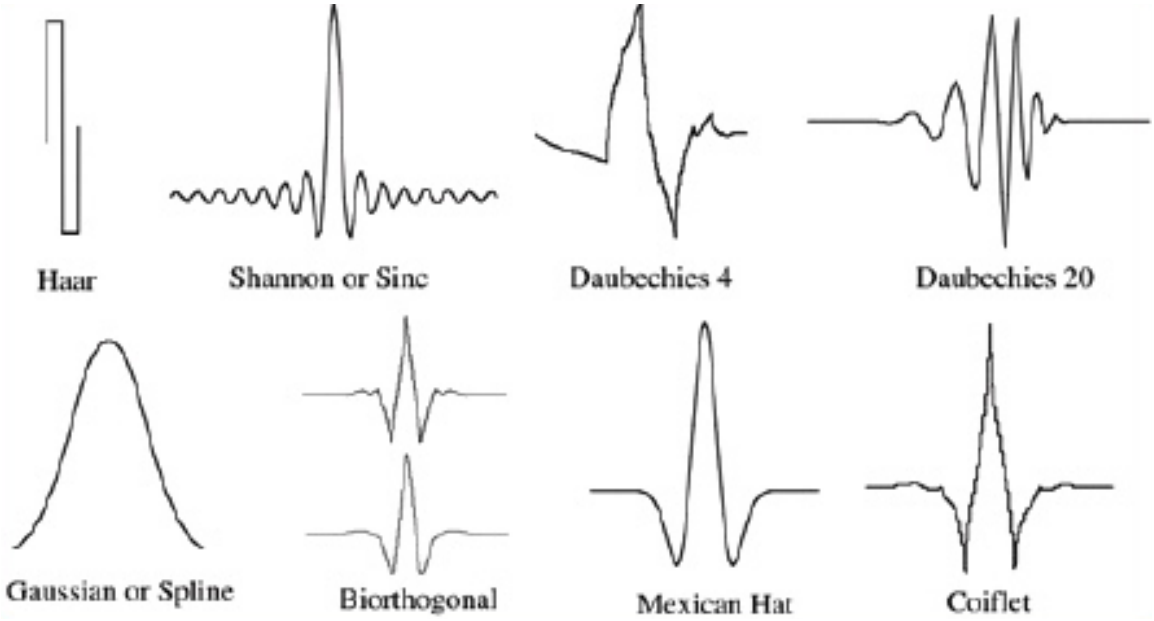


Figure 3.1: Common types of wavelets [77].

Wavelet transform often begins with the generation of a parent wavelet. It is then decomposed into a series of basis functions, which is referred to as wavelet decomposition. The parent wavelet is similar to the sine wave in Fourier analysis. These decomposed functions contain shifted and stretched versions of the parent wavelet. Figure 3.2 shows a typical wavelet decomposition process.

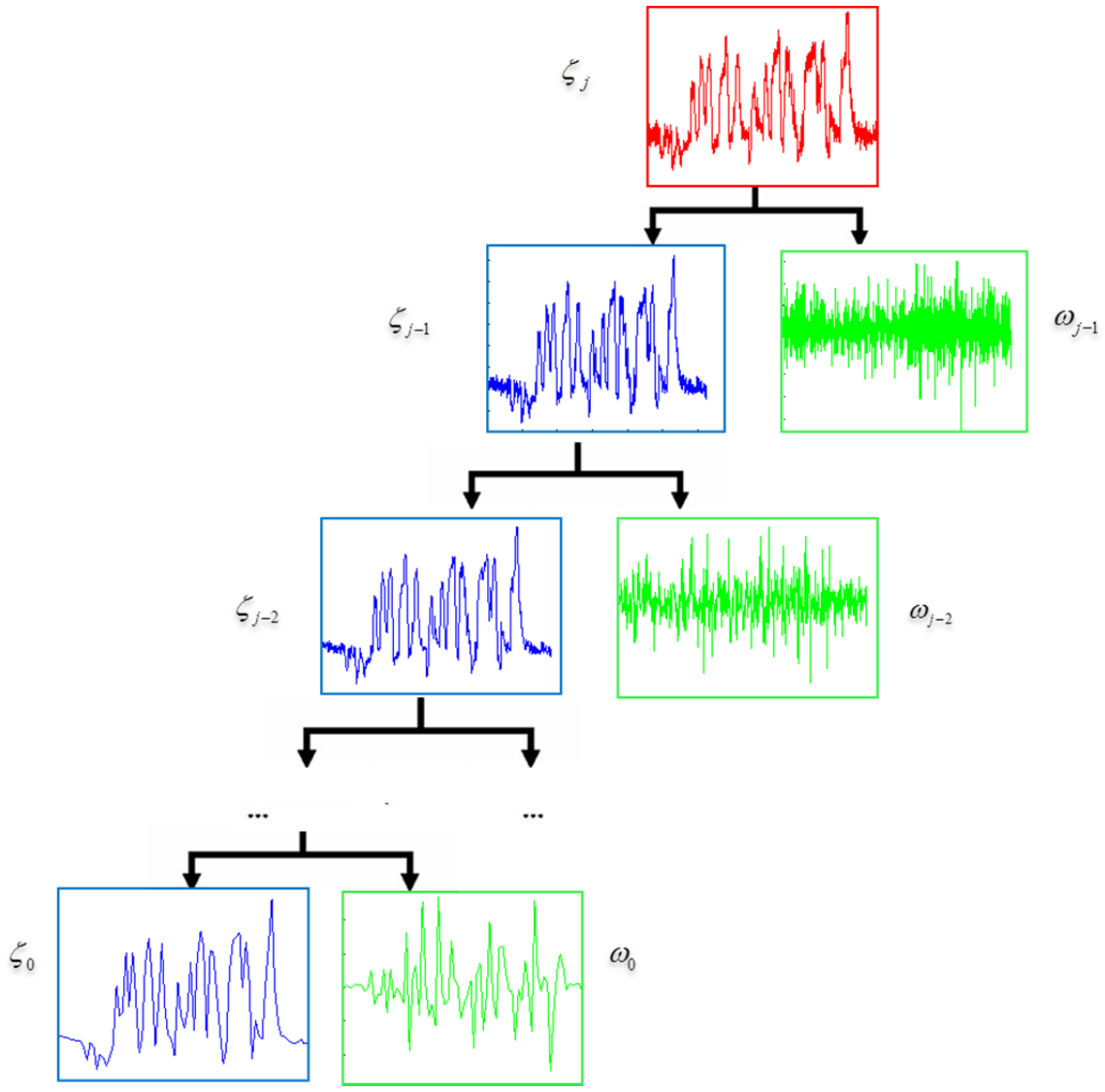


Figure 3.2: A view of wavelet decomposition basis functions.

One way to achieve wavelet decomposition is by expressing the signal as an integral over a range, proposed by Grossman and Morlet [44]. However, expressing the signal as a discrete superposition rather than the integral is more practical, and is more preferred in signal processing. Chui [78] describes some basic concepts of wavelets, such as the wavelet families. Different wavelet families were constructed by Daubechies

[79], Stromberg [80], Meyer [81], Lemarie [82], and Battle [83]. Biorthogonal bases of wavelets were also developed by Cohen, Daubechies and Feauveau [84], and Herley and Vetterli [85]. Cohen, Daubechies and Feauveau [84] did a specific mathematical study and proved that the wavelets indeed can constitute numerically stable bases. One of the first widely used wavelet families was proposed by Daubechies [79] [86]. He developed this parent wavelet by solving a dilation equation to obtain a scaling function. The Haar wavelet is the simplest among the families and is used in this study. The first step of Haar wavelet decomposition is to express the original signal ζ using the step signal

$$\zeta_j(x) = \sum_{k \in Z} c_k \varphi(2^j x - k) \tag{3.1}$$

where j and k are nonnegative integers, and $0 \le k \le 2^j - 1$, the coefficient $c_k = f(\frac{k}{2^j})$, $\varphi(x)$ is the Haar scale function, and is given by

$$\varphi(x) = \begin{cases} 1, & 0 \le x < 1 \\ 0, & x < 0 \text{ and } x \ge 1 \end{cases} \tag{3.2}$$

The original signal ζ is decomposed by

$$\zeta_j = \omega_{j-1} + \zeta_{j-1} \tag{3.3}$$

where

$$\omega_{j-1} = \sum_{k \in Z} d_k^{j-1} \psi(2^{j-1} x - k) \in W_{j-1} \tag{3.4}$$

$$\zeta_{j-1} = \sum_{k \in Z} c_k^{j-1} \varphi(2^{j-1} x - k) \in V_{j-1} \tag{3.5}$$

$$d_k^{j-1} = \frac{c_{2k}^j - c_{2k+1}^j}{2} \tag{3.6}$$

$$c_k^{j-1} = \frac{c_{2k}^j + c_{2k+1}^j}{2} \tag{3.7}$$

where $\psi(x)$ is the Haar wavelet function, and is expressed by

$$\psi(x) = \varphi(2x) - \varphi(2x-1) \tag{3.8}$$

where V_j is a space consisting of $\varphi(2^j x - k)$, W_j is an orthogonal complement space of V_j in space V_{j+1}. ζ_{j-1} can be further decomposed to ω_{j-2} and ζ_{j-2}. Thus,

$$\zeta_j = \omega_{j-1} + \omega_{j-2} + ... + \omega_o + \zeta_0 \tag{3.9}$$

This multi-scale wavelet decomposition process reveals hidden information in the original time history. Wavelet coefficients generated during the decomposition process can aid in the analysis and simulation of transient responses. They can also be utilized to draw out useful signal information. For example, they give the time-scale scalogram which enables the identification of time-varying transient bursts. These are not readily discernible in time or frequency domain analysis. In other words, wavelet coefficients preserve full energy representation to facilitate signal reconstruction and simulation.

Hybrid Optimization

In regard to choosing the optimization algorithm for parameter tuning and validation, gradient and direct search methods are widely used in parameter estimation and tuning [87] ~ [89]. However, these algorithms are sensitive to initial parameters and may also converge to local minima. Therefore, a stochastic global optimization genetic algorithm is adopted to tune the model. This algorithm is independent of the initial guess

point and less sensitive to the local optima which enable avoidance of the spurious solutions given by the gradient search method. This algorithm can also converge to the small range of the global optimum and can handle optimization problems with numerous parameters. The GA was first proposed by Goldberg [90] in 1989 and has been used in optimization and identification in areas such as HVAC systems, proposed by Caldas [91]. The use of the GA has only been recent because it is computationally intensive and requires numerous calculations of the objective function. As a result, the previous decade witnessed a fast rise in using GA [95]~[98]. The GA is found to be a popular global optimization algorithm. Zhang et al. [94] also presented an improved GA for optimizing power system reactive power. The objective function in his paper is to minimize the system active power loss. His refined GA overcomes the drawbacks of conventional reactive power optimization methods.

GA is computationally inefficient at finding the exact value of the optimum. In this paper, a hybrid method is proposed to reduce the computation time while preserving robustness. This method uses the GA to find a set of parameters as the initial guessing point for the gradient method, which serves to refine the parameter tuning results.

Hybrid methods are a well-known topic in numerical methods [99]. The key concept is referred to as synergy, meaning mutually advantageous combination of several approaches. Nevertheless, there are some nontrivial problems about developing hybrid methods: how to choose combining methods and how to construct them. As for the global optimizers, the majority of work has been based on GA, as stated by Chelouah and Siarry [92], and Yen et al. [93].

The focus of using the hybrid method is to determine the transition criteria from the GA to the gradient search method. An early transition point may speed up the tuning process, but it may also cause the convergence to a local optimum. Fernandez et al. [99] proposed a new hybrid method with automatic searching for transition point to perform parameter estimation, and reduced computation time by one order of magnitude. Experiments were also designed to evaluate their method. Maeder et al. [100] also used hybrid optimization method in their effort to estimate the model parameters.

Algorithm Configuration

Figure 3.3 shows the detailed steps of the proposed parameter estimation and tuning method. Above all, physical laws are used to establish the nonlinear model for each component of the system. These models are then linearized respectively and combined to form a complete system model. Discretization and convolution are introduced to predict the outputs of the complete system model using the inputs from the data. These predicted outputs are compared with the experimental data using wavelet coefficients obtained through wavelet decomposition. The objective function generated during the comparison is then optimized using the hybrid algorithm.

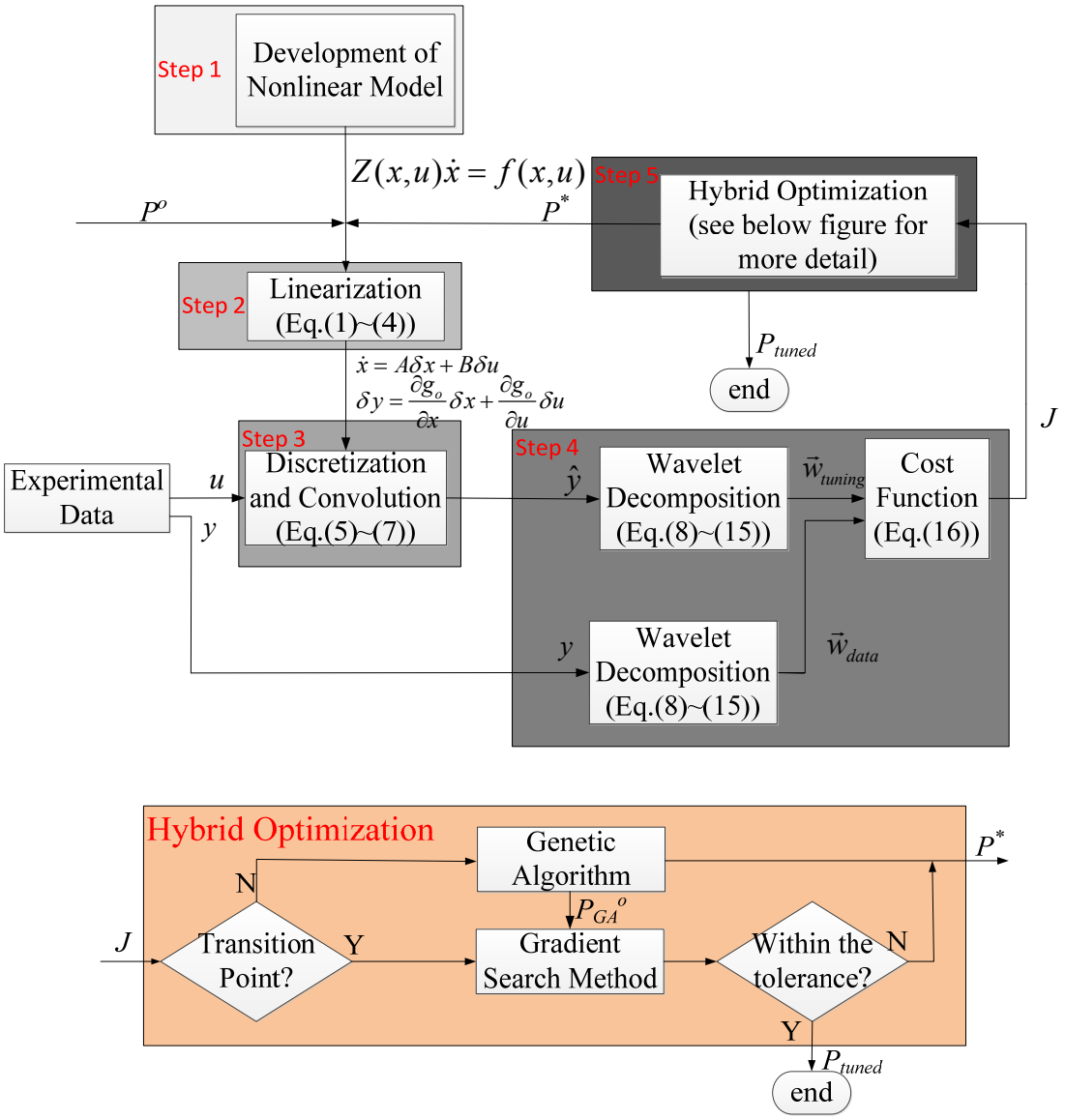


Figure 3.3: Configuration of parameter estimation and tuning method.

Development of Nonlinear Model

Nonlinear models are genearlly categorized into white-box, gray-box and black-box models. White-box models preserve the most robustness as they are established solely on physical laws, while black-box models solely rely on experimental data, thus it

is also known as system identification. This dissertation uses the gray-box model method, which is the combination of the above two methods, to obtain the nonlinear model. Each component model is developed and validated using experimental data or field data.

Model Linearization

The nonlinear model obtained is linearized in order to reduce the total computation time. The nonlinear model is given in the nonlinear descriptor form described in Equation (2.9).

Using $x = x_0 + \delta x$ and $u = u_0 + \delta u$, the state space form $\dot{x} = A\delta x + B\delta u$ is obtained:

$$\dot{x} = Z_{x_0,u_0}^{-1}\left[\frac{\partial f}{\partial x}\Big|_{x_0,u_0}\right](x - x_0) + Z_{x_0,u_0}^{-1}\left[\frac{\partial f}{\partial u}\Big|_{x_0,u_0}\right](u - u_0) \tag{3.10}$$

where x_0 and u_0 are the equilibrium points used for linearization. The outputs are given by

$$y = g_o(x,u) \tag{3.11}$$

such that,

$$\delta y = \frac{\partial g_o}{\partial x}\delta x + \frac{\partial g_o}{\partial u}\delta u \tag{3.12}$$

Discretization and Convolution

The combined linear model is discretized in order to use discrete-time convolution:

$$y(n) = g(n) * u(n) = \begin{cases} 0 & , \ n < M + N \\ \sum_{i=N}^{M} g(i)u(n-i), & n \geq M + N \end{cases} \tag{3.13}$$

where N is the index of the first non-zero value of $g(n)$, M is the index of the first non-zero value of $u(n)$, n is the given time, y is the output, and g is the impulse response with respect to u. The impulse response for MIMO system is the collection of impulse responses for each input:

$$G(n) = \begin{bmatrix} g_{11}(n) & g_{12}(n) & \cdots & g_{1k}(n) \\ \vdots & \vdots & \vdots & \vdots \\ g_{n_o 1}(n) & g_{n_o 2}(n) & \cdots & g_{n_o k}(n) \end{bmatrix} \tag{3.14}$$

where n_o is the number of outputs, and k is the number of inputs.

Each output of MIMO system is obtained using the discrete-time convolution and superposition:

$$y = y_1 + y_2 + \ldots + y_k \tag{3.15}$$

where $y_1, y_2, \ldots, y_k$ are the responses to different inputs calculated by Eq. (5), k is the number of inputs.

Construction of Objective Function

Wavelet decomposition theory is adopted to construct the objective function. This metric is based on the comparison of wavelet coefficients generated by wavelet decomposition. Wavelets are superior because they are less sensitive to white noise, retain comprehensive information, and capture the aperiodic transient functions of the signals. This feature is extremely important in processing the dynamic signals of MIMO systems since the outputs often contain complicated dynamic information.

Wavelet decomposition produces a series of coefficients for predicted and experimental data, and they are normalized to avoid the handling of outputs with different units:

$$J = \sum_{i=1}^{m} \alpha_i \frac{\left\| \vec{w}_{data} - \vec{w}_{tuning} \right\|_2}{\left\| \vec{w}_{data} \right\|_2} \tag{3.16}$$

where $\vec{w}_{data}$ and $\vec{w}_{tuning}$ are the wavelet coefficients at each level defined in Equation (3.4) and (3.5), *m* is the level of wavelet decomposition, and α_i is the weighting factor for each output. The weighting factor can be determined according to the influence of the predicted outputs. This will be discussed in the result section.

Hybrid Optimization and Transient Point

In this research, the hybrid optimization method consists of the GA and a gradient method. When optimization starts, the GA randomly generates an initial population and evolves to the optimal solution. The detailed process of the GA is described as follows. The individuals in the group with different values of tuning

parameters in the current generation yield different cost function values, and then some individuals are stochastically selected. Mutation, crossover or recombination is then used to update these individuals, thus forming a new generation. This process is repeated until the transition criterion is satisfied. The solution given by the GA serves as the initial value for the gradient search method, which is repeatedly used to find the exact global optimum until the tolerance is met. The GA does not require an initial guess value, but constraints on the parameters are determined based on the properties. For instance, the constraint of the diameter may be set to 10% to 200% of the value provided by the equipment supplier.

The focus of using the hybrid method is determining the amount of search performed by each algorithm. The transition point largely affects the robustness and efficiency of the hybrid method. An early transition point may decrease computation time but fail in finding the optimal solution, while a later transition point is more likely to find the optimal solution at the expense of large computation cost. Compromising between robustness and efficiency is a challenge with this method. In making this compromise, one has to guarantee that the convergence is within the vicinity of the optimal solution and to keep the computation time within the design requirement.

In order to show the significance of using the hybrid method, a gradient search method is used after the GA stops. The efficacy of each optimization algorithm can be evaluated using the percent of time (*POT*) and percent of improvement (*POI*) spent on each parameter estimation stage. The *POT* is the time fraction spent on each optimization stage, and the *POI* is defined as:

$$POI = \frac{\varepsilon_k - \varepsilon_{k-1}}{\varepsilon_{norm} - \varepsilon_{final}} \times 100\% \tag{3.17}$$

where ε_{norm} is the relative error obtained by nominal parameters, ε_k and ε_{k-1} are the relative error at current and previous parameter estimation stage, and ε_{final} is the final relative error of the complete parameter estimation process. The relative error in is defined as:

$$\varepsilon = \sum \frac{\|y - \hat{y}\|}{\|y\|} \tag{3.18}$$

where y is the experimental data and $\hat{y}$ is the predicted outputs.

Wavelet-Based Objective Function

The weighting factors for the cost function are determined as follows. Take the evaporator for example. The pressure generally has more influence because it greatly affects other outputs, i.e. superheat and mass flow rate. On the other hand, the outlet refrigerant temperature has little significance since it is rarely used in control design. Thus, the weighting factor for the pressure should be larger than other outputs, and it should be smaller for the outlet refrigerant temperature (or even set to zero) if matching it with the data is not significant.

CHAPTER IV

VERIFICATION OF PROPOSED PARAMETER TUNING AND VALIDATION METHOD

To verify the proposed parameter tuning and validation method, it is applied on a simple experimental HVAC system with the heat exchanger modeled using moving boundary method. As mentioned in Chapter I, moving boundary method can accurately predict the salient dynamics of multiple fluid phase heat exchangers while preserving the simplicity of lumped parameter method. This feature makes it suitable to verify the proposed tuning and validation method, as it is important that the model used in this method is both accurate and fast.

A simple HVAC system is modeled using moving boundary method for the heat exchangers. This system consists of an evaporator, an electronic expansion valve, a condenser and a compressor. The configuration of this HVAC system can be seen in Figure 1.1.

This chapter first compares the proposed simplified model, i.e. discrete-time convolution linearized model, with the nonlinear model; both were developed through moving boundary method. The idea is to prove that the proposed simplified model can largely reduce simulation time, and the possibility to eliminate the characteristics of singularly perturbed systems as it can adopt the same sample time as that of the data. This characteristic is extremely advantageous over the nonlinear model which generally requires small numerical step size in order to simulate.

Second, a moving boundary evaporator model is tuned and validated using the simulated data obtained from a linearized evaporator model. This is an attempt to verify the proposed parameter tuning and validation method, as it determines whether or not the proposed method can find the hypothetical parameters used to generate the simulated data.

The experimental system used to collect data for parameter tuning and validation is then presented. This experimental system is a custom-built and small-scale air-conditioner system. This system is designed and established for the purpose of dynamic model validations, control design and fault detection and diagnosis. The test results for low cooling and high cooling are also presented.

Finally, the simple HVAC system model is tuned and validated using the obtained experimental data to verify the proposed tuning and validation method. In this scenario, several tuning and validation issues are specifically discussed. The tuned model is also cross-validated using experimental data with different operating conditions.

Linear Model with Discrete-Time Convolution vs. Nonlinear Model

Using the linearization in step 2 of the proposed method, which is described in the previous chapter of this dissertation, the nonlinear moving boundary heat exchanger HVAC system models developed in Chapter II are linearized. The outputs of the linear model are further obtained using discrete-time convolution. The nonlinear and linear discretized models are simulated for a data range of 2500 sec, and the computation time

for each model is listed in Table 4.1. The outputs comparison is shown in Figure 4.1. The simulations were conducted on a standard desktop computer: an Intel Core i-3 Processor with the clock speed of 3.2 GHz and an 8 GB memory on a 64-bit Windows 7 operating system.

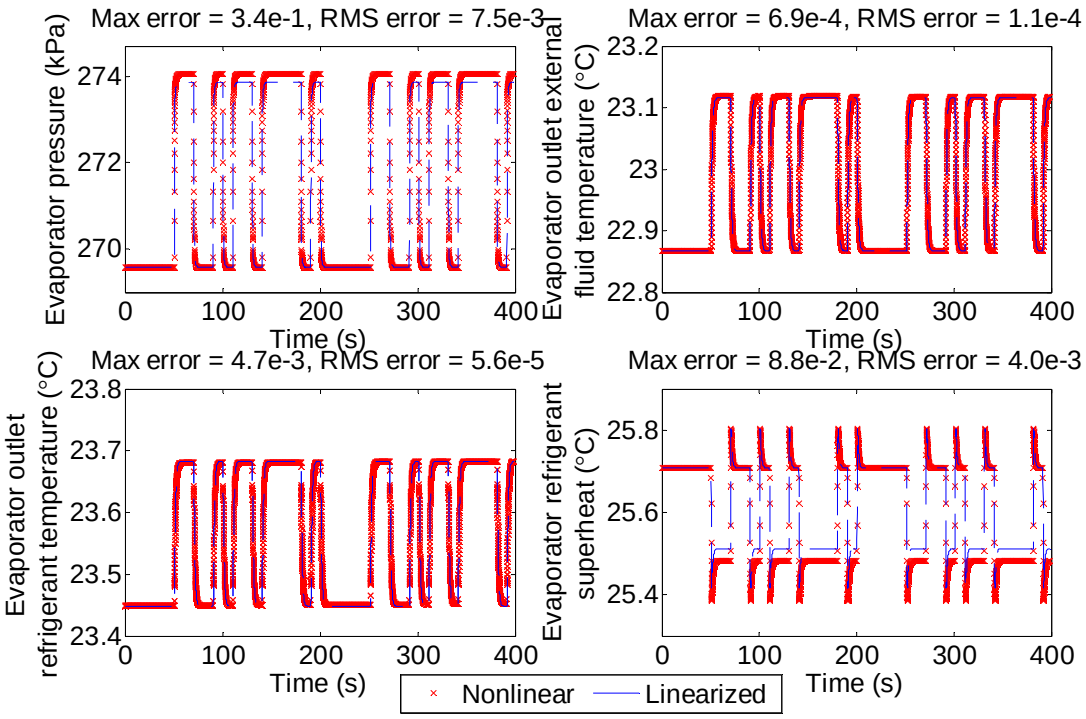


Figure 4.1: Linearized discrete-time model outputs vs. nonlinear model outputs.

The results indicate that the linear model with discrete-time convolution is slightly different from nonlinear model, and is almost 1165 times faster than the nonlinear model. Therefore, linear model with discrete-time convolution is an accurate representation of the nonlinear model. Moreover, since the simulation time of the model

is largely reduced, the parameter tuning time will be drastically shortened as repeated simulations of the model are required.

Table 4.1: Computation time of nonlinear and linear model.

Model type	Computation time (s)	
Nonlinear model	**213.32**	
Linear model with discrete-time convolution	Simulation step	
	1st :Solving for Mass/Energy Equations	0.0576
	2nd :Discretization	0.0609
	3rd: Convolution	0.0645
	Total	**0.1830**

Verification Using Simulated Data

Using a series of hypothetical parameters, the simulated data of a linear evaporator model is used to tune the model itself to verify the feasibility and efficacy of the proposed method. The simulated data was generated from a linearized moving boundary evaporator model established in MATLAB/Simulink. The inputs and outputs of this model are shown in Figure 4.2. The inputs include change in inlet refrigerant enthalpy, which was a series of random step changes generated by a MATLAB function. Other inputs were kept constant. Figure 4.3 and Figure 4.4 show the inputs and outputs of the generated simulated data for one set of hypothetical parameters.

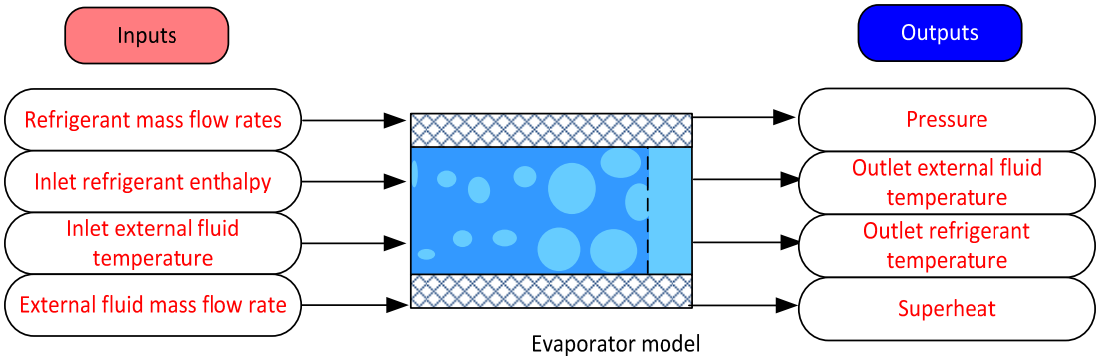


Figure 4.2: Inputs and outputs of MB evaporator model.

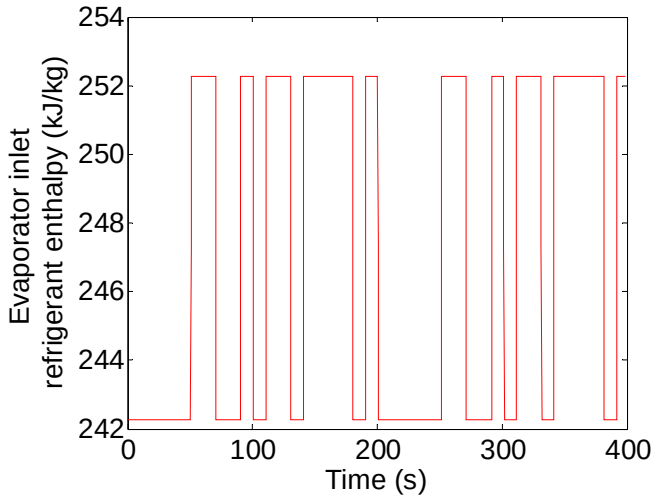


Figure 4.3: Input change of MB evaporator model established in MATLAB/Simulink.

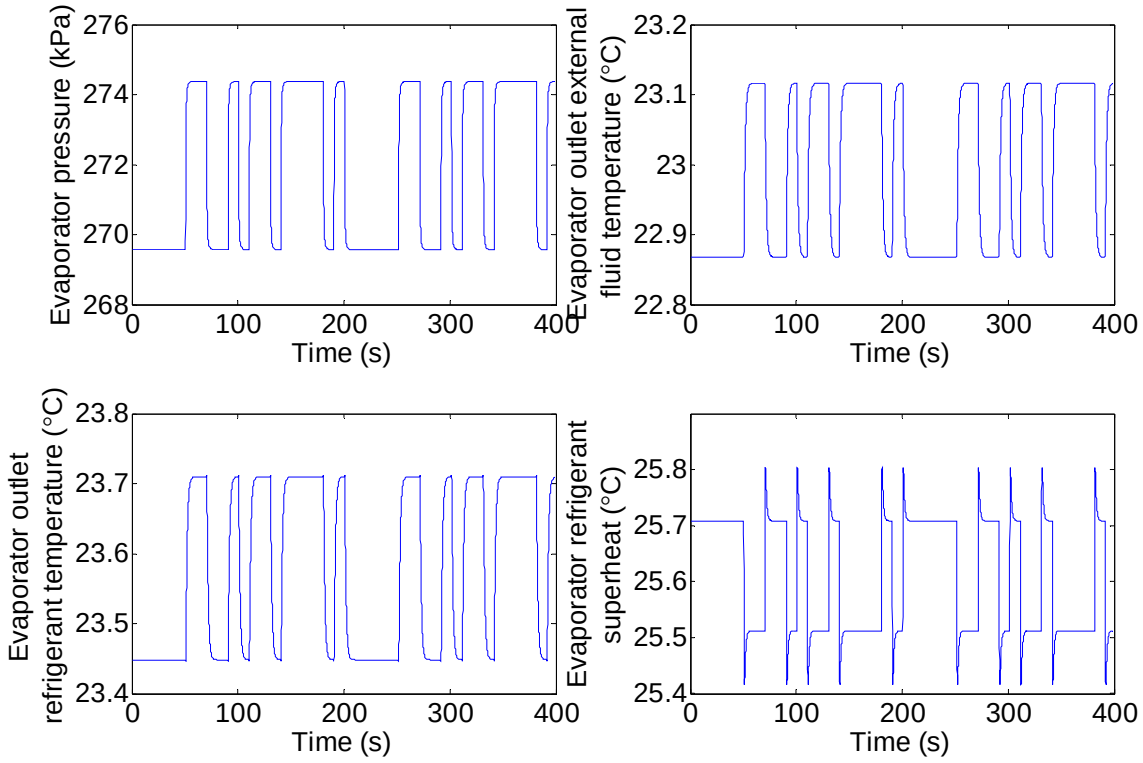


Figure 4.4: Outputs of linearized moving boundary evaporator model established in MATLAB/Simulink.

Using wavelet reconstruction theory, i.e. the reverse of decomposition, the outputs in Figure 4.4 can be retrieved from wavelet coefficients generated from wavelet decomposition. The original and reconstructed model outputs are compared in Figure 4.5. This signifies the accurate representation of the data using wavelet coefficients.

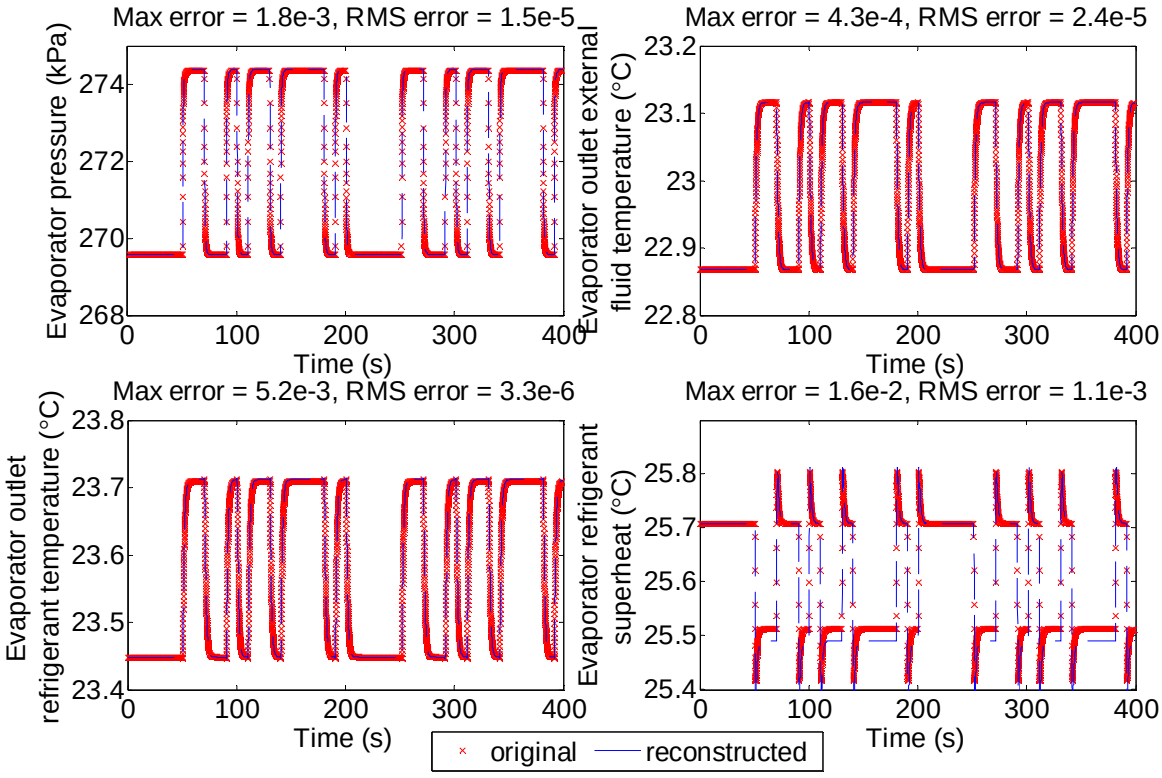


Figure 4.5: Reconstructed model outputs vs. original model outputs.

A range of the hypothetical parameters is provided for tuning, the purpose of which is to test whether or not the proposed method can find the hypothetical parameters used to generate the simulated data. Monte Carlo simulation was conducted to explore the sensitivity of the linearized discrete-time evaporator model by varying parameters within statistical constraints. This process is easily realized in GA since it is a stochastic optimization method which randomly generates an initial population when the optimization starts.

In the Monte Carlo simulations, a total of 100 parameter tuning simulations were performed for this model and the results are shown in Figure 4.6 and Figure 4.7.

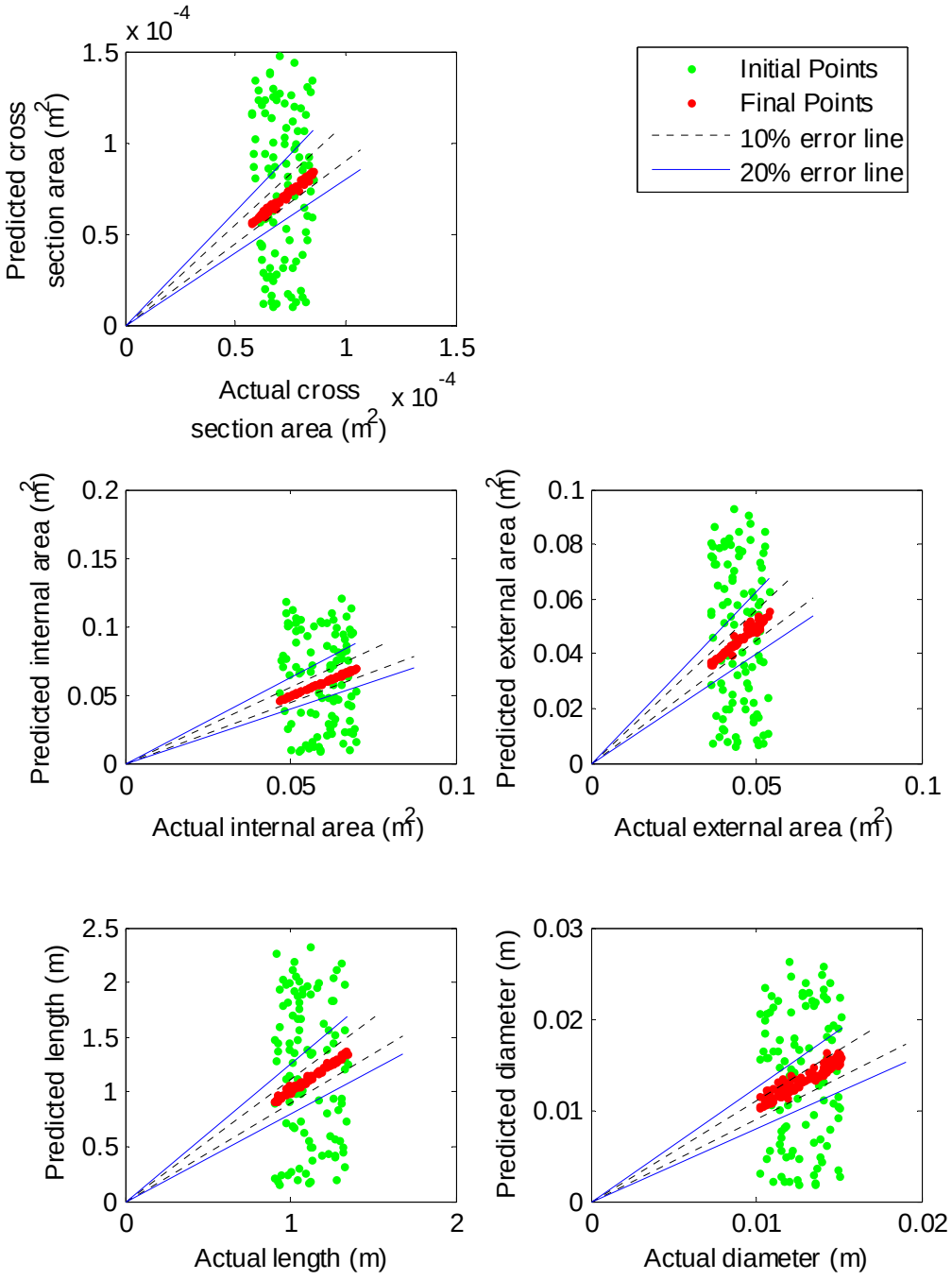


Figure 4.6: Monte-carlo simulation results.

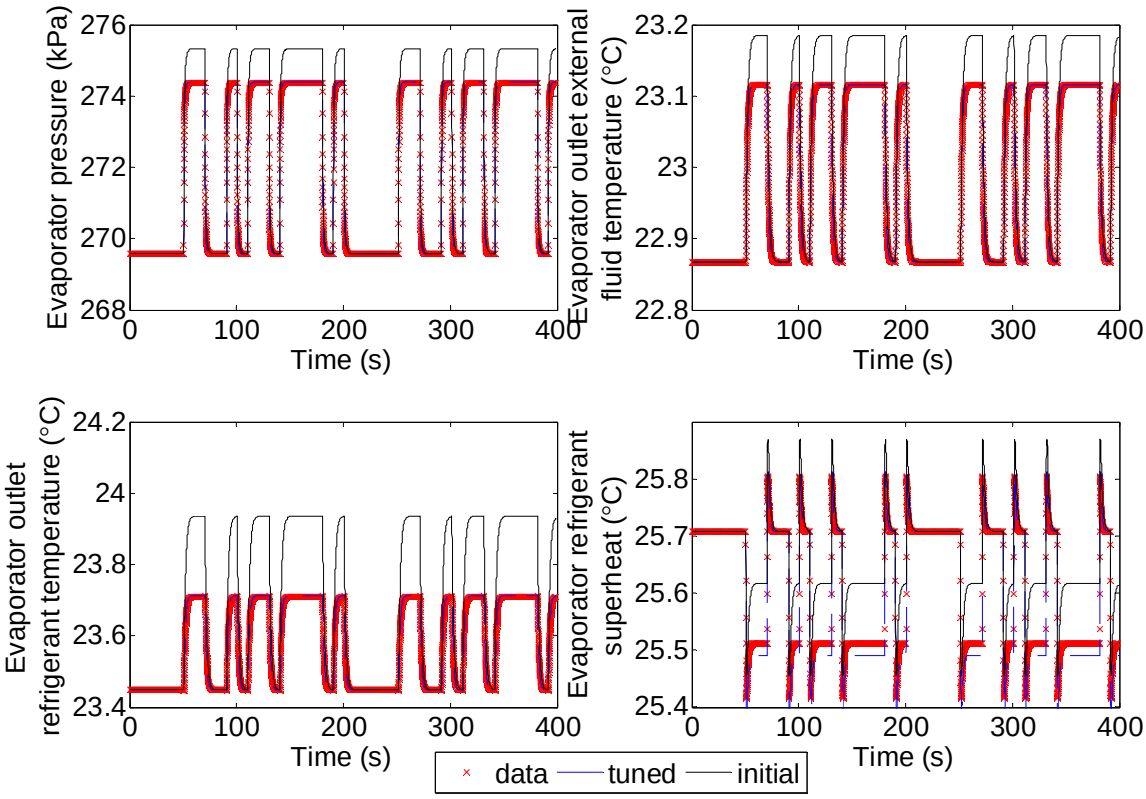


Figure 4.7: Tuning results using the evaporator simulated data.

The horizontal coordinate represents the actual parameters that were used to generate the simulated data. The vertical coordinate represents the predicted/tuned parameters given by the proposed parameter tuning and validation method. These figures indicate that the initial parameters, shown by the green dots, diverge from the actual values, but they gradually converge to the final predicted values, shown by the red dots. Figure 4.7 shows the predicted outputs using one of the 100 final tuned parameters.

These results show that the predicted parameters are close to the actual values. The predicted outputs obtained from the model with tuned parameters match very well with the simulated data. The proposed estimation and tuning method can indeed enable

the predicted parameters to converge to the actual values, even if the initial points diverge from the actual values.

Experimental Test

A series of experimental tests were performed in order to be used for the verification of the proposed tuning and validation method.

Experimental Apparatus

The simple HVAC system modeled in Chapter II was constructed in the Thermo-Fluids Control Laboratory. It is a custom-built and small-scale air-conditioner system. This system is designed for the purpose of dynamic model validations, control design and fault detection. It consists of a primary (refrigerant) system and a secondary (water) system, shown in Figure 4.8.

The specifications of the sensors used in this system are shown in Table 4.2. To measure fluid temperature, type T thermocouples are immersed in the tested fluid. These thermocouples are of the low-noise variety, and they have ungrounded sealed tips. The thermocouples are immersed in the tested fluid; a Swagelok tube fitting grips the shaft of the thermocouple, sealing the tested fluid from the air. It is manufactured by CR Magnetics.

Figure 4.8: Experimental system.

Table 4.2: Specifications of the sensors.

Sensor	Mfr.	Part Number	Operating Range	Output Listed	Accuracy +/-
Thermocouples	Omega	GTMQSS-062U-6	-270-400 °C	TC	0.5 °C
Evaporator pressure	Cole-Parmer	07356-03	0-160 psig	1-5 V	1.0%
Condenser pressure	Cole-Parmer	07356-04	0-300 psig	1-5 V	1.0%
Refrigerant flow	McMillan	102-5-E-Q-B4-NIST	50-500 mL/min	0-5 V	3.0%
Compressor current	CR Magnetics	CR5210	0-50 amps DC	0-5 V	1.0%
Tachometer	Masterflux	-	1800-6500 RPM	0-2600 Hz	-

Pressure is measured using sealed stainless steel diaphragm-type pressure transducers manufactured by Cole-Parmer. A transducer with maximum pressure of 300 psi is used to measure pressure at the outlet of the condenser, and a transducer with maximum pressure rating of 100 psi is used at the inlet of the compressor.

To measure the mass flow rate of the refrigerant, three McMillan Volumetric turbine-style flow meters are installed at the end of the evaporators. These transducers output a 0-5V signal to the DAQ board. The compressor motor control outputs a 0-5V tachometer pulse that indicates motor speed. The frequency of the pulse is proportional to the motor speed.

Test Results

In this experimental test, both low cooling and high cooling tests were performed on the apparatus described in the previous section. The sample time is 1 second. For the low cooling test, the test scenarios include the random step changes of low electronic valve opening area and low evaporator external mass flow rate. Other inputs, i.e. evaporator inlet external fluid temperature, compressor speed, condenser inlet external fluid temperature and mass flow rate were kept constant during the test. Figure 4.9 and Figure 4.10 show the inputs and outputs of the low cooling test, respectively.

For the high cooling test, the test scenarios include the random step changes of large electronic valve opening area and evaporator external mass flow rate. Other inputs, i.e. evaporator inlet external fluid temperature, compressor speed, condenser inlet external fluid temperature and mass flow rate were kept constant during the test, although the evaporator and condenser external fluid inlet temperature have some fluctuations. Figure 4.11 and Figure 4.12 show the inputs and outputs of the high cooling test, respectively. The operating conditions in these tests are within the range of the sensors, signifying low errors of the measurements.

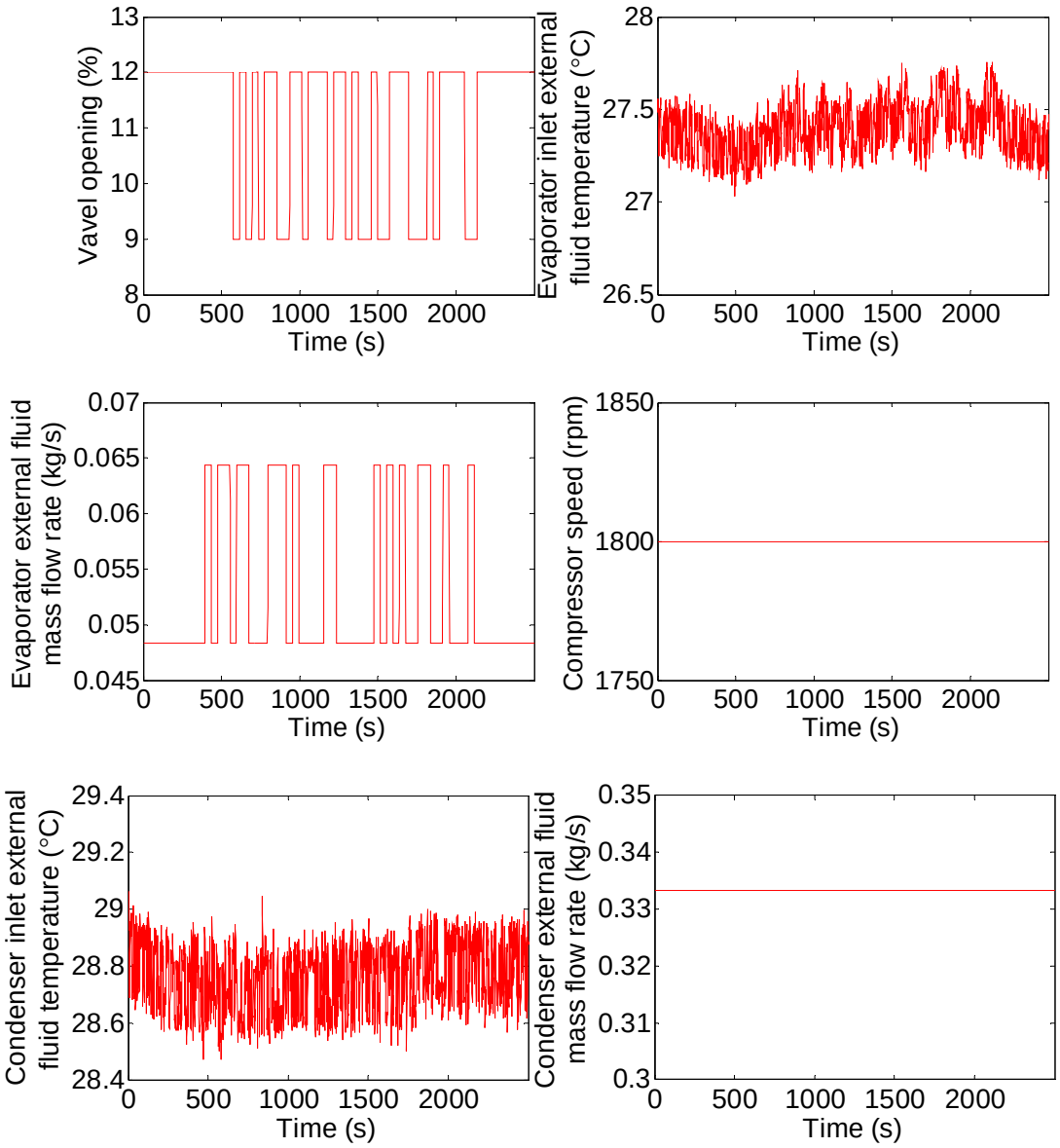


Figure 4.9: Inputs of low cooling test.

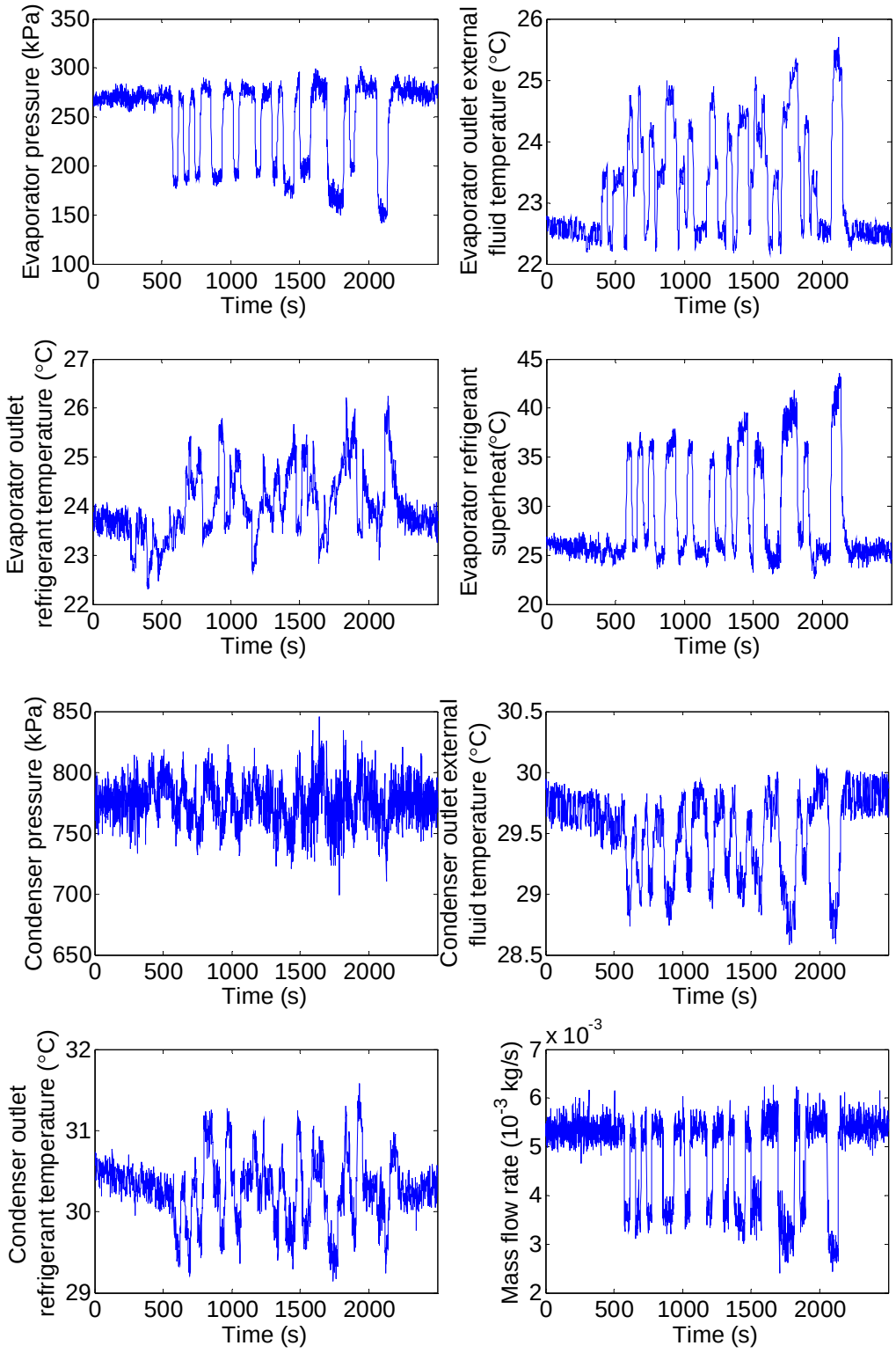


Figure 4.10: Outputs of low cooling test.

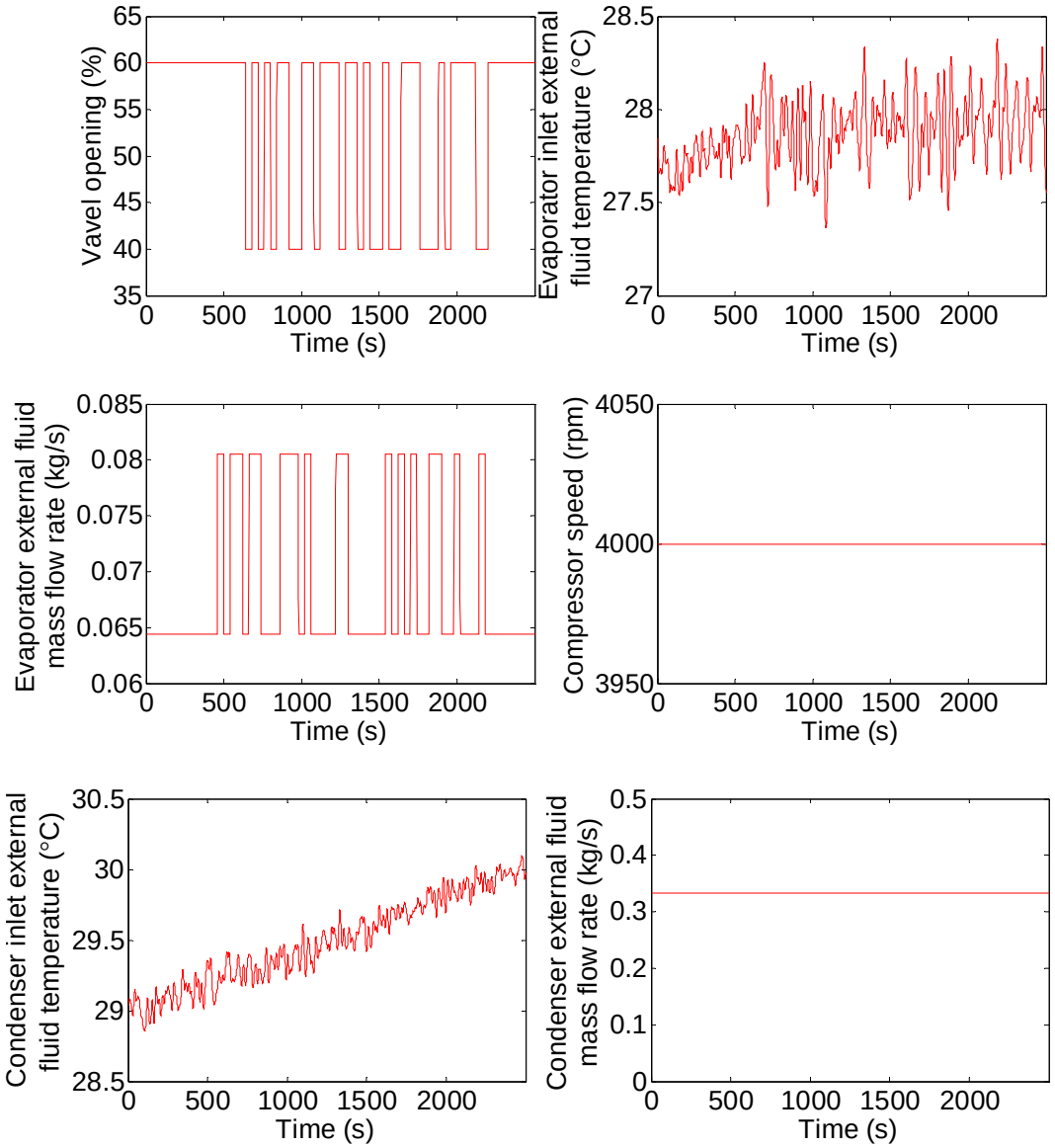


Figure 4.11: Inputs of high cooling test.

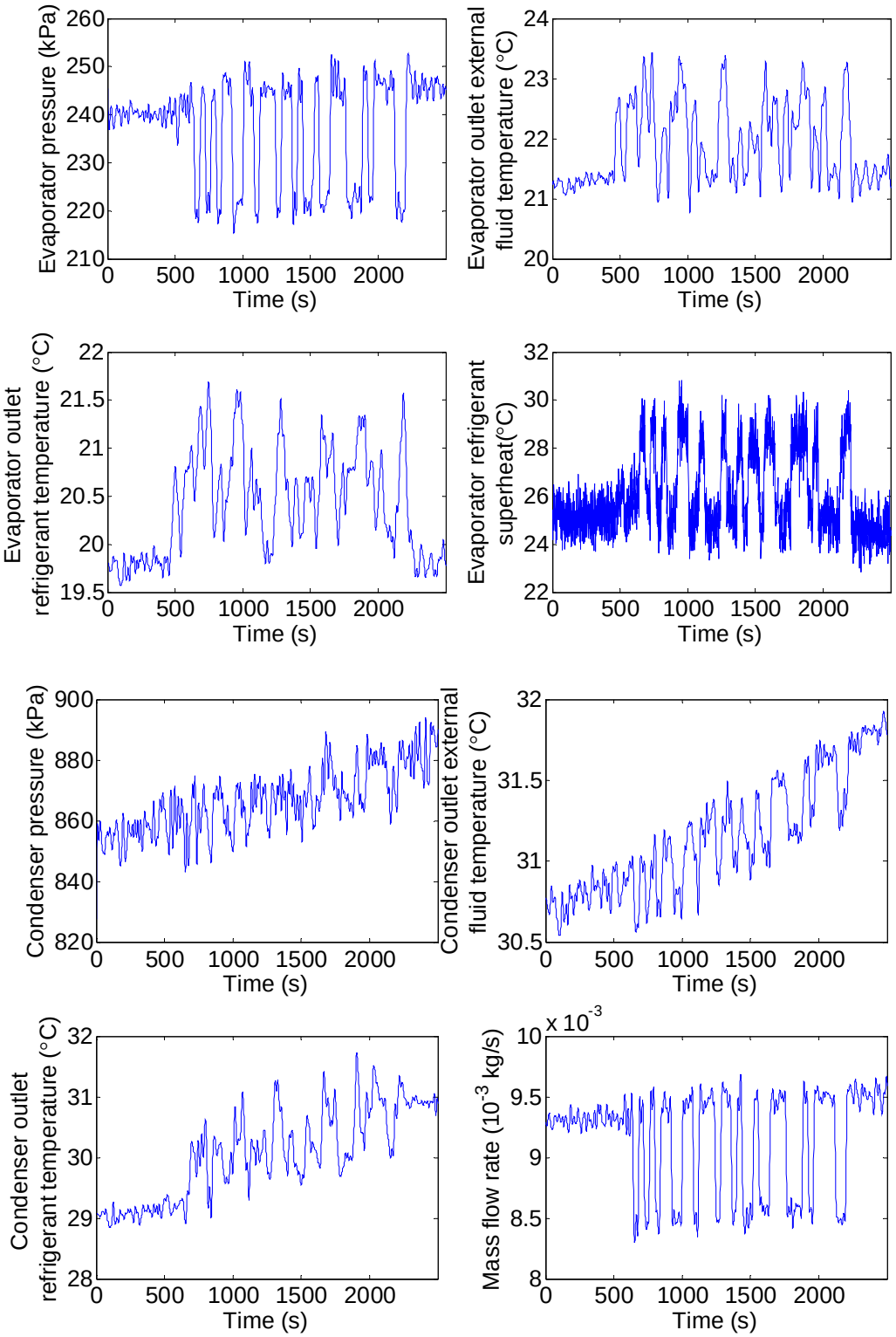


Figure 4.12: Outputs of high cooling test.

Heat Exchanger Model Parameter Tuning and Validation

The evaporator model is then tuned using experimental data, and the results are shown in Table 4.3 and Figure 4.13. The tuning parameters for this evaporator include cross section area, internal surface area, tube length and diameter. These parameters are lumped parameters because approximations are made during modeling. Therefore, some physical parameters become immeasurable. The selected data set has 2500 data points but only 1400 points are shown for clarity. The sample time is the same as the data: 1 sec.

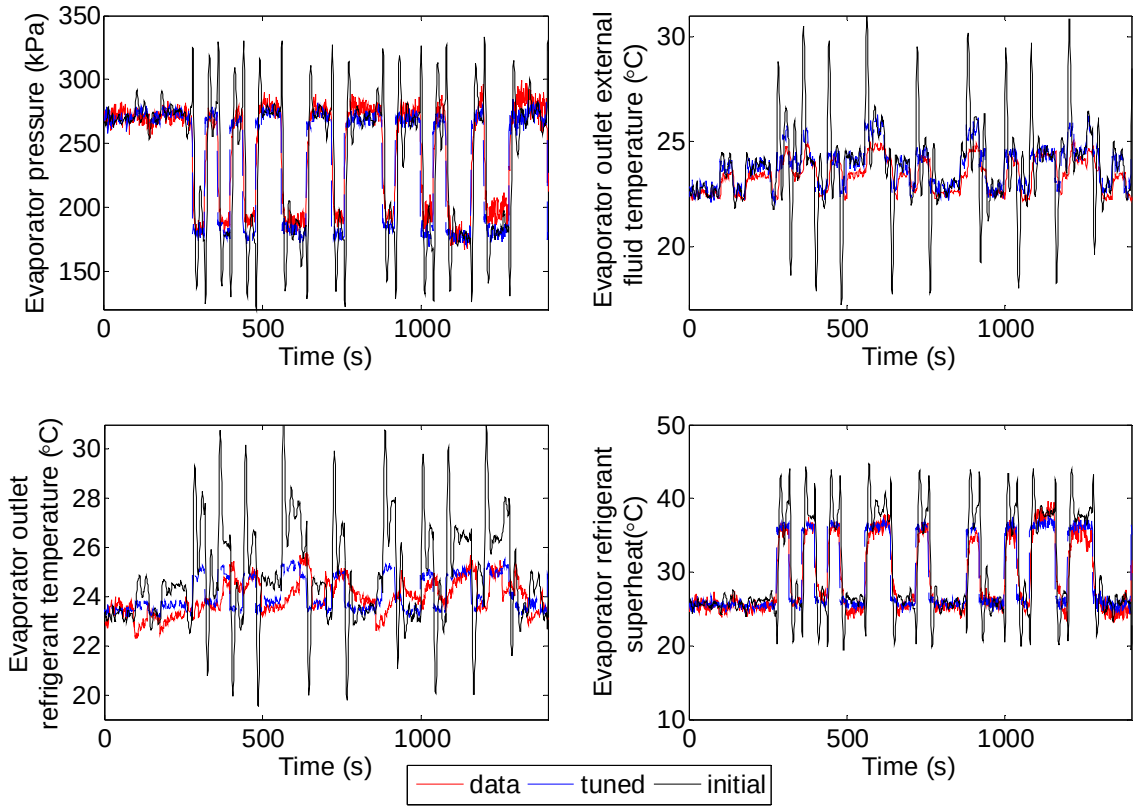


Figure 4.13: Tuned results using the evaporator experimental data.

Table 4.3: Summary of parameter tuning and validation results.

Models	Tuned parameters	Predictive ε (%)		Nominal ε (%)	Simulation time (s)
Evaporator	$A_{e,cs}$, A_{ei}, A_{eo}, $L_{e,total}$, d_e	4.56		10.9	354.8
Complete system (single data set)	$A_{e,cs}$, A_{ei}, d_e, $d_{c,}$ $A_{c,cs}$, A_{ci}, valve, compressor map coefficients	3.58		11.5	456.1
Complete system (multiple data sets, same tuning parameters)		Dataset1	6.72	Data set 1 11.5	834.4
		Dataset2	5.21		
Complete system (multiple data sets, different tuning parameters)		Dataset1	6.53	Data set 2 12.4	1336.2
		Dataset2	4.82		

These figures indicate that the proposed parameter estimation and tuning approach can indeed enable the predicted outputs to evolve towards the data, even if the outputs given by the initial model are disparate from the data. Table 4.3 also shows that the relative error of the model outputs with tuned parameters is lower than that of the nominal parameters. The computation time is only 354.8 sec.

The uncertainty of the sensors used in experiments has effect on the experimental results. In Figure 4.13, only the uncertainty of the superheat is derived from other uncertainties shown in Table 4.2. The uncertainty of superheat can be calculated by

$$uT_{sh} = \sqrt{uT_{e,ro}{}^2 + uT_{e,sat}{}^2} \tag{4.1}$$

where $uT_{e,ro}$ is the uncertainty of the thermocouple measuring outlet refrigerant temperature given in Table 4.2, $T_{e,sat}$ is the uncertainty of saturated temperature and is expressed by

$$uT_{e,sat} = \frac{\partial T_{e,sat}}{\partial P_e} uP_e \tag{4.2}$$

where uP_e is the uncertainty of the pressure measured by the sensor given in Table 4.2.

According to these equations, the averaged uncertainty of superheat for the simulated data range is calculated to be 0.35 °C. The individual absolute error of the tuned superheat, i.e. the absolute difference between the data and tuned value, is 0.28 °C. This signifies the tuned results have a smaller uncertainty than sensor uncertainties.

HVAC Model Parameter Tuning and Validation

Selection of Tuning Parameters

Complete HVAC models contain lots of parameters. Tuning all of them would greatly increase computation time. Selection of tuning parameters is necessary before tuning the complete HVAC system model. This task can be performed by studying their influences on the output error, i.e. parameter sensitivity study. The parameters studied are shown in Table 4.4. These parameters can be tuned because modeling assumptions cause them to become lumped parameters.

A simple way to do this is calculating the relative error of predicted outputs at minimum and maximum parameter values. The change of relative error can tell how much the parameter variation influences the predicted outputs. This dissertation sets the

minimum and maximum parameter variation as -50% and 50%~90% of the initial value. The results are shown in the last column in Table 4.4.

Table 4.4: Influences of different parameters on the output error.

Parameters	Minimum value change (%)	Relative error change (%)	Maximum value change (%)	Relative error change (%)
$A_{e,cs}$	-50	+1.4	+50	-0.72
A_{ei}	-50	-1.5	+50	+1.0
A_{eo}	-50	+0.0010	+50	+0.0010
$L_{e,total}$	-50	+0.079	+50	-0.075
d_e	-50	+0.33	+50	-0.18
$A_{c,cs}$	-50	+0.24	+50	-0.20
A_{ci}	-50	-0.31	+50	0.14
A_{co}	-50	-0.00042	+50	-0.00042
$L_{c,total}$	-50	+0.13	+50	-0.093
d_c	-50	+0.059	+50	-0.036
C_d map coefficients	-90	+0.24	+90	-0.14
V_k	-50	-0.073	+50	+0.078
η_a and η_v map coefficients	-90	-0.23	+90	+0.15

Tradeoffs are made when selecting tuning parameters from these results. The more parameters chosen, the more tuning time required. This dissertation recommends choosing parameters that have at least 1 magnitude influence on the output error. According to this, the relative errors given by A_{eo}, $L_{e,total}$, A_{co}, $L_{c,total}$, d_c *and* V_k are 1~3

magnitude smaller than those from other parameters. Thus other parameters, $A_{e,cs}$, A_{ei}, d_e, $A_{c,cs}$, A_{ci}, valve map coefficients, compressor volumetric efficiency map coefficients, and isentropic efficiency map coefficients, were selected as the tuning parameters.

Selection of Transition Point

The focus of using the hybrid search algorithm is determining the amount of search performed by each algorithm. The transition point largely affects the robustness and efficiency of the hybrid method. An early transition point may decrease computation time but fail in finding the optimal solution, while a later transition point is more likely to find the optimal solution at the expense of large computation cost. Compromising between robustness and efficiency is a challenge when using this method. The principle of making this compromise is to guarantee that the convergence error is within the vicinity of the optimal solution and to ensure the computation cost is within the acceptable level.

To illustrate the effect of transition point, a comparative study was performed for different transition points. The cost function trajectory using only the GA is shown in Figure 4.14. Note that the time on the *x* axis is CPU time, which is different from the computation time. CPU time is larger than computation time since parallel processing is used in this simulation. The CPU time is about 600 s, and the relative error is 6.8%. In order to show the significance of using the hybrid method, a gradient search method is used after the GA stops, and the trajectory is shown in Figure 4.15 (a). The CPU time of this gradient method is about 1200 s, and the final relative error is 4.73%. The efficacy

of each optimization algorithm can be evaluated using the percent of time (*POT*) and percent of improvement (*POI*) spent on each parameter estimation stage (the stage refers to the GA or the gradient search algorithm). The *POT* is the time fraction spent on each optimization stage, and the *POI* is defined as:

$$POI = \frac{\varepsilon_k - \varepsilon_{k-1}}{\varepsilon_{norm} - \varepsilon_{final}} \times 100\% \tag{4.3}$$

where ε_{norm} is the relative error obtained by nominal parameters, ε_k and ε_{k-1} are the relative error at current and previous parameter estimation stage, and ε_{final} is the final relative error of the complete parameter estimation.

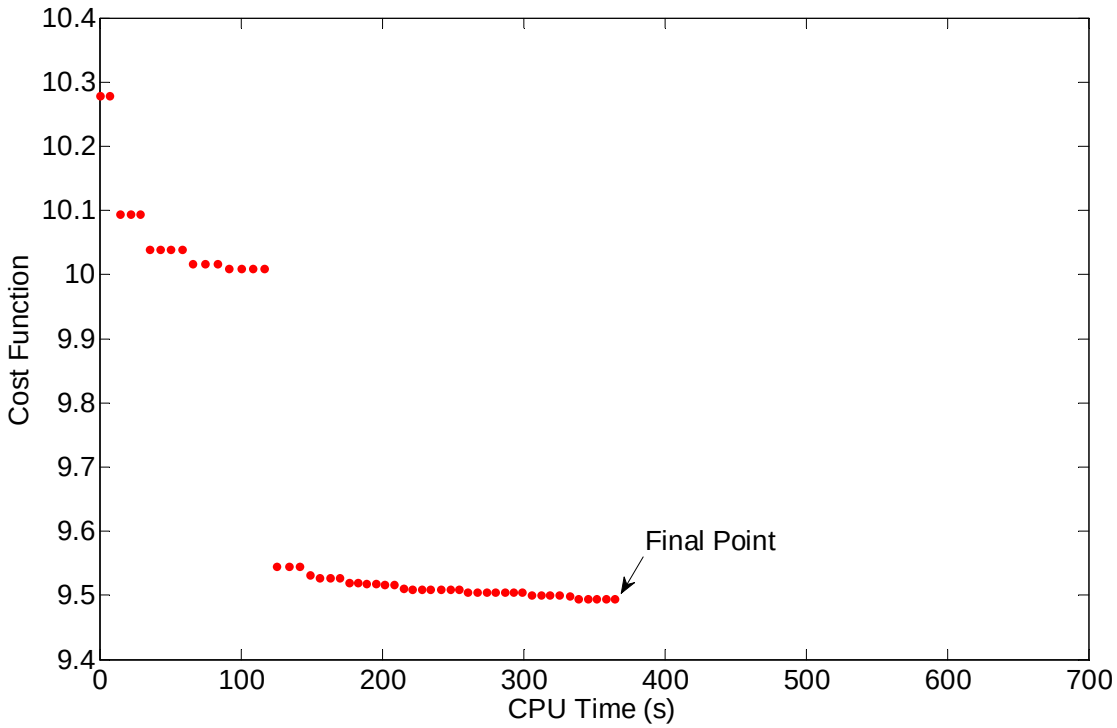


Figure 4.14: Objective function trajectory using only the GA.

The *POT* of the hybrid method is 57.4% and 42.6%, and the *POI* is 62.5% and 37.5% for each parameter estimation stage. Thus, the use of the gradient method largely improves the accuracy of the tuning results. The decision of using the gradient search method after the GA can be made according to the required tolerance and computation time. If the relative error and the computation time by using only the GA already meet the requirement, there is no need to tune the model using the gradient method. However, if there is the necessity of obtaining lower relative error within a shorter time, then the use of a hybrid method is necessary.

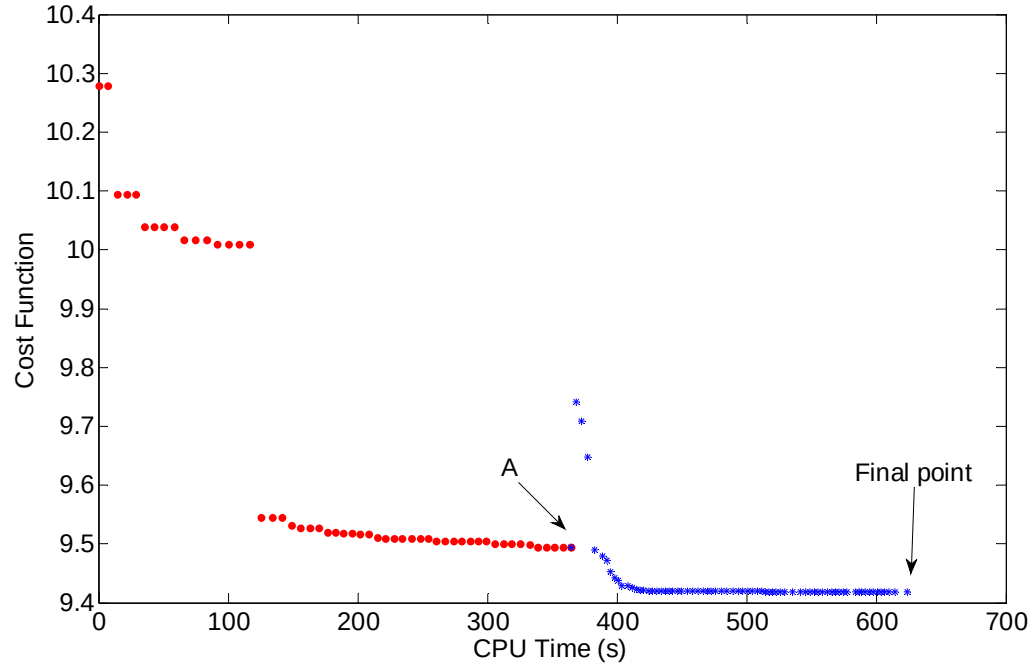
In this study, the transition point is selected such that after a given minimum number of generations, the algorithm transitions from the GA to the gradient search when the cumulative change in the cost function over the specified number of stall generations is less than the ideal tolerance. The cost function trajectories of three selected transition points are compared in Figure 4.15. Point A indicates a tuning trajectory for a full run of the GA using the default stop criteria in MATLAB. The transition criteria for point B and C are listed in Table 4.5. These were chosen to stop the GA at an earlier and later time. The state of each GA simulation is set to be the same to ensure the results are the comparable. This can be readily realized in MATALB by running GA with an output argument containing the current state, and then resetting the state to this value for all other GA runs.

Table 4.5: The definition of different transition points.

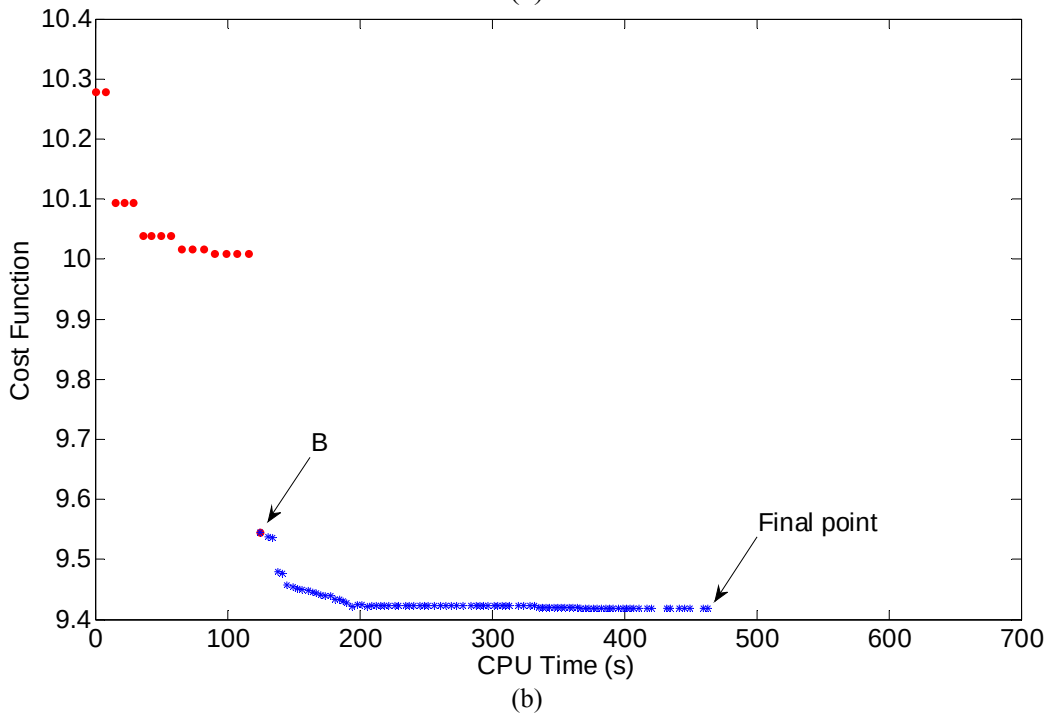
Settings	Point A	Point B	Point C
Minimum number of generations	-	30	10
Stall generations	-	5	5

Point B shows that a reasonable early transition point largely shortens the computation time as well as converges to the global optimum. Point C shows that an improper early transition point may not provide a good initial guess point for the gradient search method, thus causing the convergence to a local optimum (the value of the cost function is higher).

The trajectory of the optimization is problem dependent. Hence, a few preliminary tests are required to determine the transition point. However, these tests do not increase the total computation time since the trajectory can also be applied to other data sets and even similar models. Moreover, the stochastic algorithm is stopped prematurely, and better tuning results are always ensured compared to using local search method solely. Another method to avoid preliminary tests is to use experience. Generally, the minimum number of generations can be set to 50, and the stall generations can be set to 10 with a required tolerance of 1e-6. In most cases, this stop criterion ensures the convergence to a global minimum and can be used to locate the transition point during the preliminary tests.

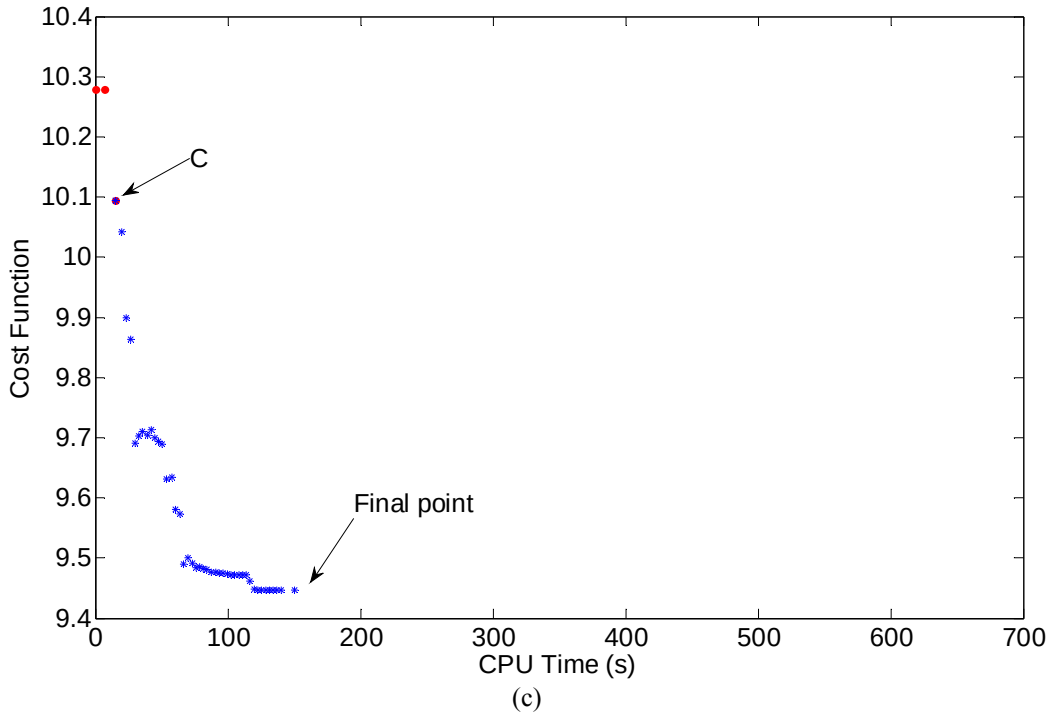


(a)



(b)

Figure 4.15: Objective function trajectory for different transition points.



(c)

Figure 4.15: Continued.

Determination of Weighting Factor

The weighting factor for each output in the objective function (3.16) is determined as follows. For the evaporator, pressure generally has a more significant influence because it greatly affects other outputs, i.e. superheat and mass flow rate. On the other hand, the outlet external fluid temperature has little significance since it is rarely used in control design. Thus, the weighting factor for the pressure should be larger than other outputs, and it should be smaller for the outlet external fluid temperature (or set to zero) if matching it with the data is not significant. Figure 4.16 compares the tuning results when (1) the weighting factor for the pressure is the same as other outputs;

(2) the weighting factor for the pressure is 2.5 times larger than other outputs; (3) the weighting factor for the pressure is 10 times larger than other outputs;

These results show that the tuning results improve when the weighting factor for the pressure is 2.5 times larger than other outputs, shown by the green line. The outlet refrigerant and external fluid temperatures are not a close match to the experimental data because the weighting factor is 5 times smaller than other outputs with the exception of pressure, which is larger than other outputs. The blue line signifies when the weighting factor for the pressure is the same as other outputs (except the outlet external fluid and refrigerant temperature), the results match poorly. When the weighting factor for the pressure is 10, the tuning results are similar to the green line (weighting factor for the pressure is 2.5). Therefore, it is suggested that the weighting factor for the most important outputs be set 2 to 3 times larger than the other outputs and that the least important outputs have a weighting factor set 5 times smaller or zero.

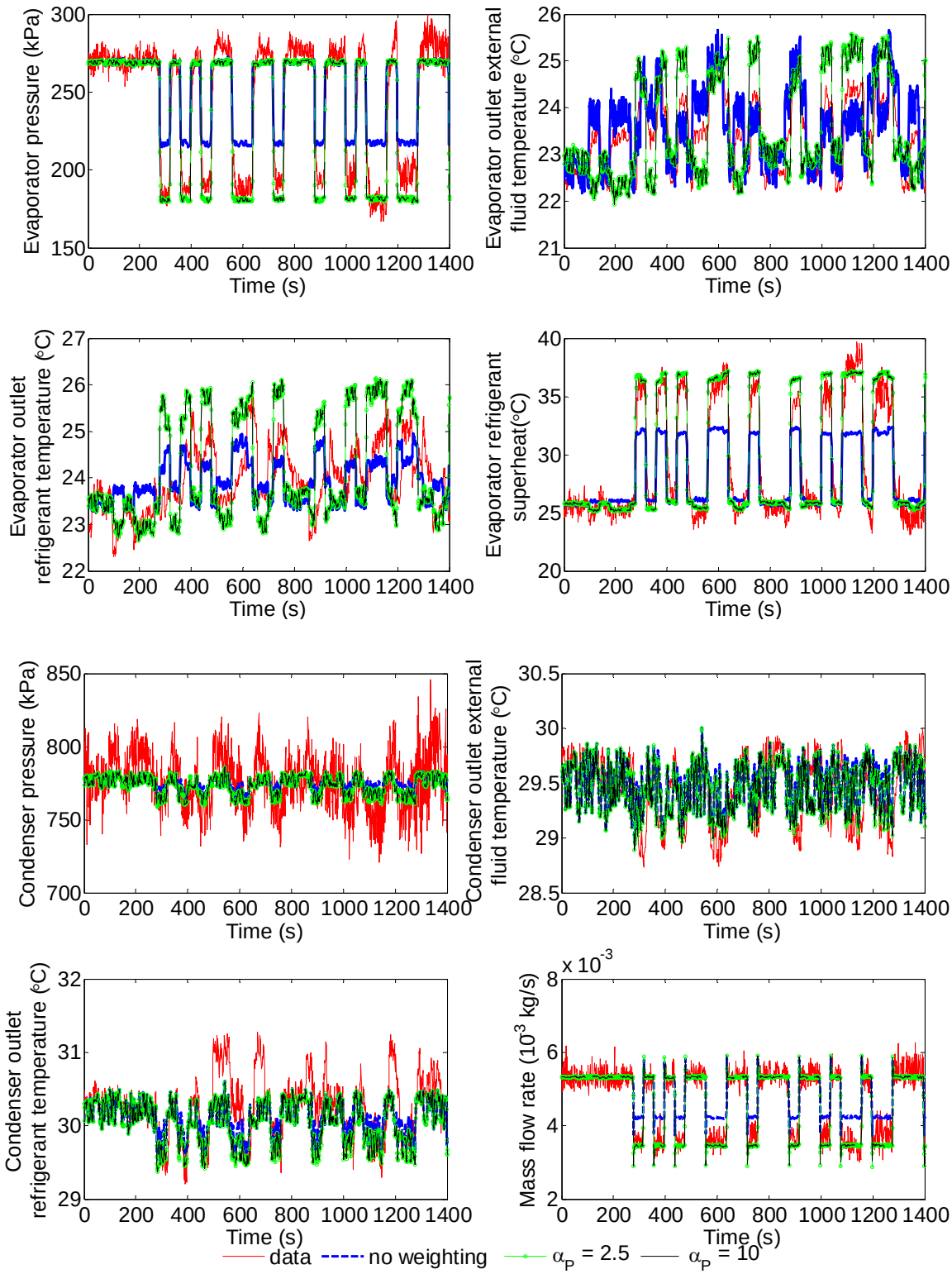


Figure 4.16: Tuning results with different weighting factors.

Parameter Tuning and Validation Using Single Data Set

The tuning results of the complete HVAC system model using one data set are given in Figure 4.17 and Table 4.3. The red lines represent the experimental data, and the blue lines represent the outputs using the final tuned parameters. The relative error of tuned outputs is 3.58%, which is smaller than that of the nominal parameters (11.5%). Thus, the predicted accuracy of the tuned model is improved. The simulation time is longer than tuning a single heat exchanger, which is reasonable since the complete HVAC model contains more parameters, more outputs, and more calculations. The proposed parameter tuning method drives the predicted outputs toward the data. These figures prove that the proposed approach is effective in improving the capability of finding the optimal parameters.

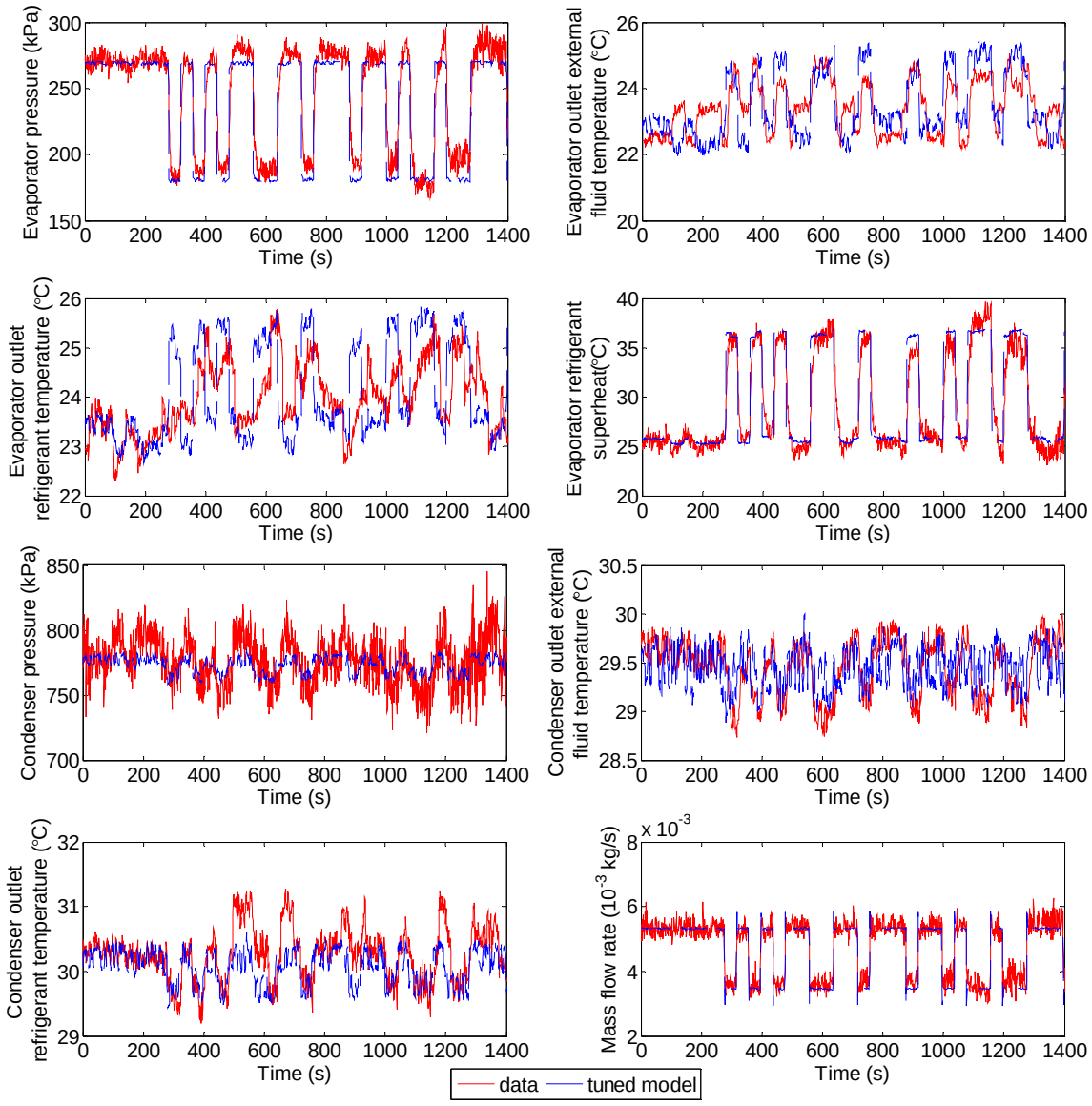


Figure 4.17: Parameter tuning and validation results using single data set.

Parameter Tuning and Validation Using Multiple Data Sets with Different Operating Conditions

In order to show that the proposed method has a good robustness, parameter tuning is also performed on the same HVAC model using two different data sets from

the same experimental system, each with different operating conditions. The tuning parameters are the same as that tuned in single data validation, given in Table 4.3. Among these parameters, the physical ones can be assumed the same or different for the two data sets. Both cases were performed. The tuning parameters EXV and volumetric efficiency coefficients were set differently in both cases. This is because the prediction of mass flow rates, calculated by the coefficients, can be adjusted differently for the two data sets. The objective function described in Equation (3.16) is modified to include both field data sets.

The results in Figure 4.18 and Figure 4.19 are the tuning results when the tuning parameters are assumed the same for the two data sets. Figure 4.20 and Figure 4.21 are the results when the tuning parameters are different for the two data sets. The tuned errors and time can be found in Table 4.3. Note the large savings in computation time due to the proposed method described in Chapter III. The results obtained when assuming different tuning parameters between data sets are slightly better than those obtained when the parameters for both data sets are assumed to be the same. Therefore, the tuning parameters for the two data sets are suggested to be the same when the accuracy requirement is relaxed. Further discussions about tuning using multiple data sets can be found in Chapter V.

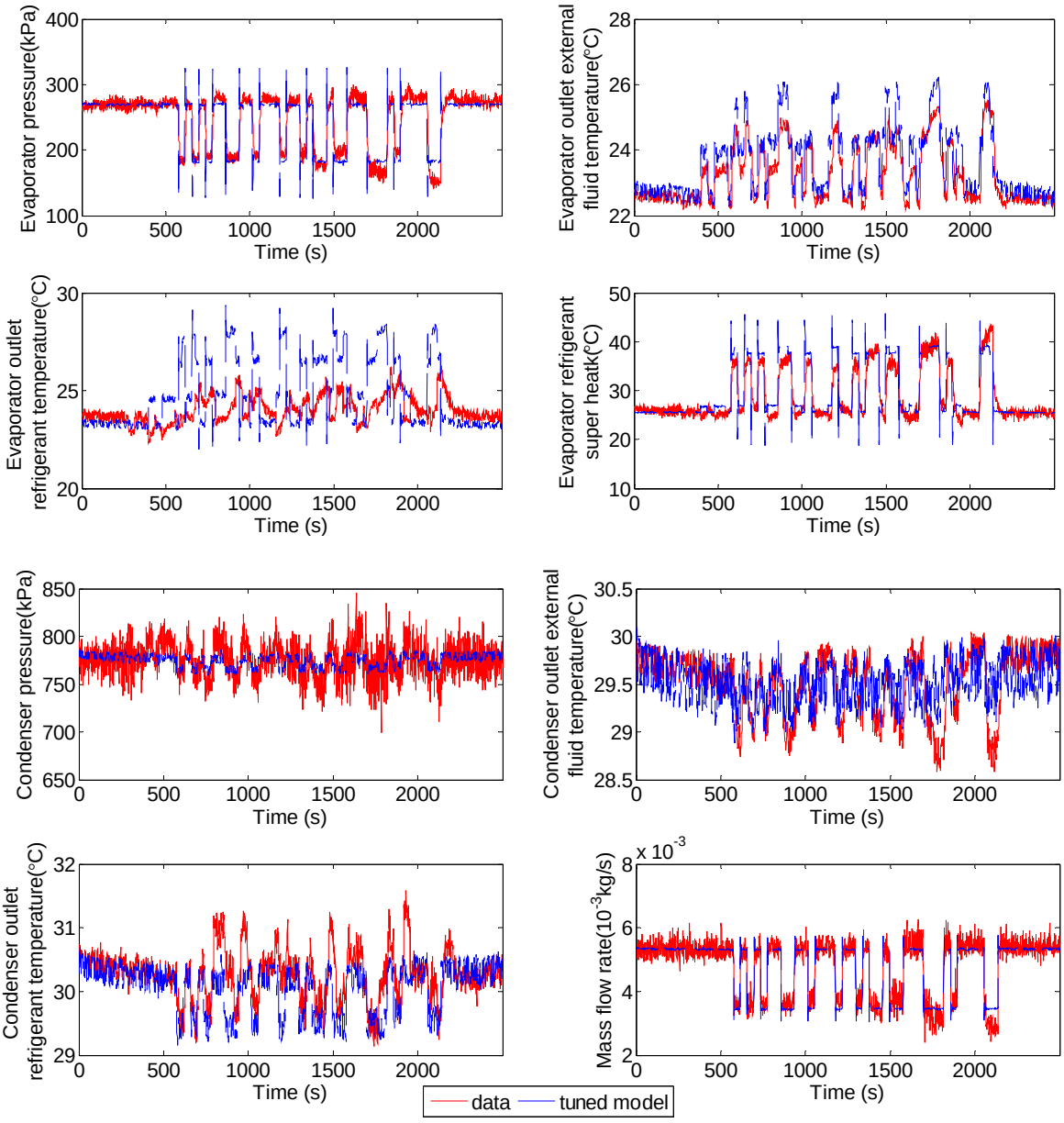


Figure 4.18: Parameter tuning and validation results using multiple data sets with the same tuning parameters (data set 1).

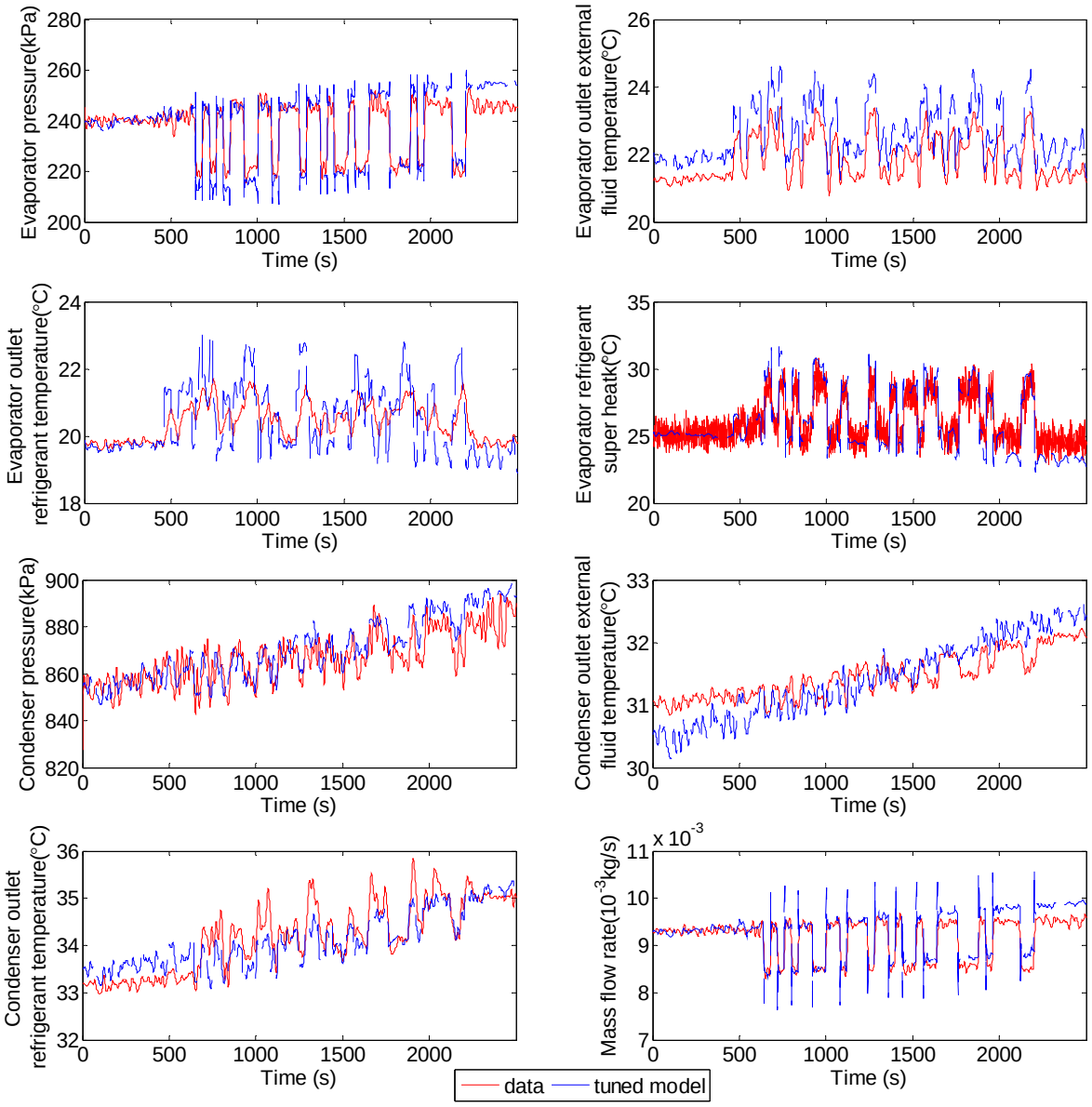


Figure 4.19: Parameter tuning and validation results using multiple data sets with the same tuning parameters (data set 2).

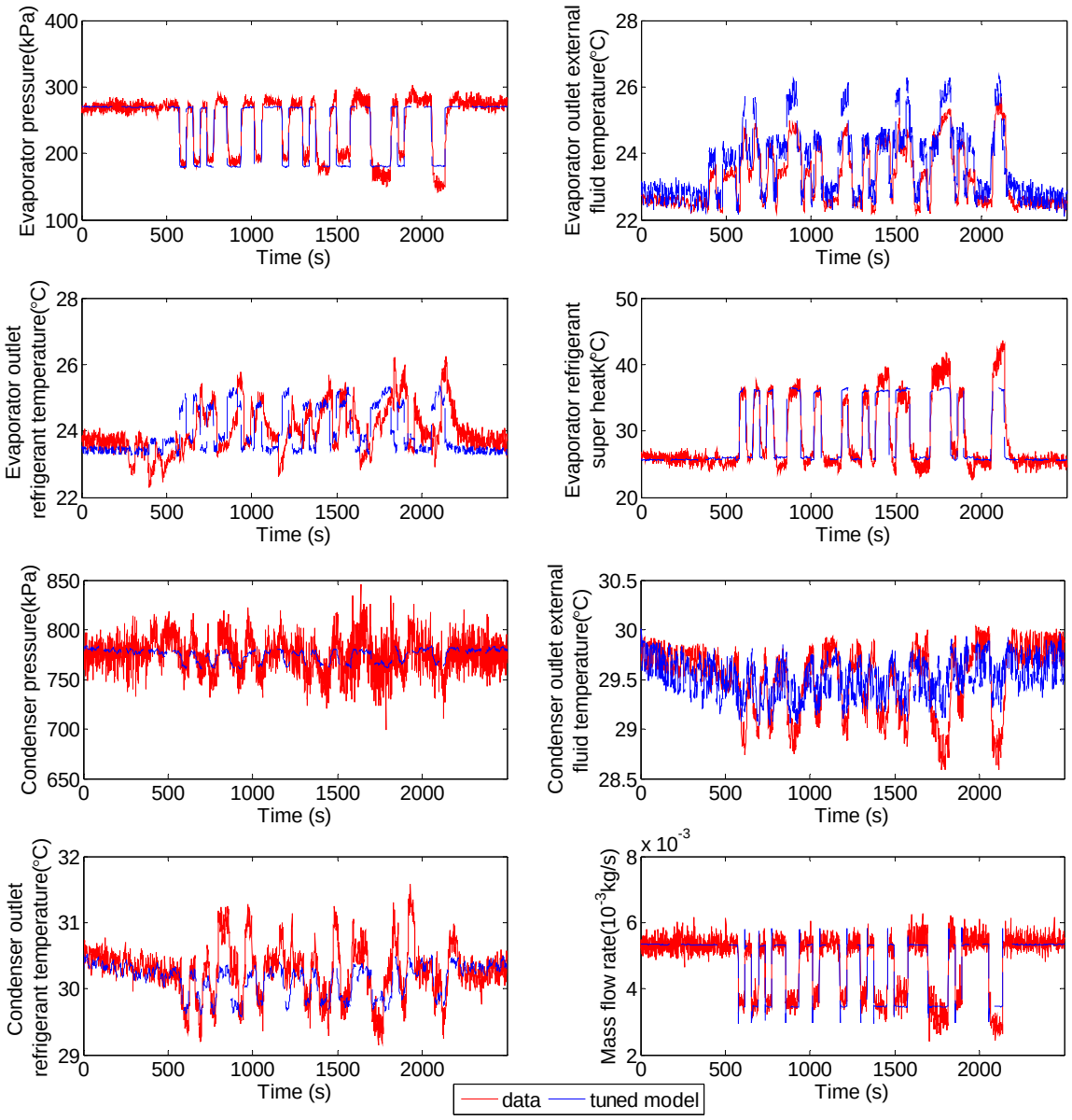


Figure 4.20: Parameter tuning and validation results using multiple data sets with different tuning parameters (data set 1).

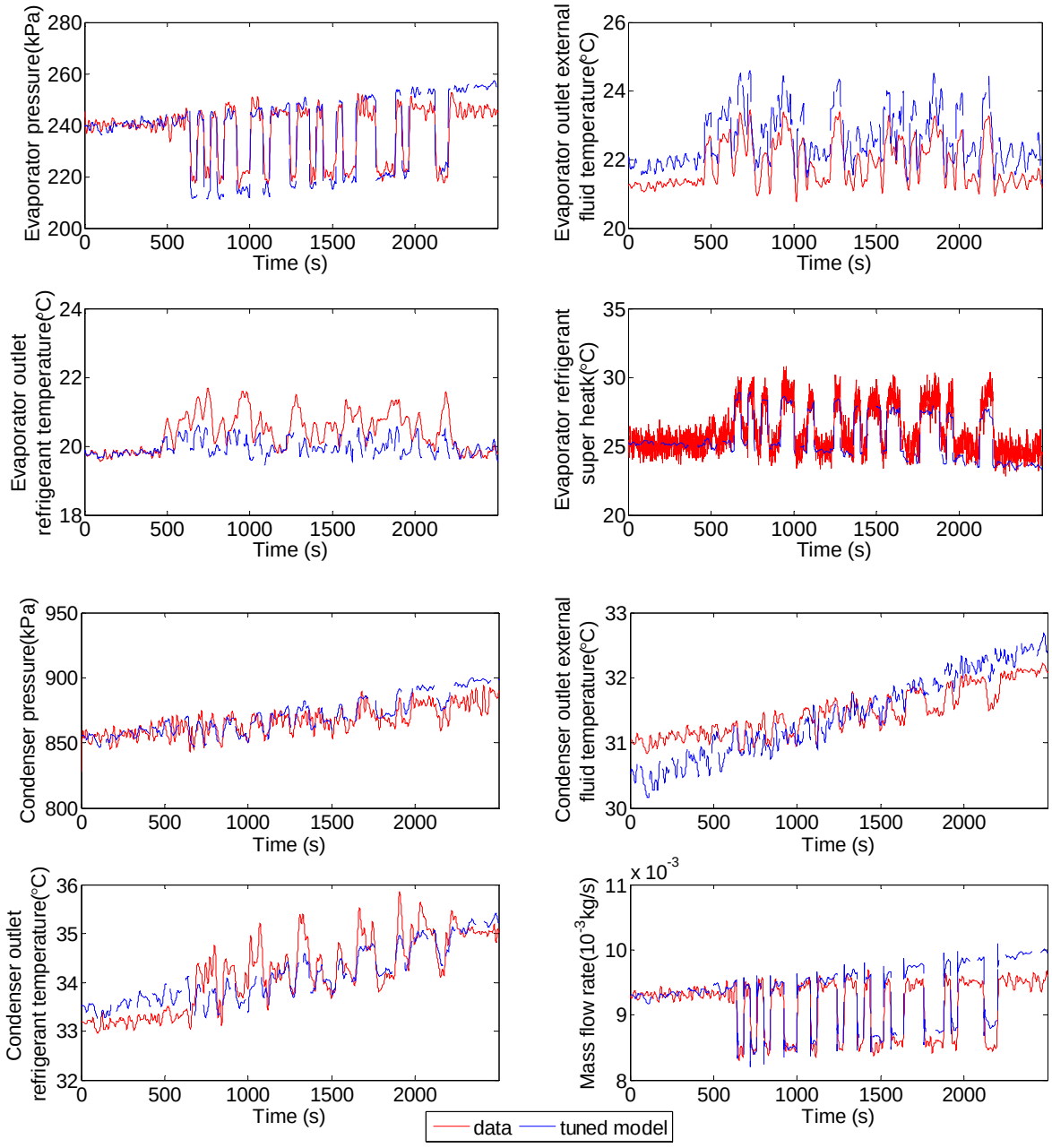


Figure 4.21: Parameter tuning and validation results using multiple data sets with different tuning parameters (data set 2).

Cross Validation

The tuned solution obtained using the single data set is validated using another data set from the same experimental system on both linear and nonlinear models. Figure 4.22 and Figure 4.23 are the validation results. The relative errors of the outputs shown in these figures are 7.04% and 14.3% for linear and nonlinear model, respectively. The validation results indicate that, with tuned parameters, both the linear model and nonlinear model are effective in predicting the outputs, implying a good robustness of the model.

The validation results using the linear model are better than that of the nonlinear model, since the proposed parameter estimation and tuning approach is based on the linear model. However, the application of the tuned solution on the nonlinear model shows that the predicted outputs are also close to the experimental data, signifying the proposed parameter estimation and tuning approach is highly effective in improving the performance of the nonlinear model. On the other hand, if the parameter tuning and validation is performed using the nonlinear model instead of the discrete-time linearized model used in the proposed parameter tuning method, the computation cost will be very large.

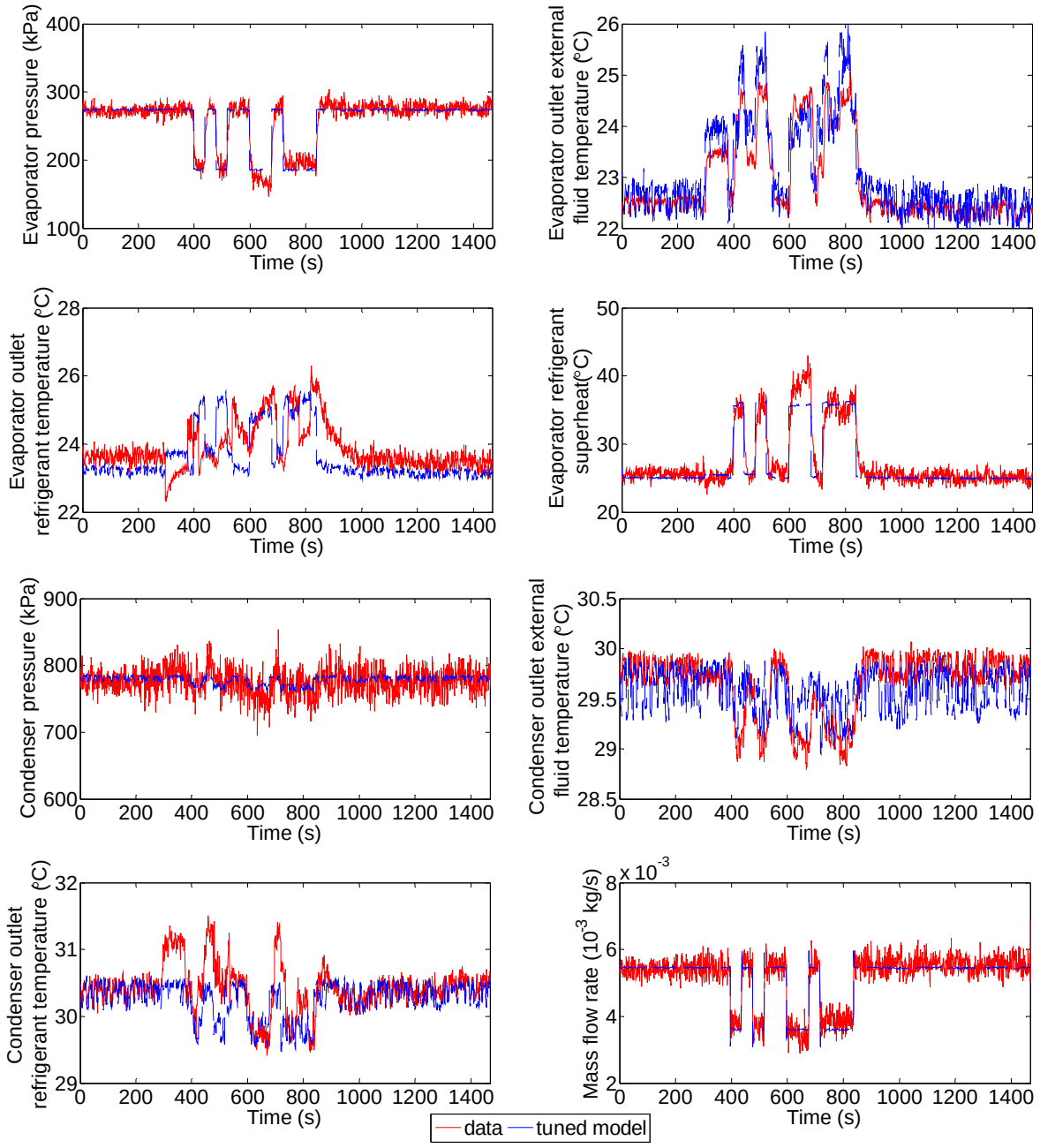


Figure 4.22: Cross validation of the tuned solution on linearized model.

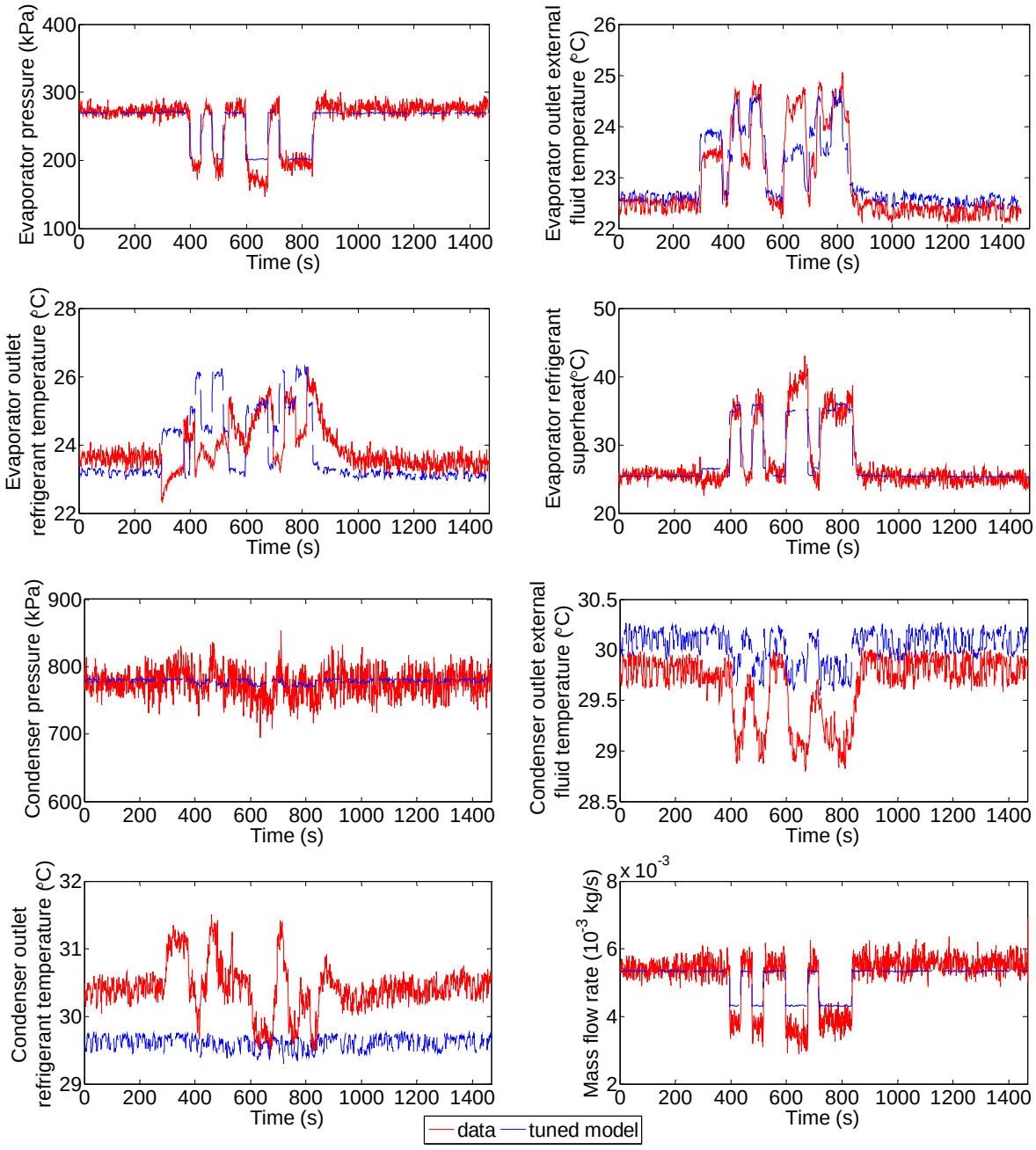


Figure 4.23: Cross validation of the tuned solution on nonlinear model.

Additionally, if the error of the cross validation on nonlinear model is higher than the required tolerance, the tuning results can be improved by directly tuning the

nonlinear model using the gradient method with the initial guess value obtained from the parameter tuning of the discrete-time linearized model. Table 4.6 shows the *POT* and corresponding *POI* spent on each parameter estimation stage. Stage 1 and 2 illustrate the parameter tuning and validation using hybrid optimization method. Stage 3 is the additional parameter tuning process using gradient search method. This stage is only performed if further improvement of tuning results is required.

Table 4.6 shows that the *POT* spent on the third stage is extremely large, but the corresponding improvement is small. Therefore, the slight improvement of tuning the nonlinear model may not be cost-efficient. The decision of applying the third parameter tuning stage is based on the tolerance requirement.

Table 4.6: Comparison of parameter tuning performed during each optimization algorithm.

Variable	Parameter tuning of discrete-time linearized model		Parameter tuning of nonlinear model (optional tuning)
	GA (Stage 1)	Gradient method (Stage 2)	Gradient method (Stage 3)
POT (%)	10.4	5.8	83.8
POI (%)	82.2	17.6	0.2

CHAPTER V

APPLICATION TO EMERSON VARIABLE SPEED HEAT PUMP MODEL

The proposed parameter tuning and validation method is also applied on a real variable speed heat pump system model established for Emerson Climate Technologies. The heat exchangers in the heat pump model were modeled using finite control volume method. As mentioned in Chapter I, finite control volume method can predict the salient dynamics of multiple fluid phase heat exchangers more accurately than moving boundary method, but requires high computation cost. This feature poses a great challenge to model validation of finite control volume heat exchanger models. It is also a challenge to prove the effectiveness of the proposed parameter tuning and validation method.

The variable speed heat pump system consists of an evaporator, an electronic expansion valve, a thermostatic expansion valve, a condenser, a fan, a blower, a compressor and a reversing valve. It is capable of switching between heating and cooling mode through the reversing valve. The configuration of this heat pump system can be found in Figure 5.1.

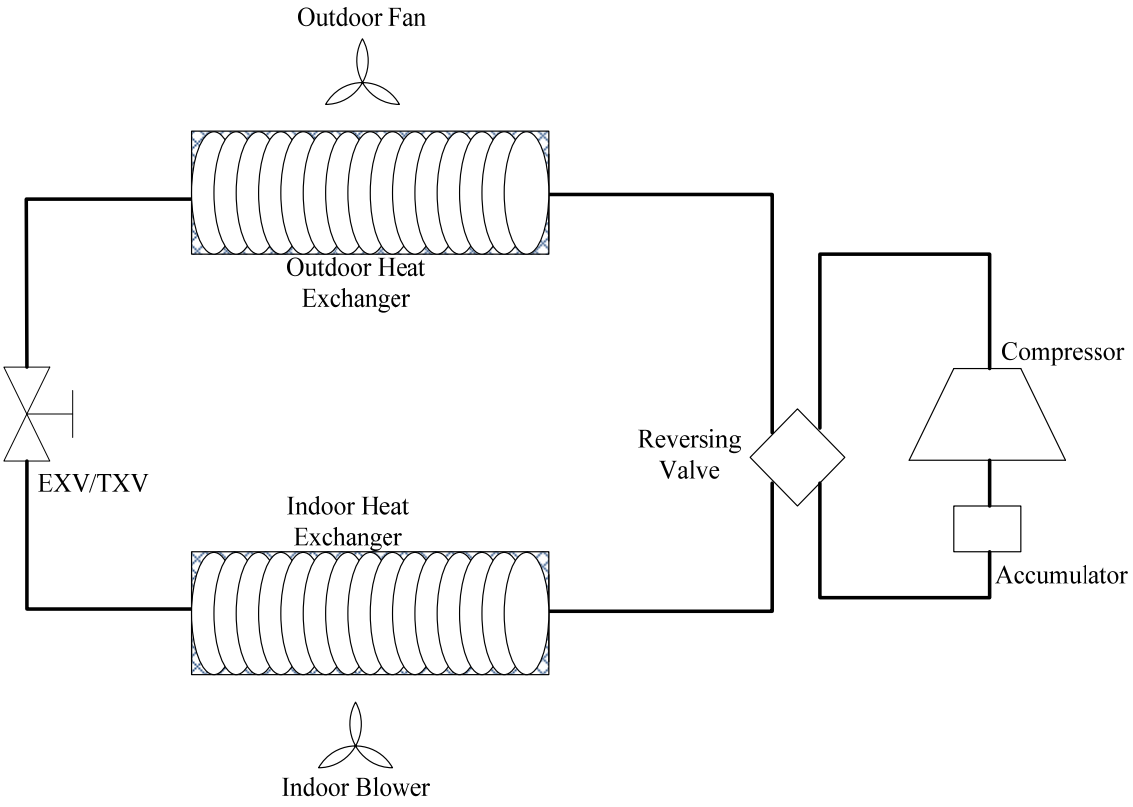


Figure 5.1: Variable speed heat pump system.

This chapter first introduces the variable speed heat pump system configuration. Two different systems are presented: Rheem 5T and Coleman 5T heat pump system. 5T means the cooling capacity of the heat pump system is 5 tons of refrigeration. Tons of refrigeration refers to a system's capacity of freezing 1 ton of 0 ºC liquid water to 0 ºC ice. Both Rheem 5T and Coleman 5T systems are residential heat pumps, and are widely used in the residential houses in the United States.

Second, the field data sets from Emerson are presented. These data were recorded under practical situations for both Rheem 5T and Coleman 5T system. They

contain numerous large and hard transients. These data sets are used to validate the finite control volume heat exchanger model developed for Emerson Climate Technologies.

Third, the pseudo-steady state data generated from the field data sets are used to validate the FCV model. The approach to generate pseudo-steady state data from Rheem and Coleman 5T heat pump field data sets is described and discussed. The tuning and validation results prove the efficacy of the proposed parameter tuning method on the FCV models.

Finally, the field data sets are used to validate the FCV heat pump model in detail. These validations include the use of several field data sets. Validation for component models, such as FCV evaporator and condenser, are presented first. Complete heat pump model validation using field data is then discussed, followed by simultaneous parameter tuning and validation using multiple data sets.

Emerson Heat Pump System Configuration

Two heat pump systems, Rheem 5T and Coleman 5T systems from Emerson are modeled. These two systems are widely used in residential houses. Rheem 5T system operates both heating and cooling modes, while Coleman 5T system only operates during the summer.

Figure 5.2 and Figure 5.3 show the configuration of Rheem 5T and Coleman 5T system under heating and cooling mode, respectively. The operating mode is determined by the reversing valve. During heating mode (Figure 5.2), the indoor heat exchanger serves as the condenser, and the outdoor heat exchanger serves as the evaporator. An

accumulator is also connected with the outdoor exchanger. An electronic expansion valve is used in this mode, thus the thermostatic expansion valve is bypassed. During cooling mode (Figure 5.3), the indoor heat exchanger serves as the evaporator, and the outdoor heat exchanger serves as the condenser. The accumulator is connected with the indoor exchanger. The thermostatic expansion valve is used in this mode, thus the electronic expansion valve is bypassed.

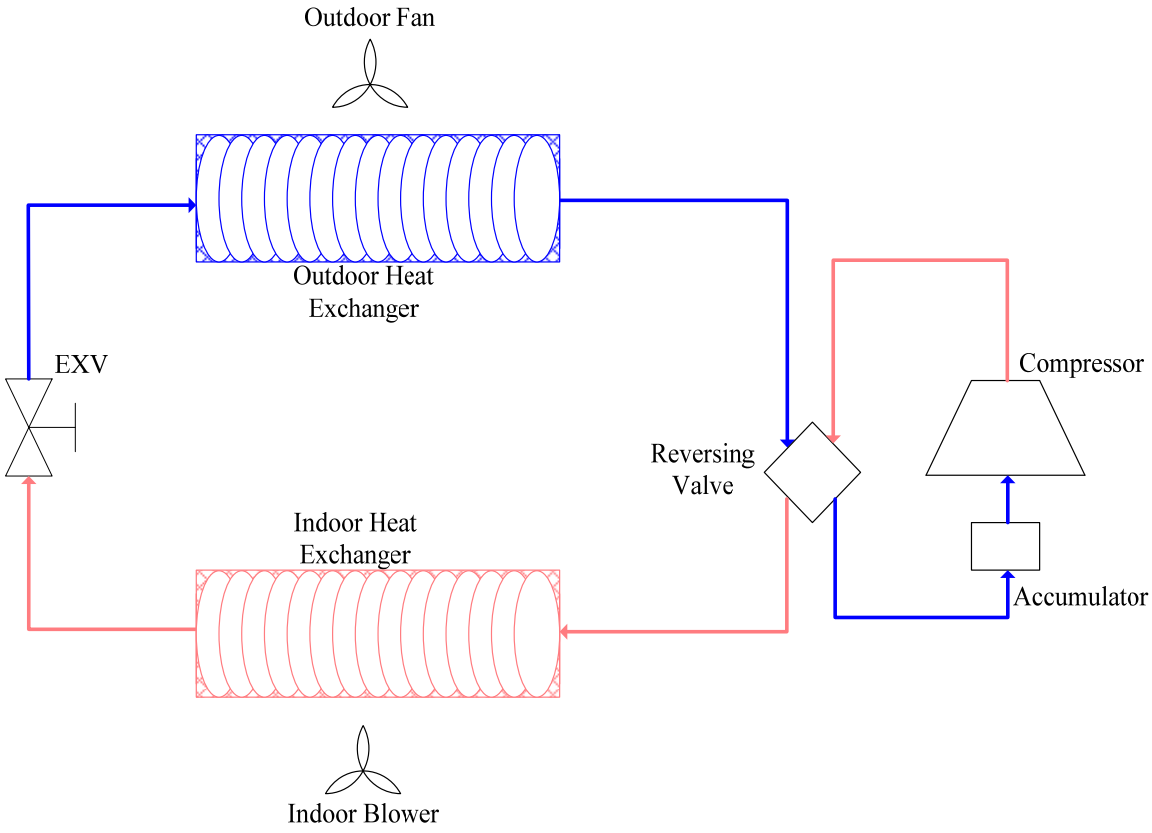


Figure 5.2: Variable speed heat pump system configuration (heating mode).

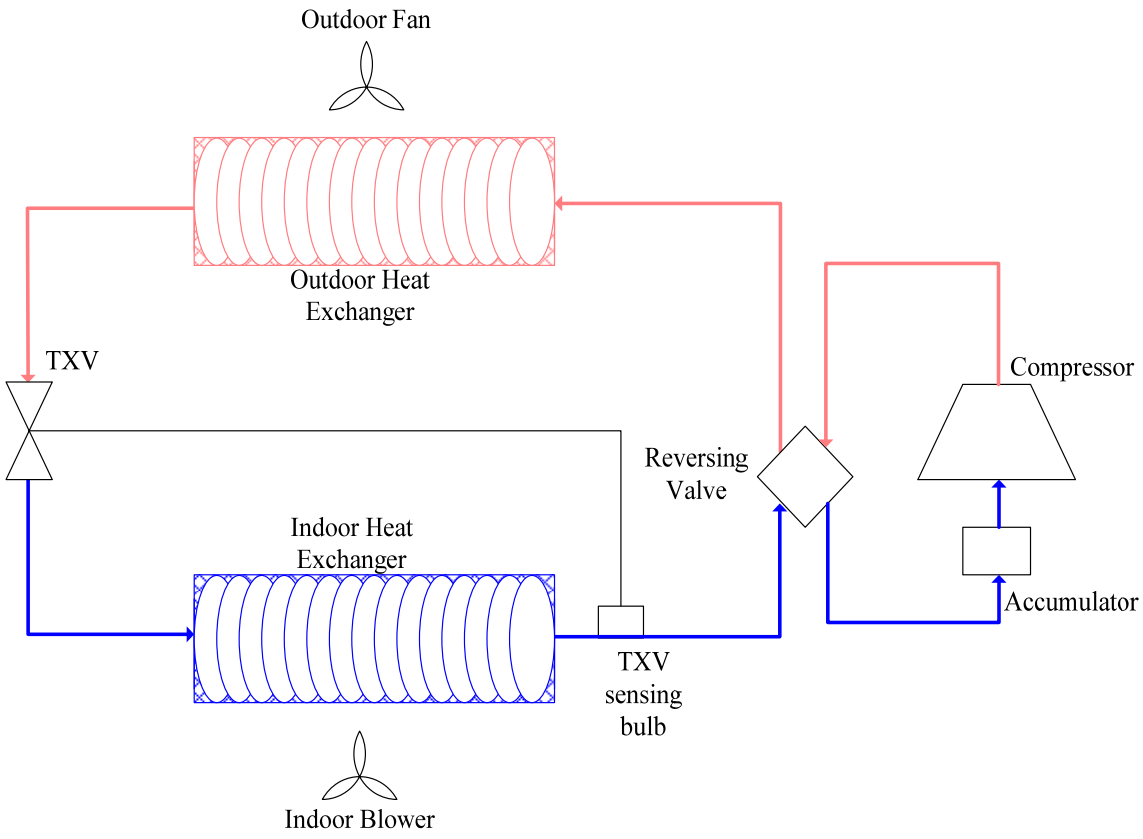


Figure 5.3: Variable speed heat pump system configuration (cooling mode).

Field Data

Hundreds of field data sets from Rheem 5T and Coleman 5T heat pump systems are provided by Emerson. These data are daily data, and they were recorded from the Rheem 5T system operated between year 2011 and 2013 in heating mode, and in 2011 in cooling mode. The data recorded from Coleman 5T system operated in 2012 in cooling mode are also used. The sample time of these data is 10 seconds. The following figures

show two typical data sets recorded during a whole day for Rheem 5T and Coleman 5T heat pump system.

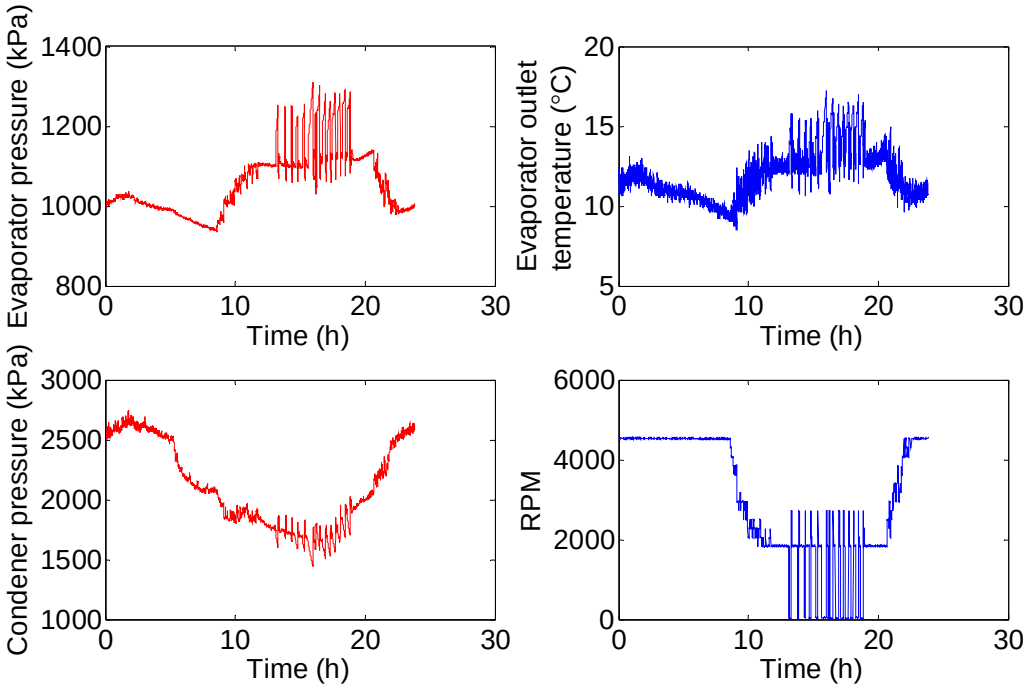


Figure 5.4: Rheem 5T field cooling data 05/30/2011.

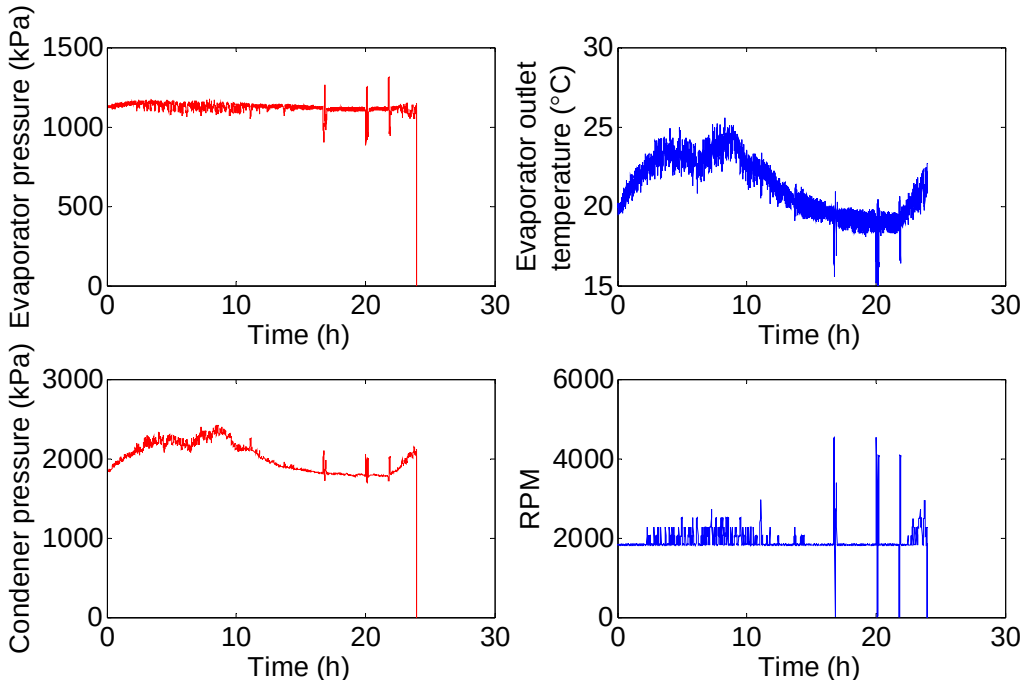


Figure 5.5: Coleman 5T field cooling data 07/15/2012.

These plots show that the field data contain lots of large and hard transients. There are data points where the compressor starts up and shuts down, posing a great challenge to simulation. When the hard transients happen, the assumptions made during modeling may not be applicable. The simulation of hard transients may also cause numerical singularities. These problems further cause the difficulties in parameter tuning using filed data sets, because there are more large and hard transients that need to be matched with data simultaneously.

Pseudo-Steady State Data Generation from Field Test Data

In order to perform steady state data validation for the vapor compression system, steady state data must be extracted from the field data sets. Since the field data sets contain too many drastic transients, steady state requirements should be carefully established. In this dissertation, the requirements for a range of data to be steady state include: (1) the time derivatives of compressor speed, inlet/outlet air temperatures and mass flow rates for the heat exchanger, and heat exchanger pressure are within a tolerance value for a number of consecutive points, shown in Table 5.1; (2) superheat, subcool, pressures and air mass flow rates are nonnegative to disregard impractical situations.

When all the tolerance requirements are simultaneously met, the range of the transient data can be considered as under pseudo-steady state. An example of pseudo-steady state range for the evaporator pressure is shown in Figure 5.6.

Table 5.1: Time derivative tolerances for pseudo-steady state data generation.

Time derivative	$\dot{rpm}$	$\dot{T}_{e,ao}, \dot{T}_{e,ai}, \dot{T}_{c,ao}, \dot{T}_{c,ai}$	$\dot{m}_{e,air}, \dot{m}_{c,air}$	$\dot{P}_e, \dot{P}_c$
Number of consecutive points		50, 80 or 100		
Tolerance value	1 rpm/s	0.1 ºC/s	0.01 kg/s^2	1 kPa/s

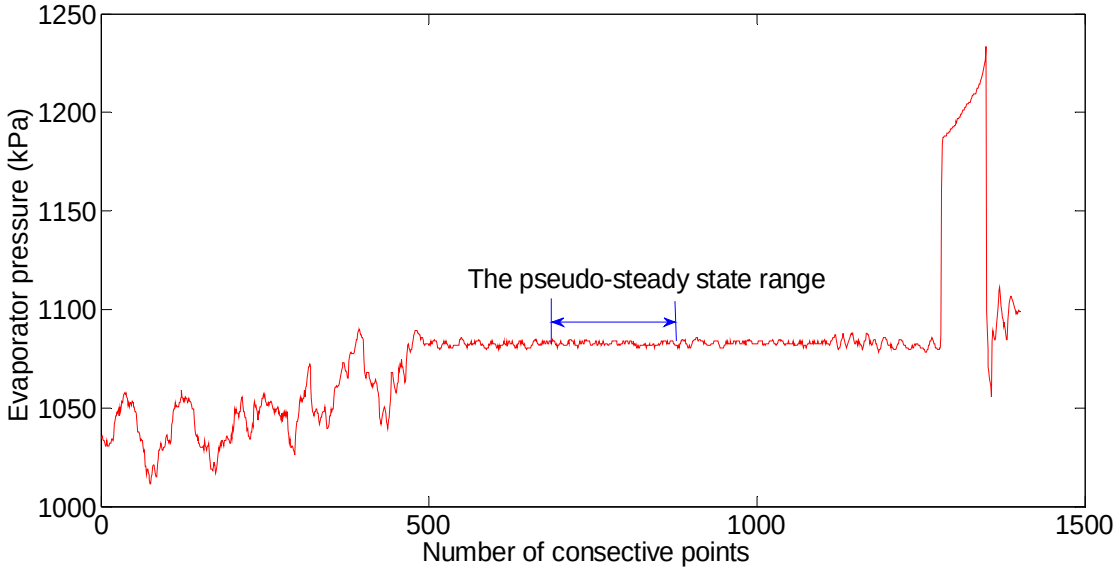


Figure 5.6: A pseudo-steady range for evaporator pressure.

Pseudo-Steady Data

For Rheem 5T System, pseudo-steady state data were generated from Rheem 5T Cooling and Heating field data sets. Figure 5.7 and Figure 5.8 are some pseudo-steady state data points. The horizontal coordinate represents the value of one variable, and the vertical coordinate represents the frequency of this value in the steady state data range. Similarly, pseudo-steady state data were also generated from Coleman 5T cooling field data sets. Figure 5.9 shows some pseudo-steady state data points.

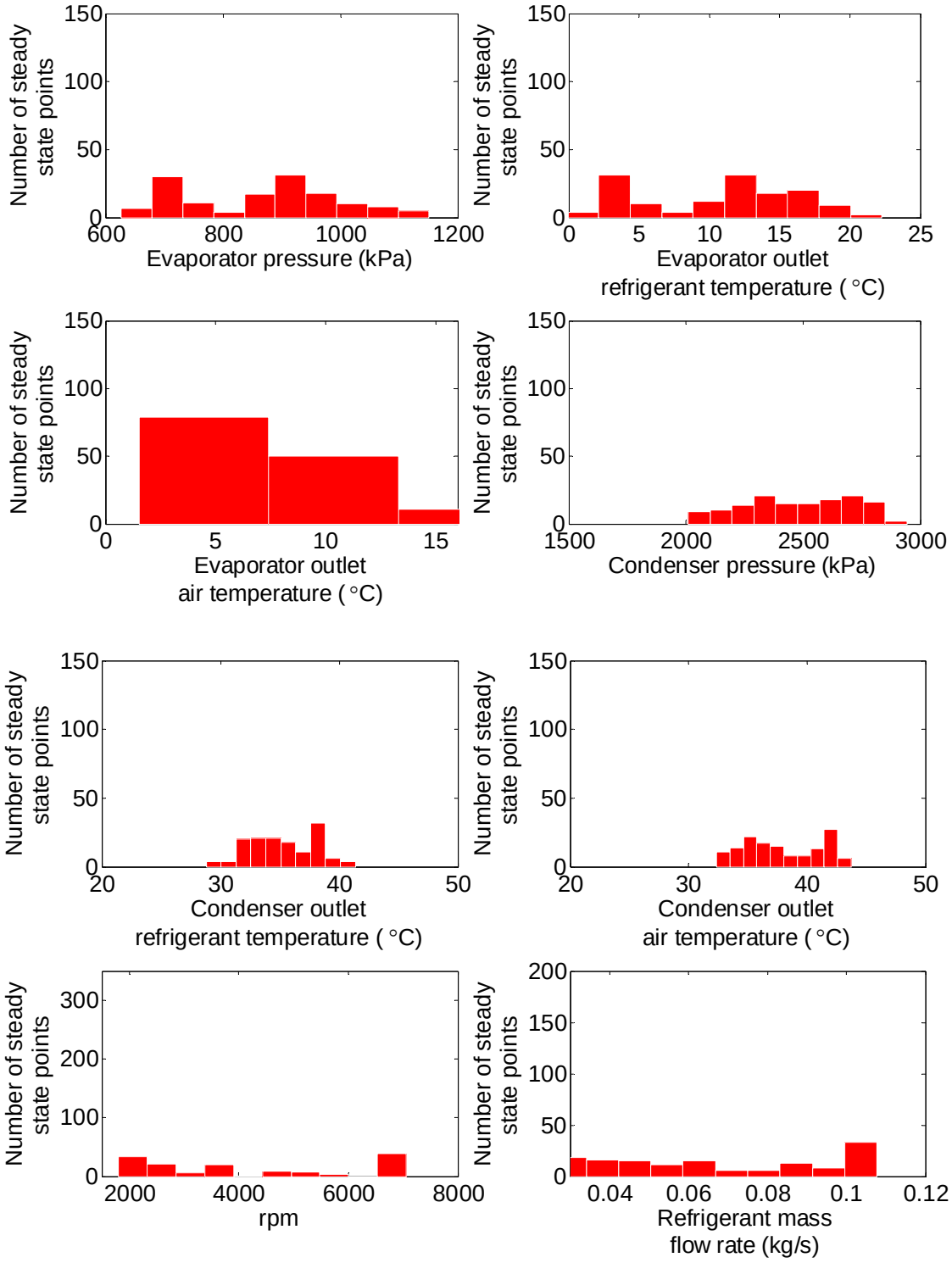


Figure 5.7: Rheem 5T heating pseudo-steady state data.

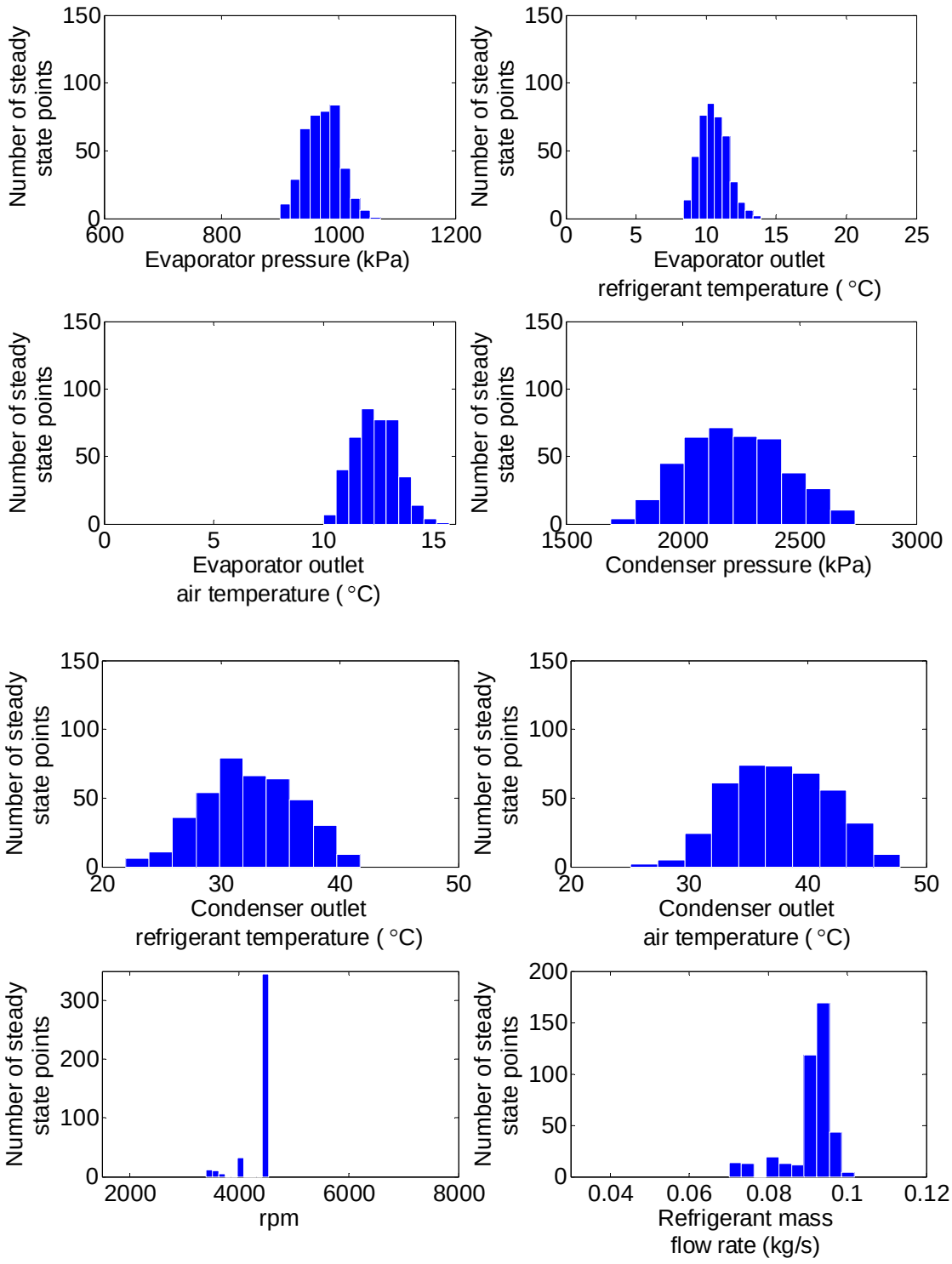


Figure 5.8: Rheem 5T cooling pseudo-steady state data.

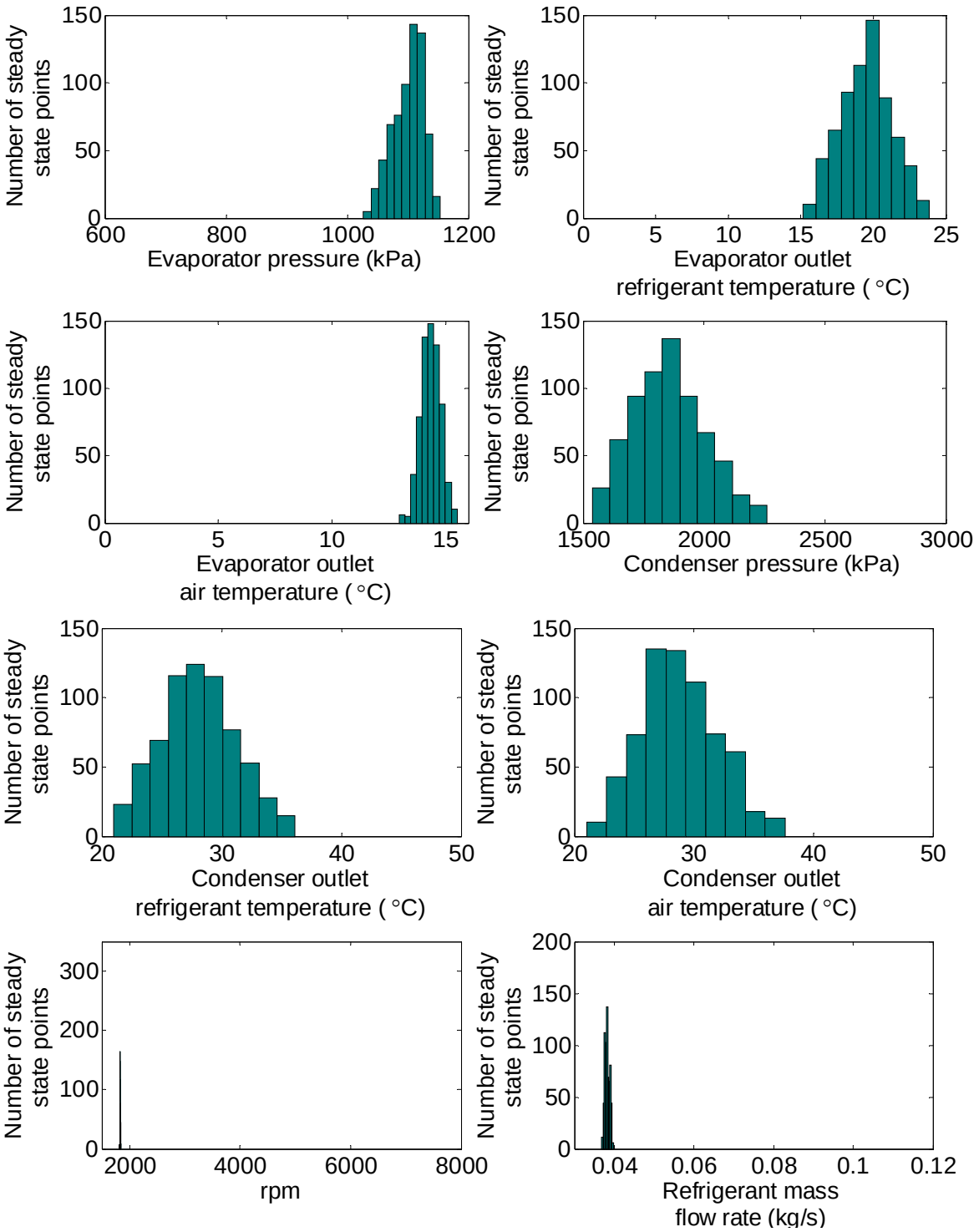


Figure 5.9: Coleman 5T cooling pseudo-steady state data.

Expansion Valves and Compressor Parameters Generated from Pseudo-Steady Data

The two TXV coefficients in Equation (2.25) are from pseudo-steady state data using numerical fitting. The following figures are the comparison between the mass flow rate from the data and that obtained using the generated parameters. The horizontal coordinate represents the pseudo-steady state data, and the vertical coordinate represents the value predicted by the generated TXV parameters.

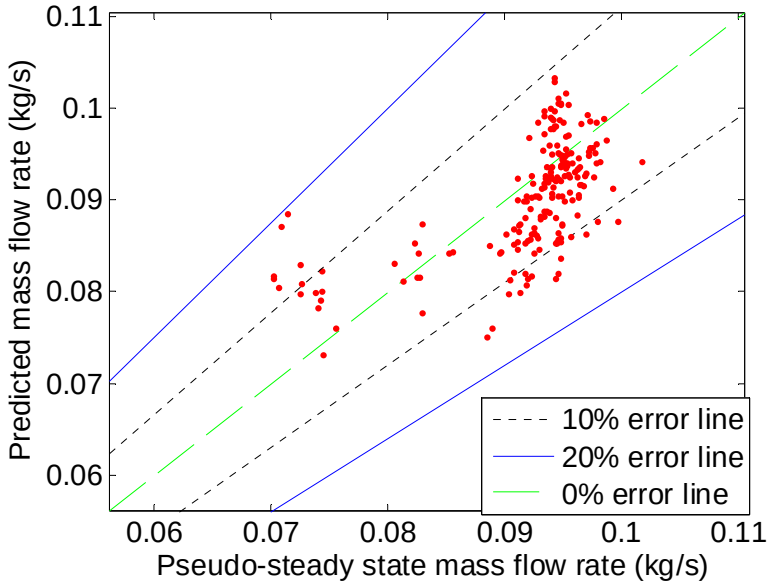


Figure 5.10: TXV mass flow rate comparison using generated parameters (Rheem 5T pseudo-steady state cooling data).

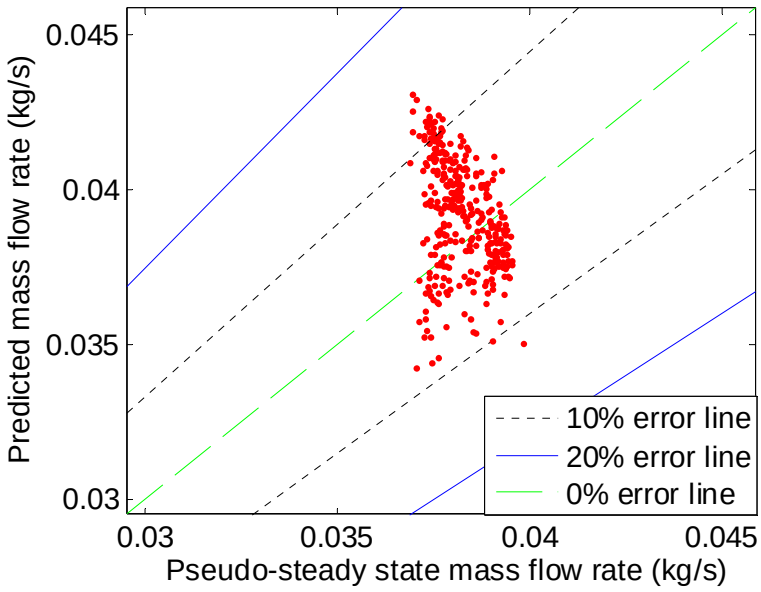


Figure 5.11: TXV mass flow rate comparison using generated parameters (Coleman 5T pseudo-steady state cooling data).

For the electronic expansion valve, the map of lumped discharge coefficient and valve area was generated. The following figure shows the comparison between the mass flow rate from the data and that predicted by the map. The horizontal coordinate represents the pseudo-steady state data, and the vertical coordinate represents the value predicted by the generated EXV maps. The results indicate that the map can accurately predict the mass flow rate.

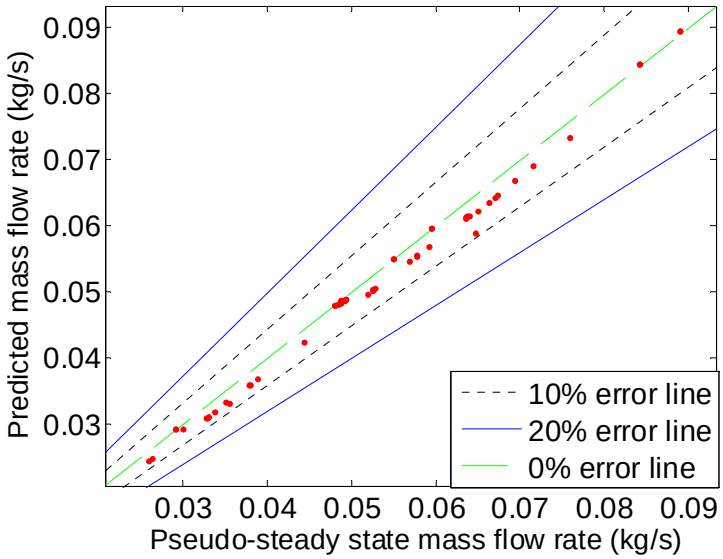


Figure 5.12: EXV mass flow rate comparison using generated parameters (Rheem 5T pseudo-steady state heating data).

For the compressor, volumetric efficiency and adiabatic efficiency as functions of pressure ratio and compressor speed were generated from the pseudo-steady state data. The following figures show the comparison between the mass flow rates and compressor power from the data and those predicted by the map. The horizontal coordinate represents the pseudo-steady state data, and the vertical coordinate represents the value predicted by the generated compressor maps. The results indicate that the map can accurately predict the mass flow rate as well as the compressor power.

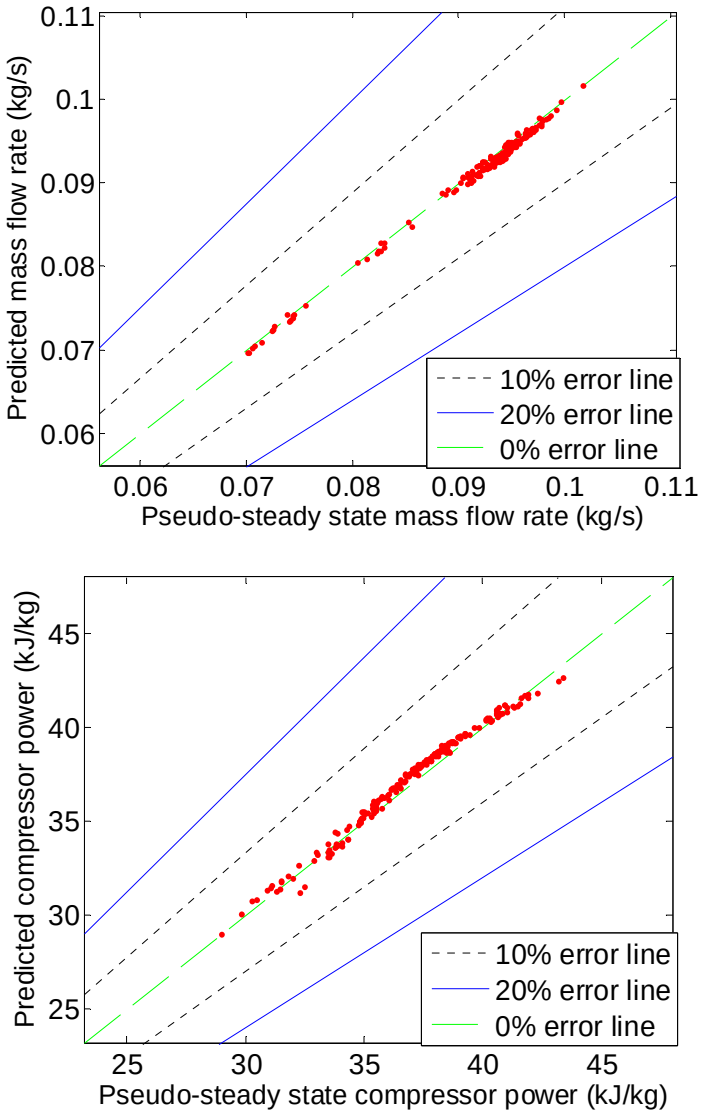


Figure 5.13: Compressor mass flow rate and power comparison using generated parameters (Rheem 5T pseudo-steady state cooling data).

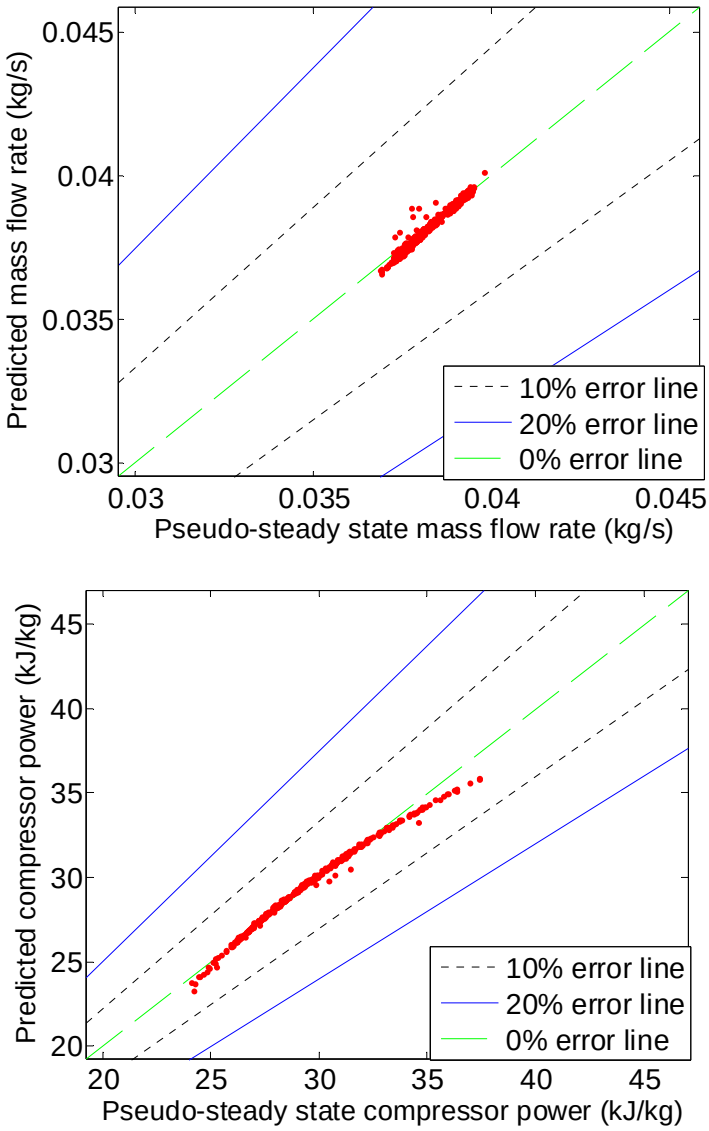


Figure 5.14: Compressor mass flow rate and power comparison using generated parameters (Coleman 5T pseudo-steady state cooling data).

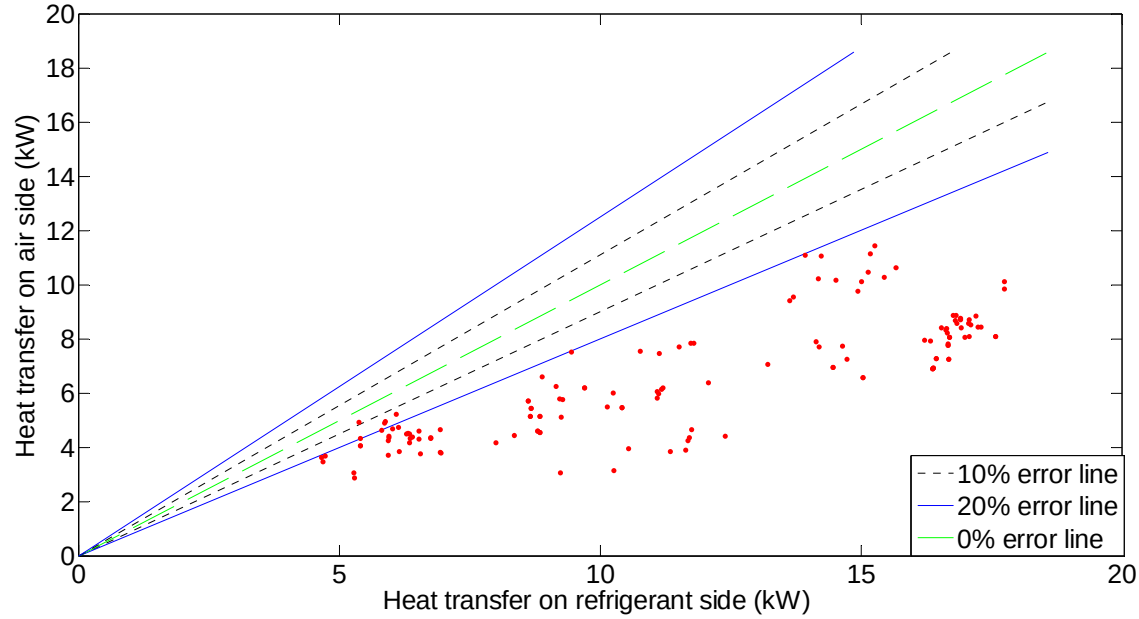
Corrections of Pseudo-Steady State Data

The heat transfer between the refrigerant side and air side are compared for the Rheem 5T data and Coleman 5T data sets, which are shown in Figure 5.15, Figure 5.17 and Figure 5.19. The horizontal coordinate represents the heat transfer on the refrigerant

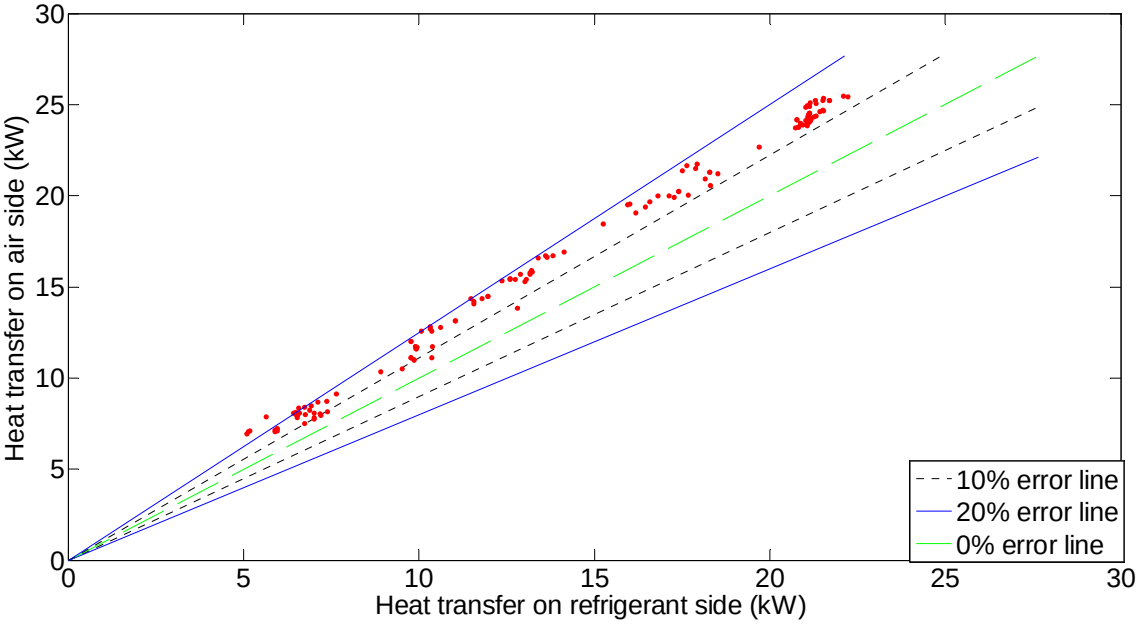
side, while the vertical coordinate represents the heat transfer on the air side. These figures show that the measured heat transfer does not match very well between the refrigerant side and air side.

Measuring air mass flow rates is difficult because air is a compressible gas, meaning the volume of a fixed air mass depends on the pressure and temperature it is subject to. Therefore, the measured air flow rate will vary with changes in temperature and pressure. Since the real purpose of measuring air flow rate is to obtain air mass flow rate, the result is not accurate due to the reasons mentioned above. The blockage effect caused by the sensor and probe is another reason is why measuring accurate air flow is difficult. The sensor and probe can block a portion of the flow path, decreasing the cross section flow area. Furthermore, the standard conditions used to measure air flow also vary, causing additional error in measurement. To increase measuring accuracy, one approach is to measure the flow repeatedly. However, this is not practical for field heat pump or air conditioning system, as real-time measurement is required.

For the refrigerant side, the mass flow rate can be predicted by the compressor. Emerson provided accurate compressor ratings including 20 performance coefficients. These ratings were used to calculate the steady state mass flow rates during pseudo-steady state data generation. Therefore, in the heat pump model validation, refrigerant side data is chosen to tune the model in order to match with the data (i.e. the goal of the validation is to match the refrigerant side properties). According to this, the corrected CFM is calculated to match with the refrigerant side, which is shown in Figure 5.16, Figure 5.18, and Figure 5.20.

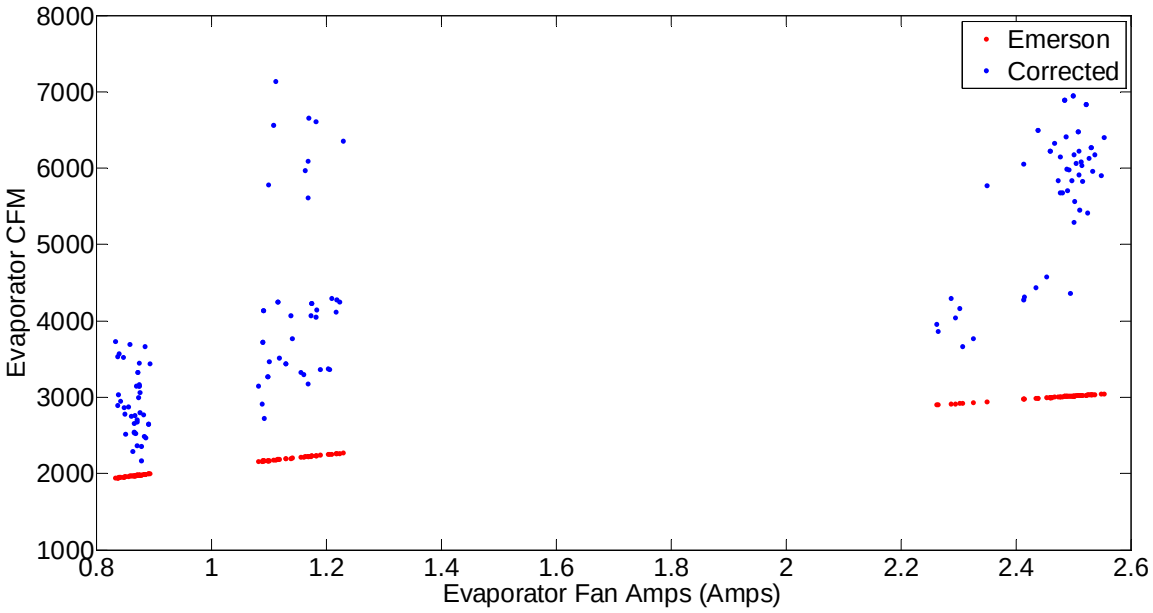


(a) Evaporator

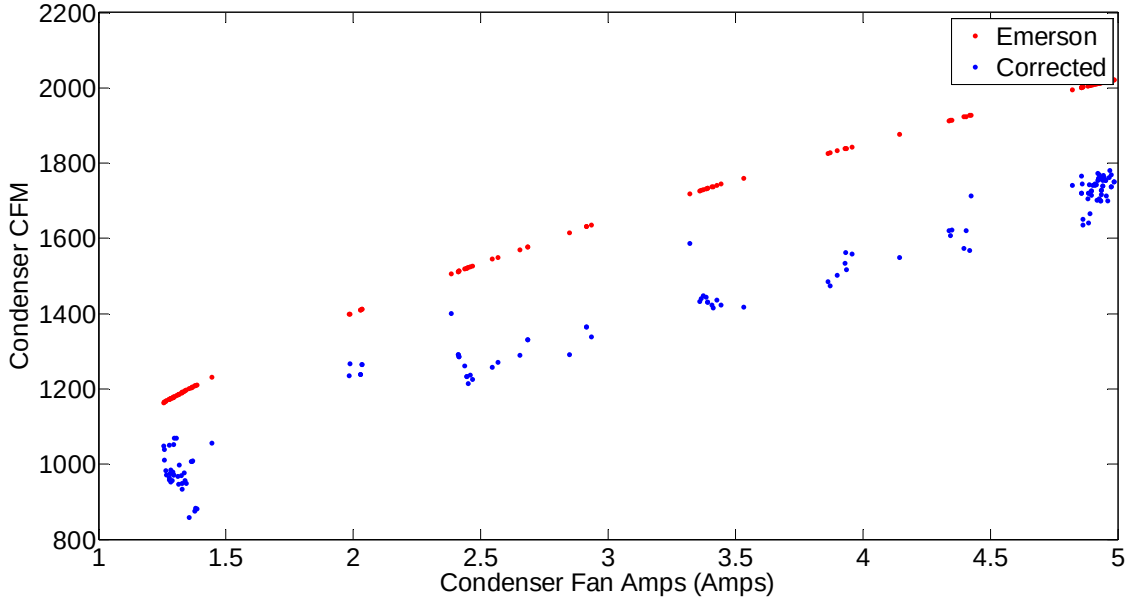


(b) Condenser

Figure 5.15: Heat transfer comparison for heat exchangers (Rheem 5T heating data).

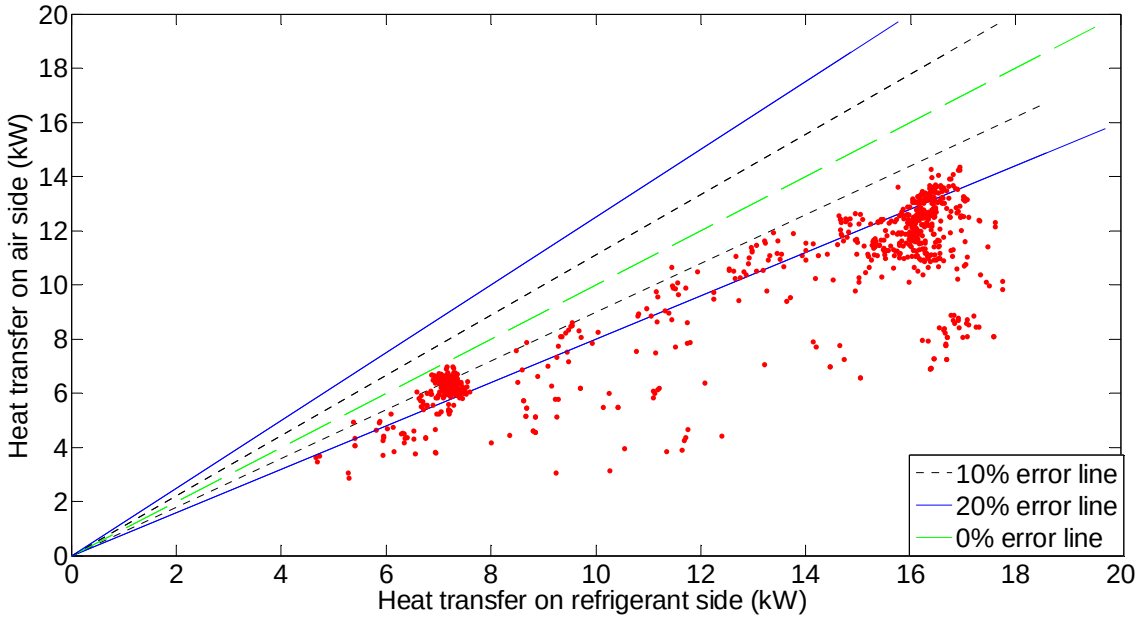


(a) Evaporator

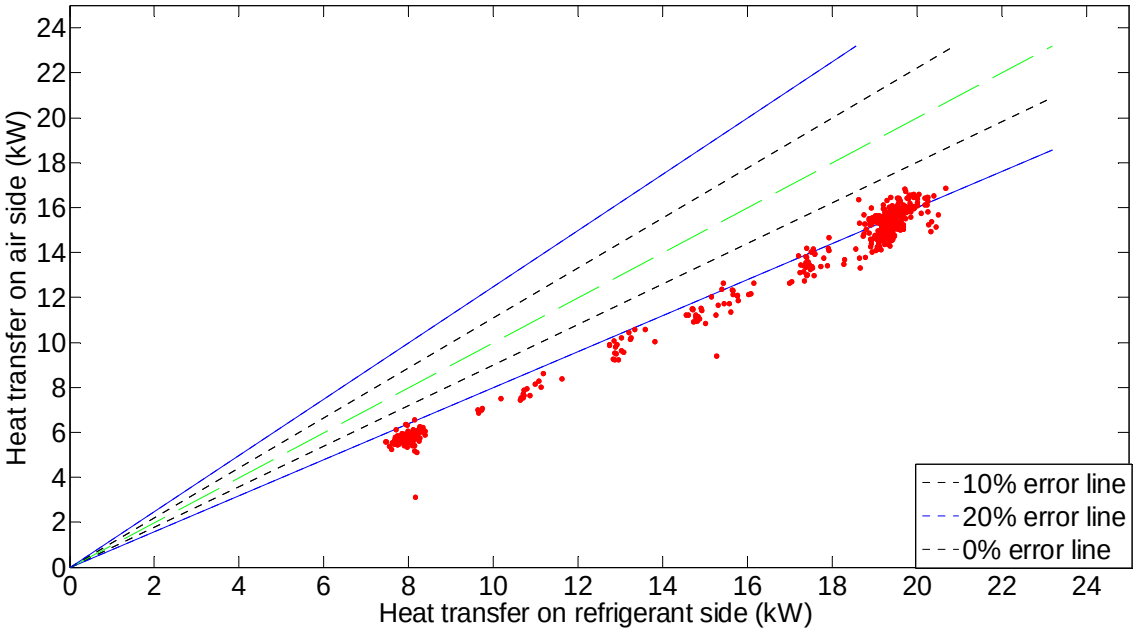


(b) Condenser

Figure 5.16: CFM correction for heat exchangers (Rheem 5T heating data).

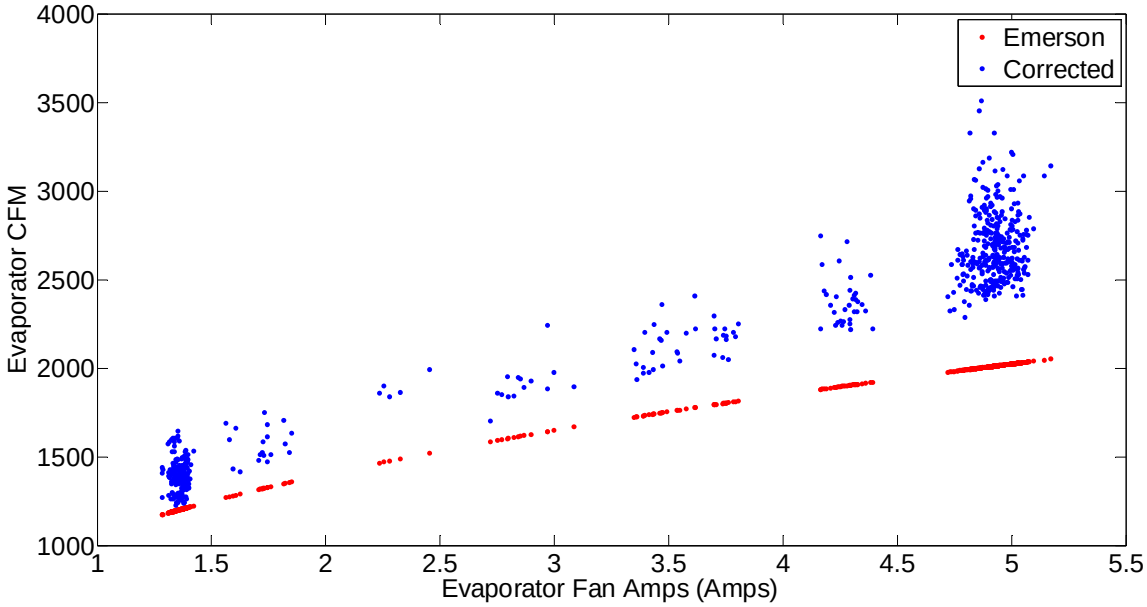


(a) Evaporator

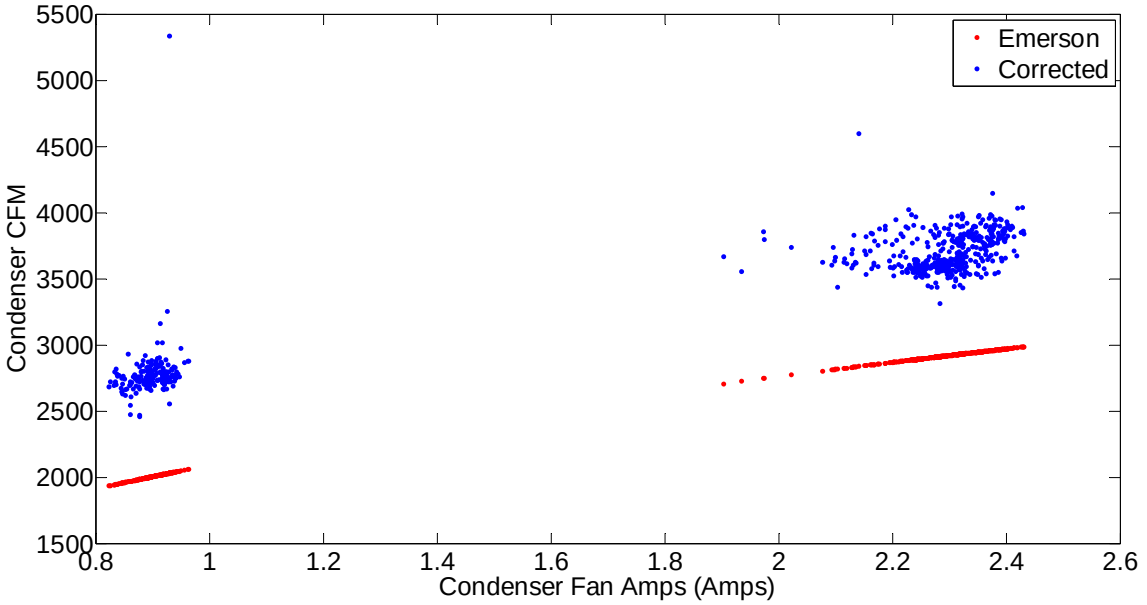


(b) Condenser

Figure 5.17: Heat transfer comparison for heat exchangers (Rheem 5T cooling data).

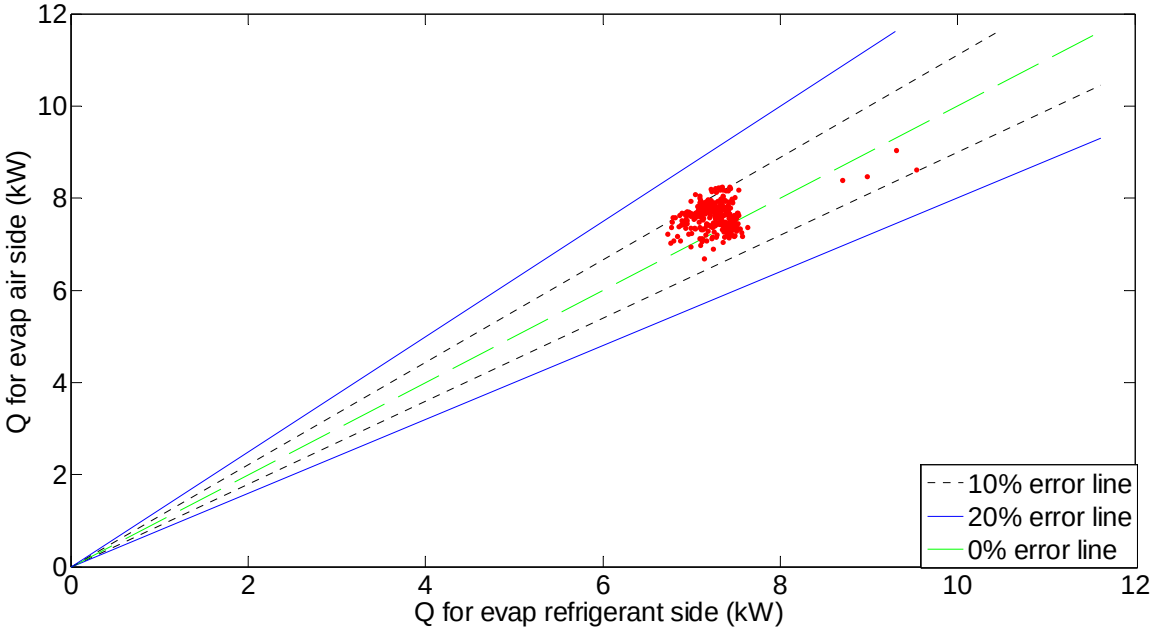


(a) Evaporator

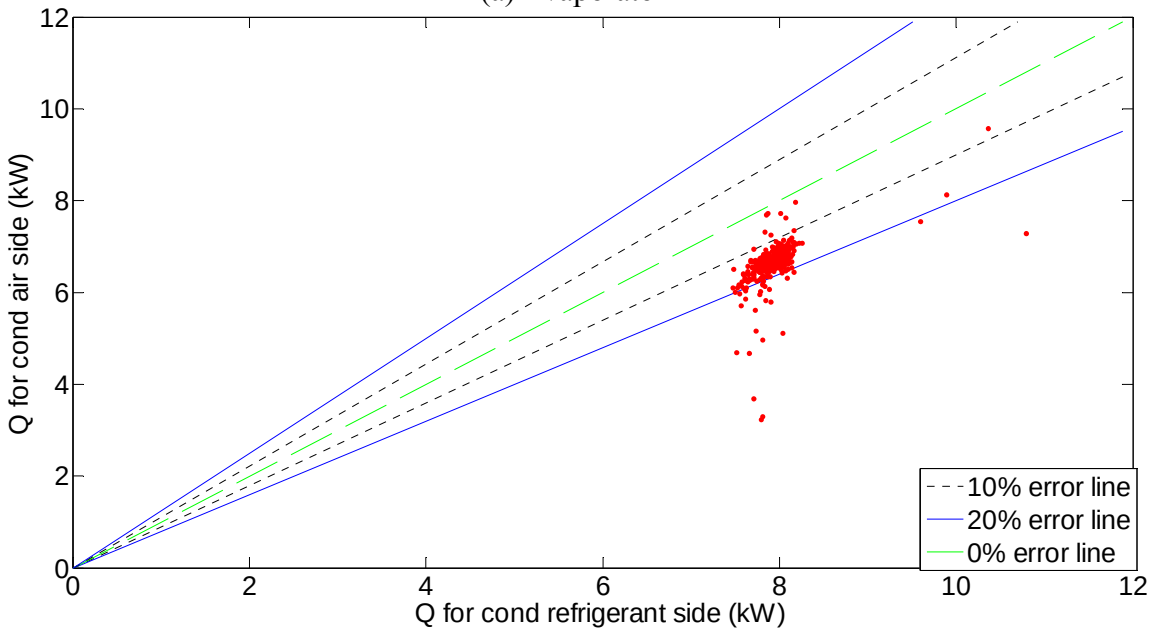


(b) Condenser

Figure 5.18: CFM correction for heat exchangers (Rheem 5T cooling data).

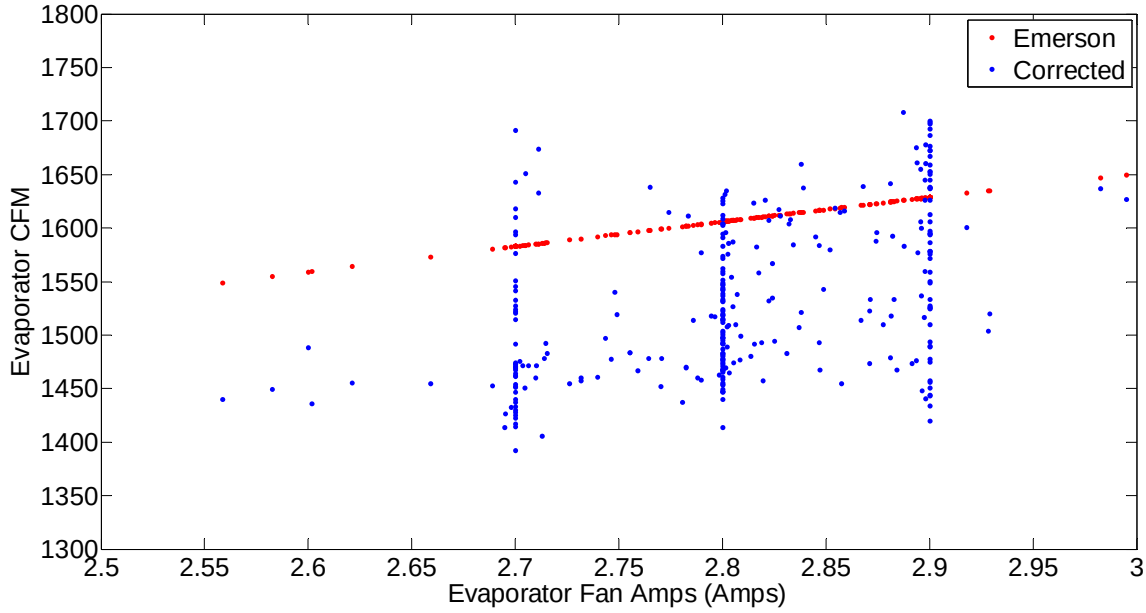


(a) Evaporator

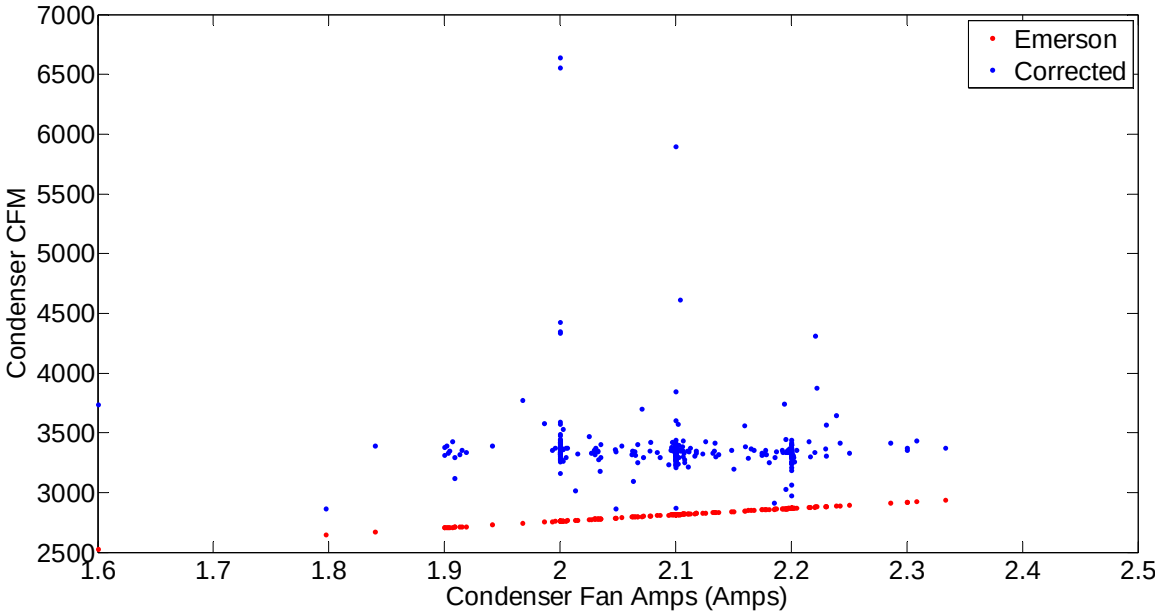


(b) Condenser

Figure 5.19: Heat transfer comparison for heat exchangers (Coleman 5T cooling data).



(a) Evaporator



(b) Condenser

Figure 5.20: CFM correction for heat exchangers (Coleman 5T cooling data).

The choice of match with refrigerant side for validation will also be proved in pseudo-steady state validation results shown in the following sections. Figure 5.21

shows the comparison of refrigerant side heat transfer between tuned model and pseudo-steady state data. These points are calculated from the tuning results from Figure 5.22.

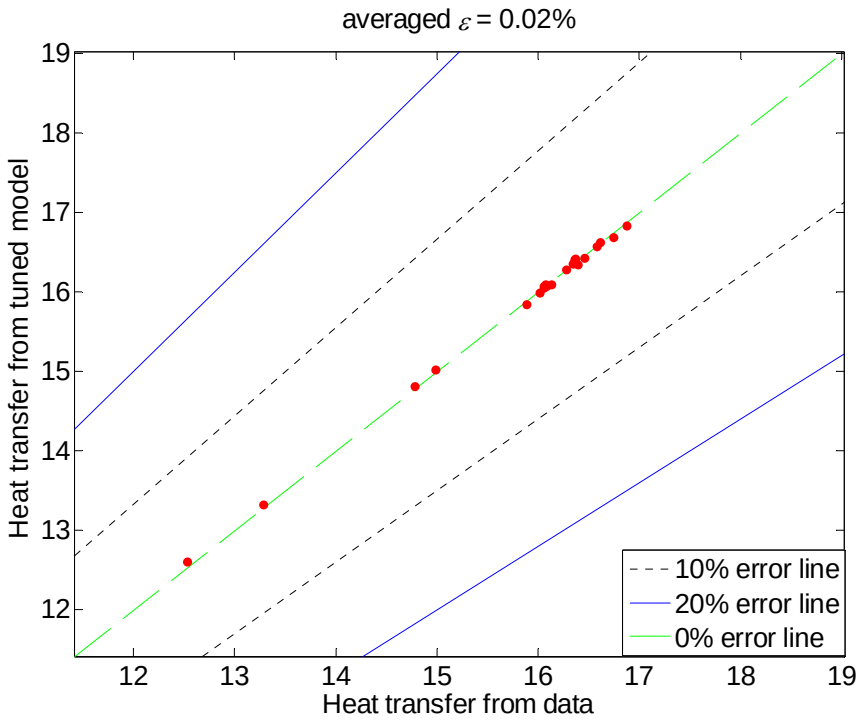


Figure 5.21: Refrigerant side heat transfer between tuned model and pseudo-steady state data.

Heat Pump Model Validation Using Steady State Data

FCV Heat Exchanger Model

Evaporator FCV Model Pseudo-Steady State Validation

The evaporator FCV model is tuned and validated using the pseudo-steady state data generated from Rheem 5T system and Coleman 5T system. The tuning parameters for the evaporator include cross section area of the heat exchanger tube, internal surface

area, external surface area, tube length and tube diameter. All the pseudo-steady state data are assumed to have the same set of tuning parameters, thus they are tuned simultaneously to find the optimal tuning parameters so that the predicted steady state outputs match with the pseudo-steady state data. The results are shown below. The horizontal coordinate represents the generated pseudo-steady state data, and the vertical coordinate represents the value predicted by the tuned model. These figures show that the predicted outputs by the tuned model match very well with the pseudo-steady state data. For each output, the averaged relative error for all steady state points is shown in the figures. For Rheem 5T system, the averaged relative errors of the three outputs are 0, 0.07%, and 2.7%, respectively. For the Coleman 5T system, the averaged relative errors of the three outputs are 0, 0.28% and 5.29% respectively.

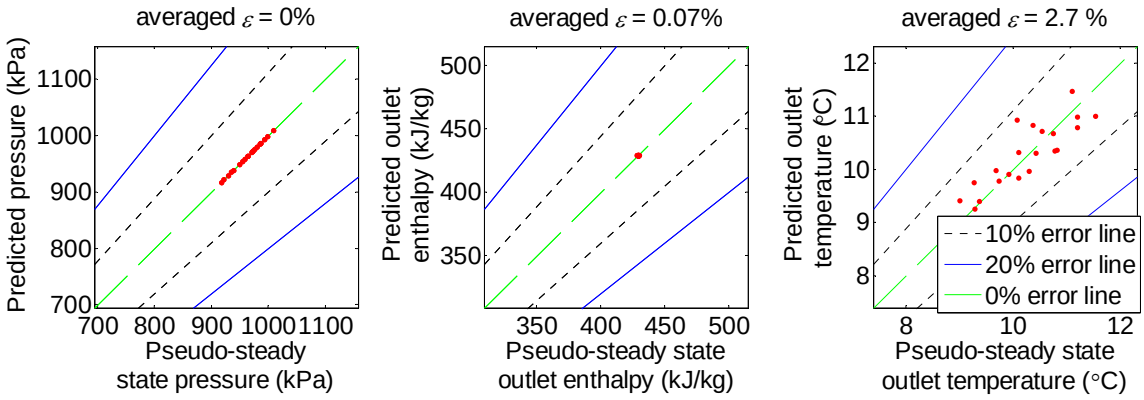


Figure 5.22: Evaporator FCV model validation using Rheem 5T pseudo-steady state cooling data.

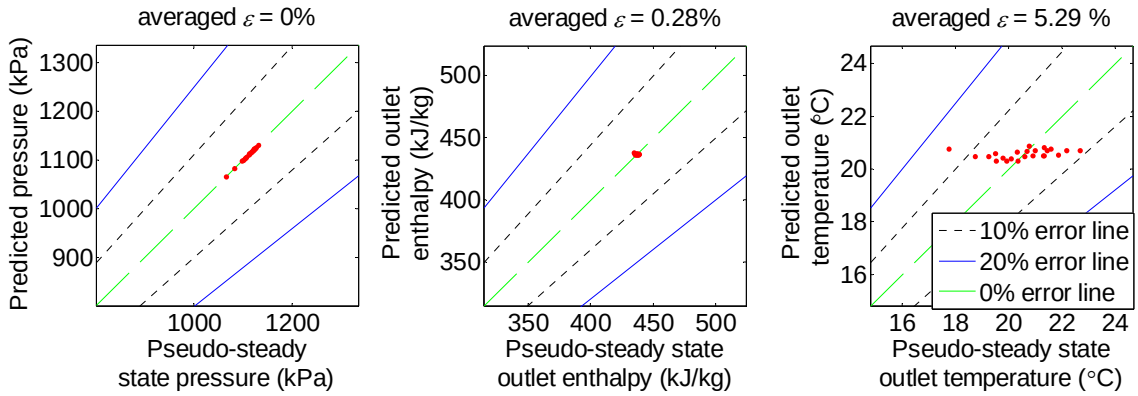


Figure 5.23: Evaporator FCV model validation using Coleman 5T pseudo-steady state cooling data.

Condenser FCV Model Pseudo-Steady State Validation

The condenser FCV model is also tuned and validated using the pseudo-steady state data generated from Rheem 5T system and Coleman 5T system. The tuning parameters for the condenser include cross section area of the heat exchanger tube, internal surface area, external surface area, tube length and tube diameter. All the pseudo-steady state data are assumed to have the same set of tuning parameters, thus they are tuned simultaneously to find the optimal tuning parameters so that the predicted steady state outputs match with the pseudo-steady state data. The results are shown below. The horizontal coordinate represents the generated pseudo-steady state data, and the vertical coordinate represents the value predicted by the tuned model. These figures show that the predicted data by the tuned model match very well with the data. For the Rheem 5T system, the relative errors of the three outputs are 0, 8.18% and 9.05%, respectively. For the Coleman 5T system, the relative errors of the three outputs are 0, 2.13% and 4.06% respectively.

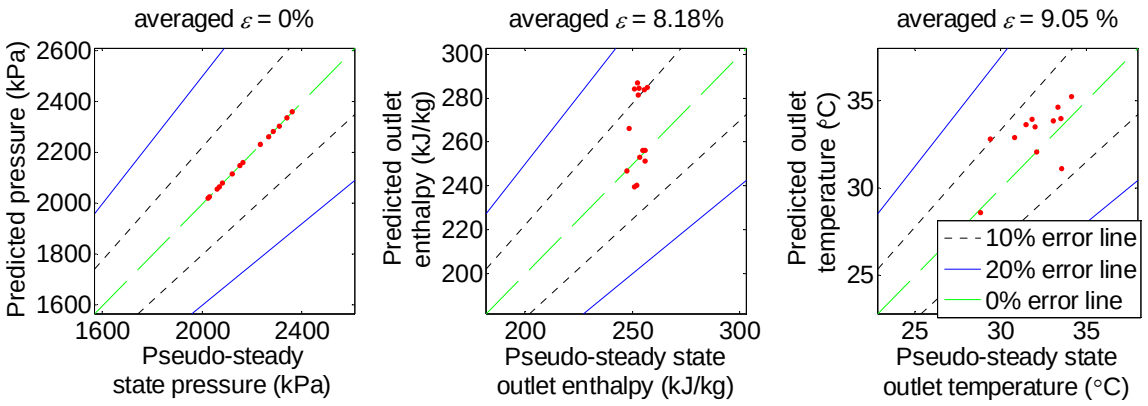


Figure 5.24: Condenser FCV model validation using Rheem 5T pseudo-steady state heating data.

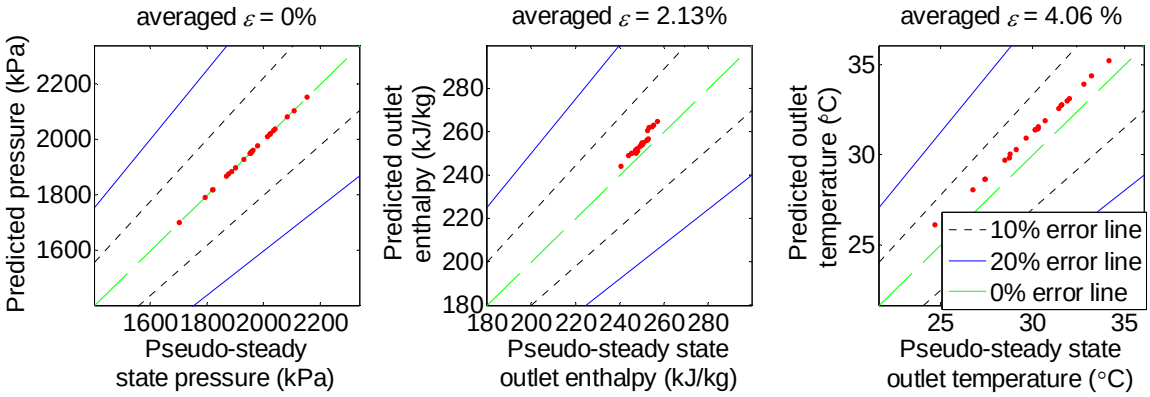


Figure 5.25: Condenser FCV model validation using Coleman 5T pseudo-steady state cooling data.

Complete Heat Pump Model

The heat pump model is validated using the pseudo-steady state data generated from Rheem 5T heating and cooling field data, and Coleman 5T cooling field data. For Rheem 5T pseudo-steady state heating data, the results are shown Figure 5.26.

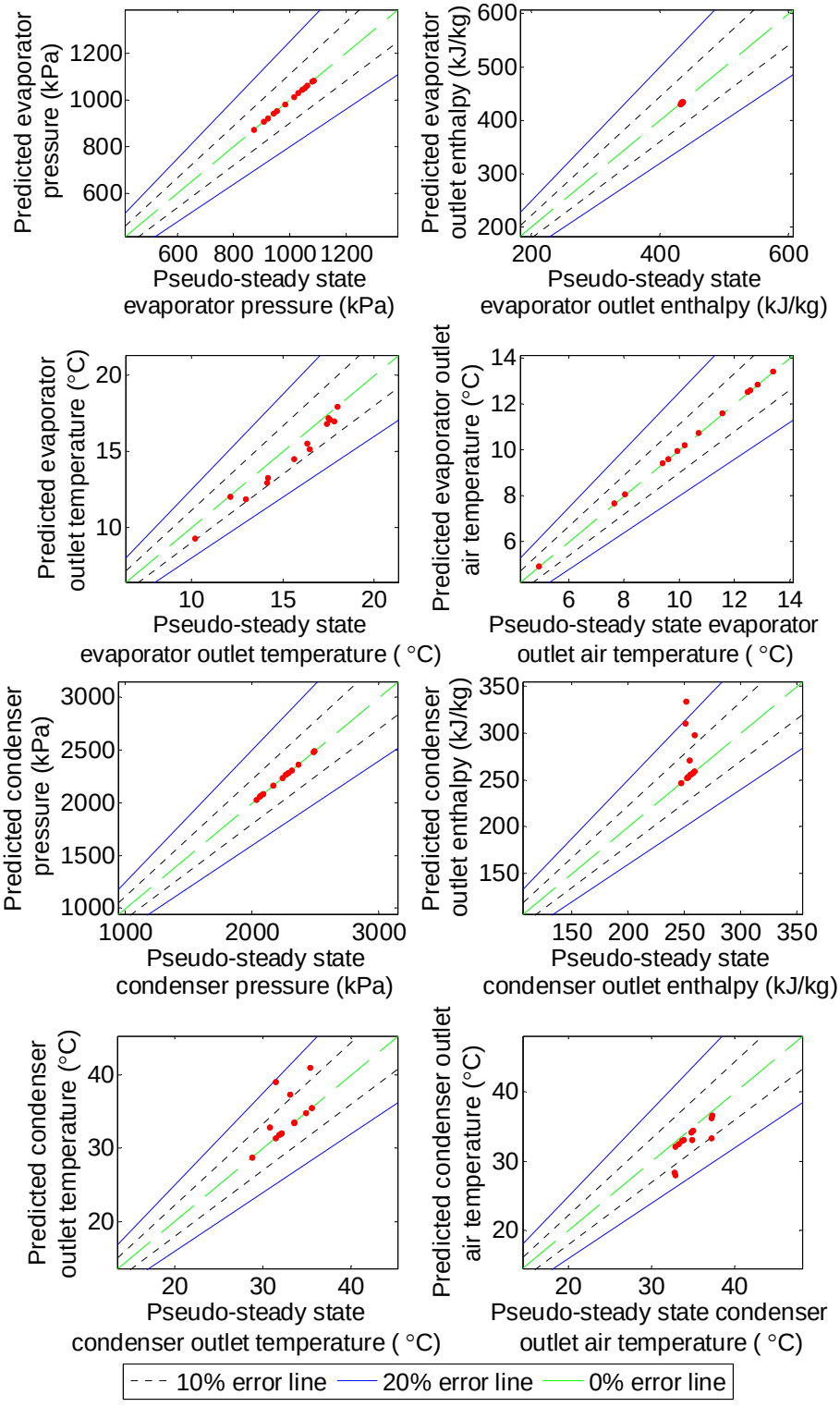


Figure 5.26: Complete HP model validation using Rheem 5T pseudo-steady state heating data.

The tuning parameters for the model include cross section area of the heat exchanger tube, internal surface area, external surface area, tube length, tube diameter, heat transfer coefficient, diameter and length of the pipe between condenser and compressor, heat transfer coefficient, diameter and length of the pipe between condenser and valve. The horizontal coordinate represents the generated pseudo-steady state data, and the vertical coordinate represents the value predicted by the tuned model. These figures indicate that the predicted data by the tuned model match very well with the generated pseudo-steady state data. The averaged relative error of all the outputs is 13.81%.

For Rheem 5T pseudo-steady state cooling data, the results are shown in Figure 5.27. For Coleman 5T pseudo-steady state cooling data, the results are shown in Figure 5.28. For Rheem data, the tuning parameters for the model include cross section area of the heat exchanger tube, internal surface area, external surface area, tube length, tube diameter, heat transfer coefficient, diameter and length of the pipe between condenser and compressor, heat transfer coefficient, diameter and length of the pipe between condenser and valve.

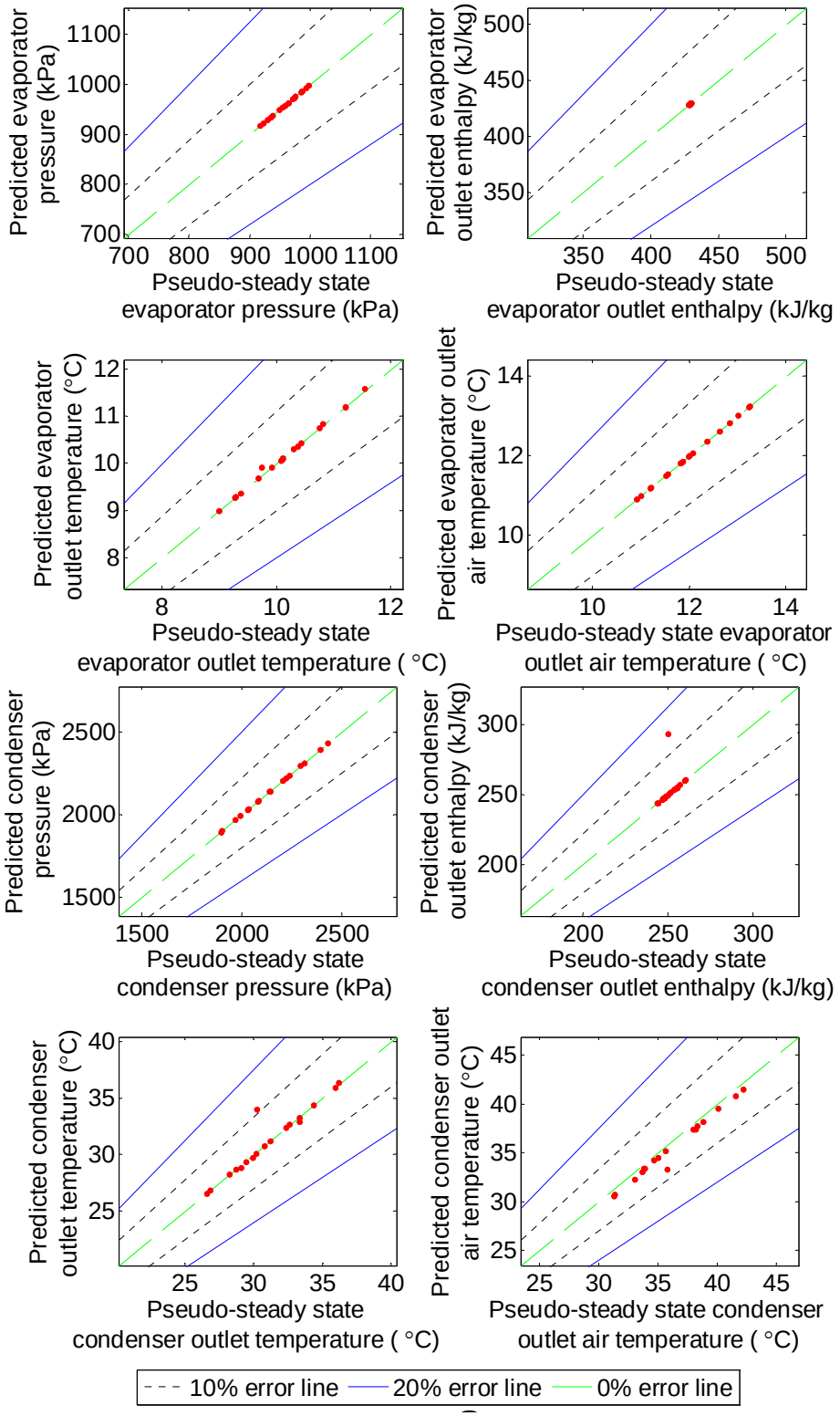


Figure 5.27: Complete HP model validation using Rheem 5T pseudo-steady state cooling data.

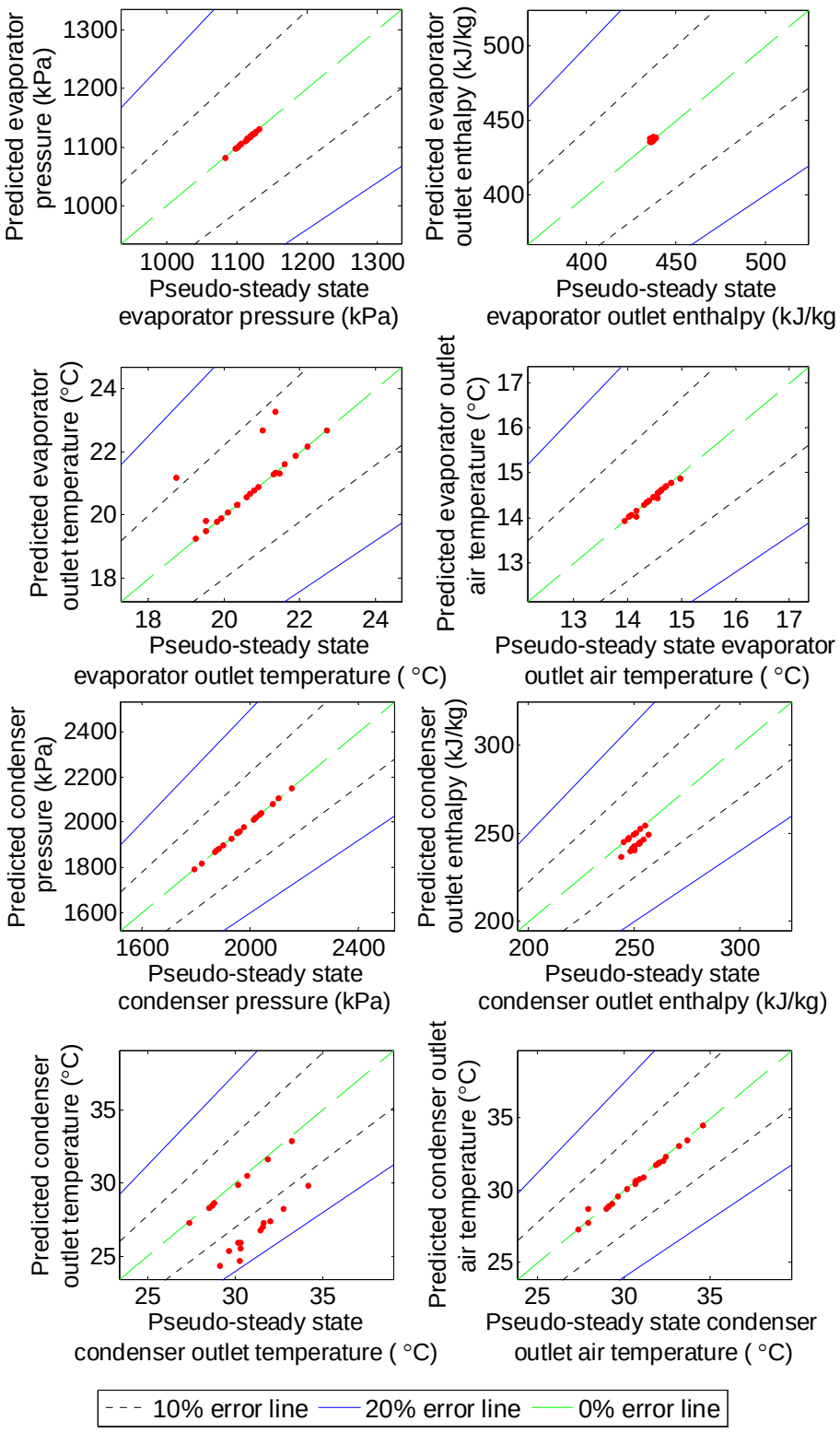


Figure 5.28: Complete HP model validation using Coleman 5T pseudo-steady state cooling data.

The horizontal coordinate represents the generated pseudo-steady state data, and the vertical coordinate represents the value predicted by the tuned model. These figures clearly show that the predicted data by the tuned model match very well with the generated pseudo-steady state data. The averaged relative error of all the outputs is 5.53%.

For Coleman 5T pseudo-steady state cooling data (Figure 5.28), the tuning parameters for the model include cross section area of the heat exchanger tube, internal surface area, external surface area, tube length, tube diameter, heat transfer coefficient, diameter and length of the pipe between condenser and compressor, heat transfer coefficient, diameter and length of the pipe between condenser and valve. The horizontal coordinate represents the generated pseudo-steady state data, and the vertical coordinate represents the value predicted by the tuned model. These figures clearly show that the predicted data by the tuned model match very well with the pseudo-steady state data. The averaged relative error of all the outputs is 17.68%.

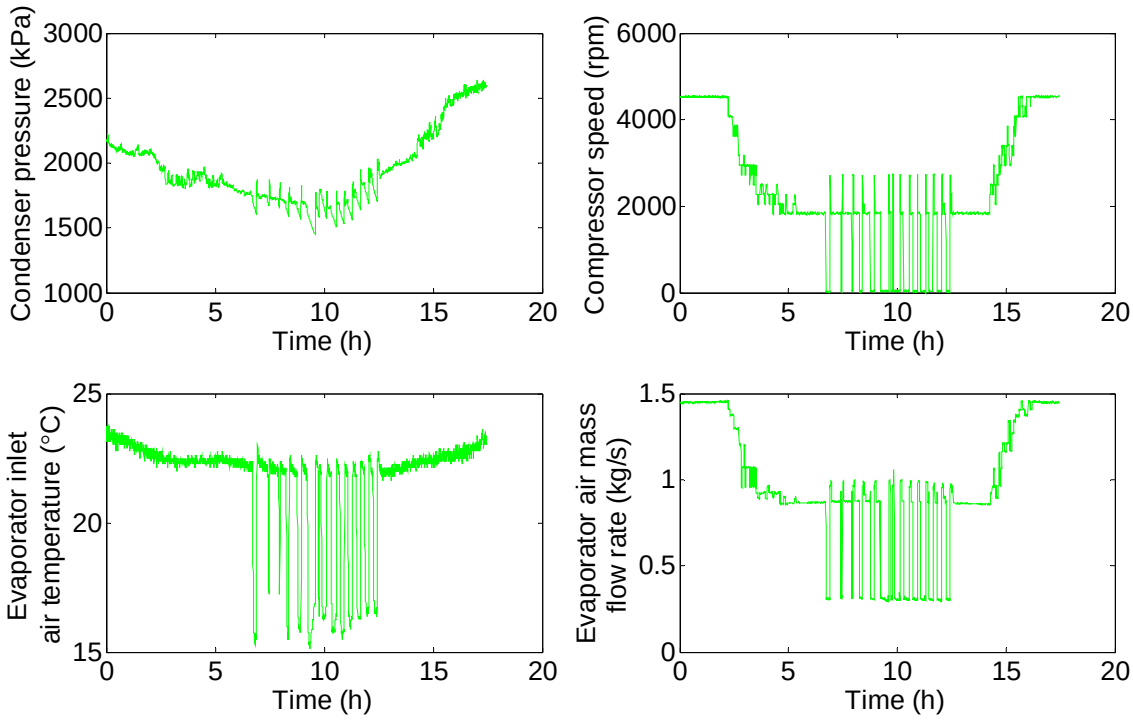
Simultaneous Parameter Tuning Using both Steady State and Transient Data

In this section, for each selected field data set, both steady state and transients in that data are used to tune the model simultaneously. The purpose is to avoid tedious steady state tuning before tuning the transient model. In this scenario, the objective function described in Equation (3.16) also includes the wavelet coefficients for steady state data.

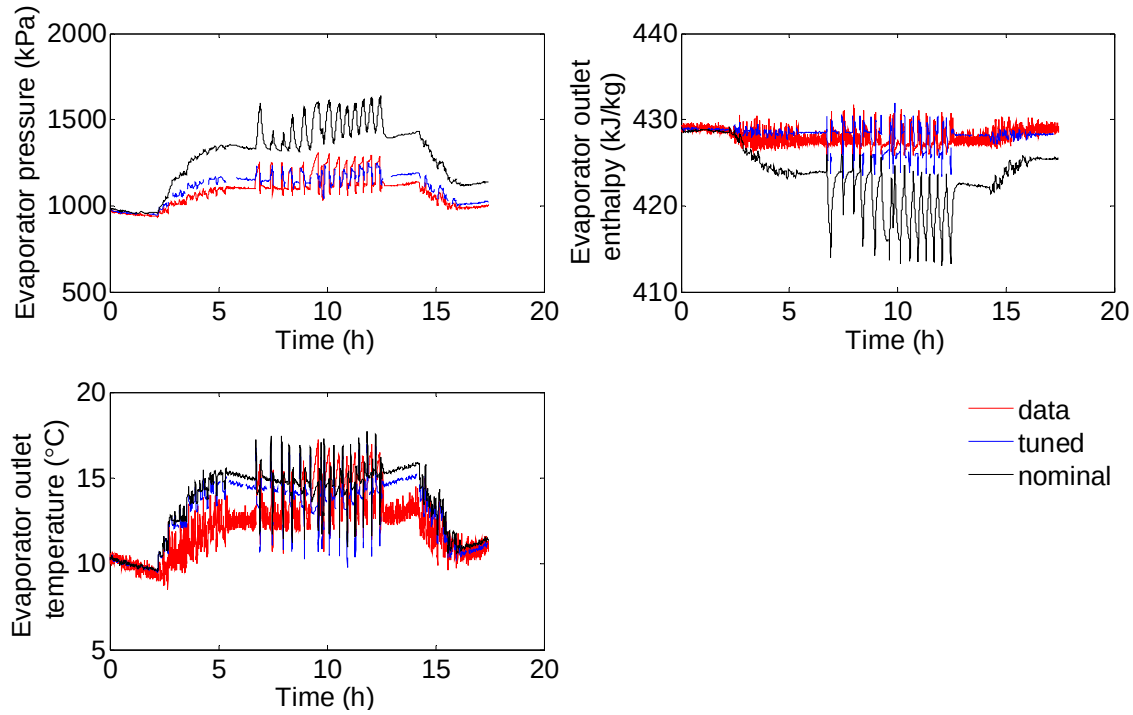
FCV Evaporator Model

Several Rheem and Coleman 5T cooling field data setswere selected to perform the FCV heat exchanger parameter tuning and validation. The tuning parameters for the evaporator include the internal and cross area of the tube, diameter, length, TXV and compressor map coefficients. The outputs used in the objective function include evaporator pressure and evaporator refrigerant outlet temperature. The following are some model tuning and validation results. Table 5.2 lists the simulation errors.

In Figure 5.29, the inputs contain numerous and frequent compressor startups and shutdowns. These drastic and hard transients are also shown in the output Figure 5.29 (b). The black lines in Figure 5.29 (b) represent the nominal predicted outputs of the evaporator model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. The match of pressure is better than that of the outlet temperature, because the weighting factor for the outlet temperature is set lower in the objective function. Note the match of steady states at the beginning of these outputs is also achieved simultaneously, which means simultaneous tuning is feasible for the FCV evaporator model using the proposed method.



(a) Inputs



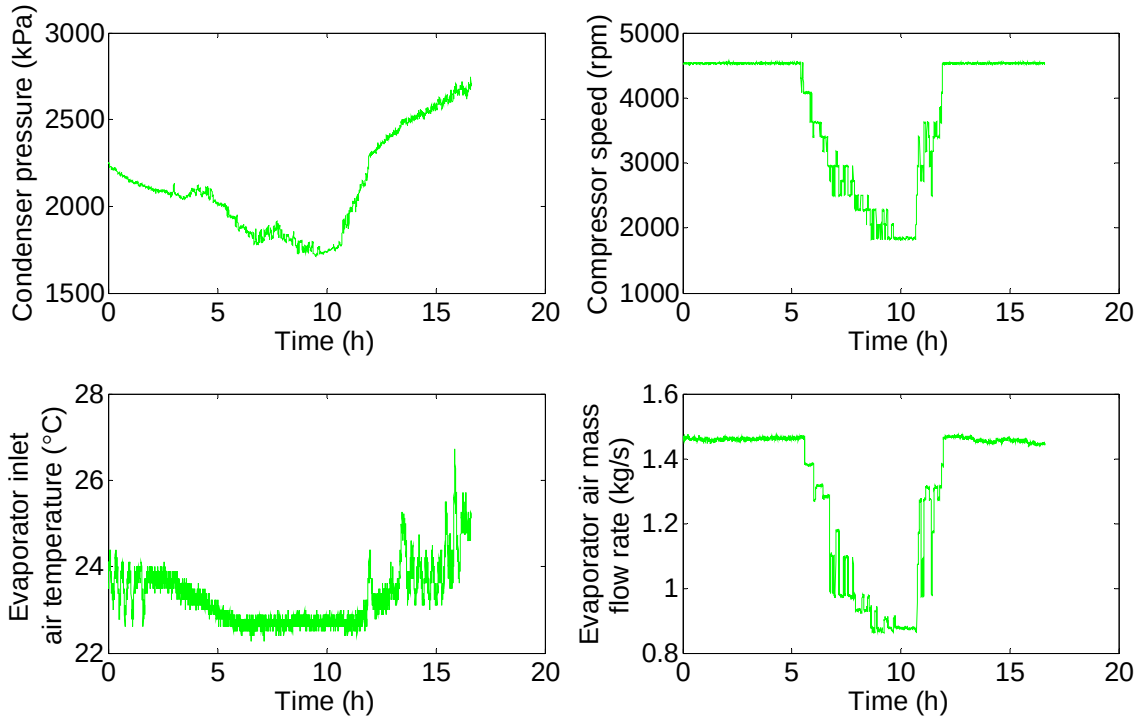
(b) Outputs

Figure 5.29: Evaporator FCV model validation using Rheem 5T field cooling data 05/30/2011.

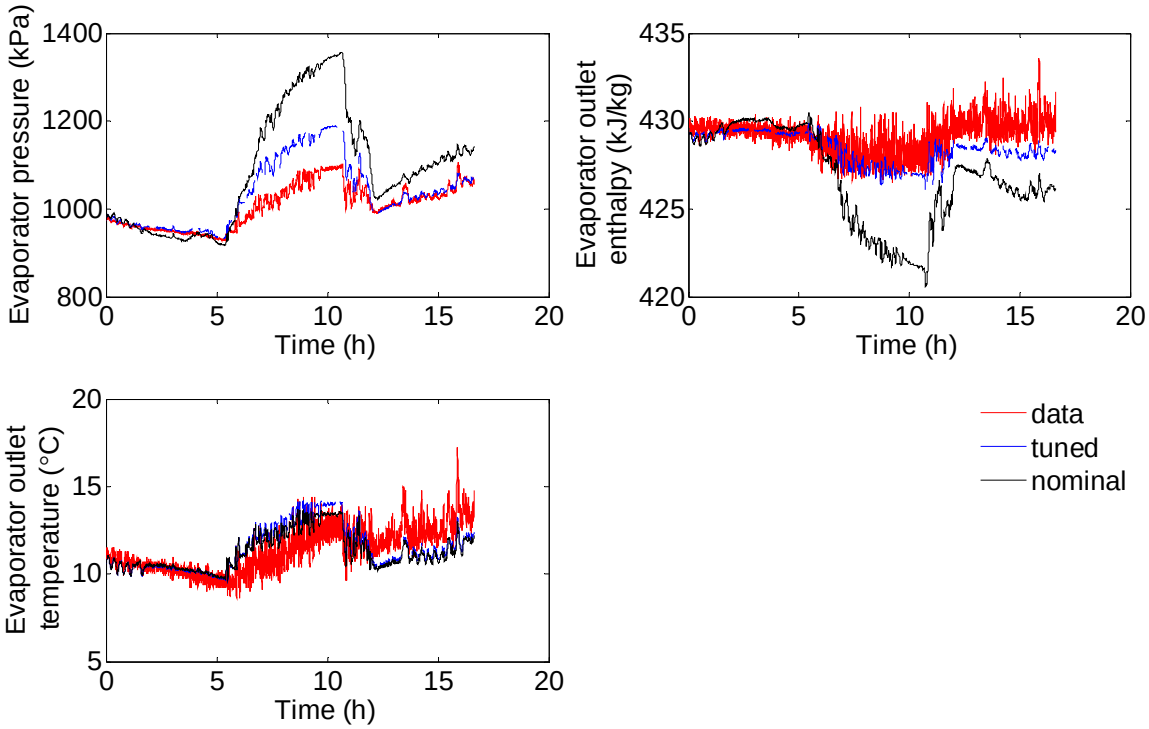
Similarly in Figure 5.30, although there is no compressor startup and shutdown, the compressor experiences frequent speed changes. These frequent transients are also seen in the output Figure 5.30 (b). The black lines in Figure 5.30 (b) represent the nominal predicted outputs of the evaporator model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. The match of pressure is worse than that of the temperature, because the weighting factor for the pressure is set lower in the objective function. Note the match of steady states at the beginning of these outputs is also achieved simultaneously, which means simultaneous tuning is feasible for the FCV evaporator model using the proposed method.

Table 5.2: Relative error of evaporator FCV model validation using field data.

Data	Rheem 5T 05/30/2011	Rheem 5T 08/01/2011	Rheem 5T 07/19/2011	Coleman 5T 05/17/2012	Coleman 5T 07/15/2012
Relative error (%)	5.85	5.20	2.84	18.39	4.44



(a) Inputs

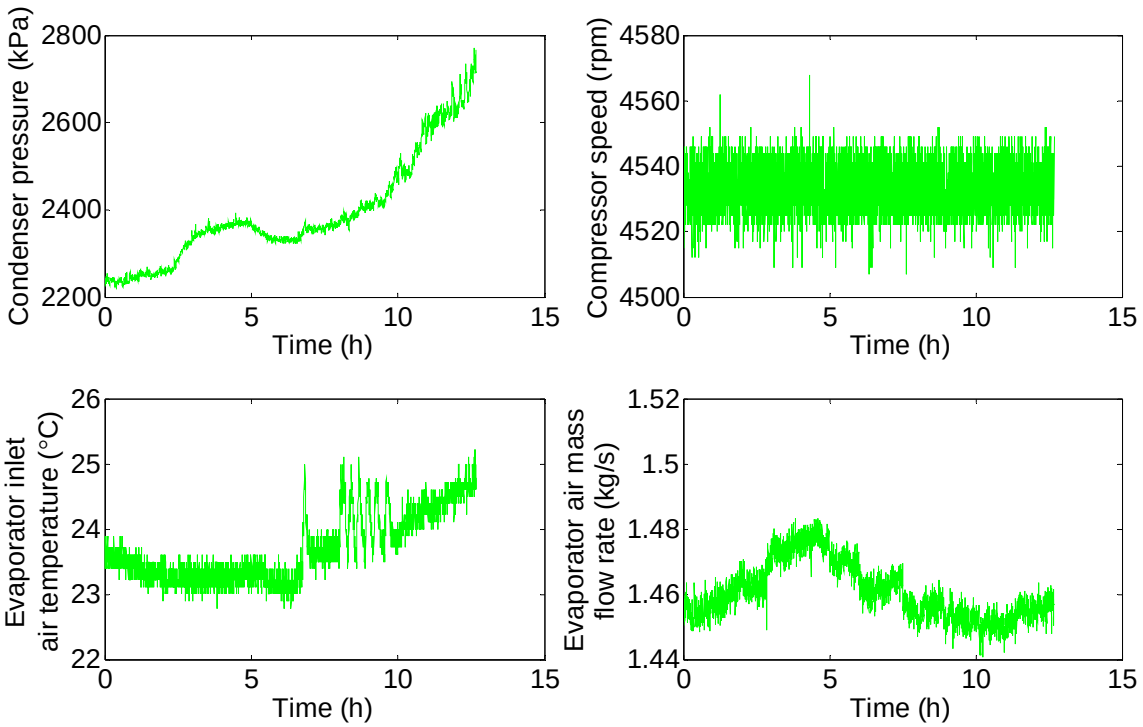


(b) Outputs

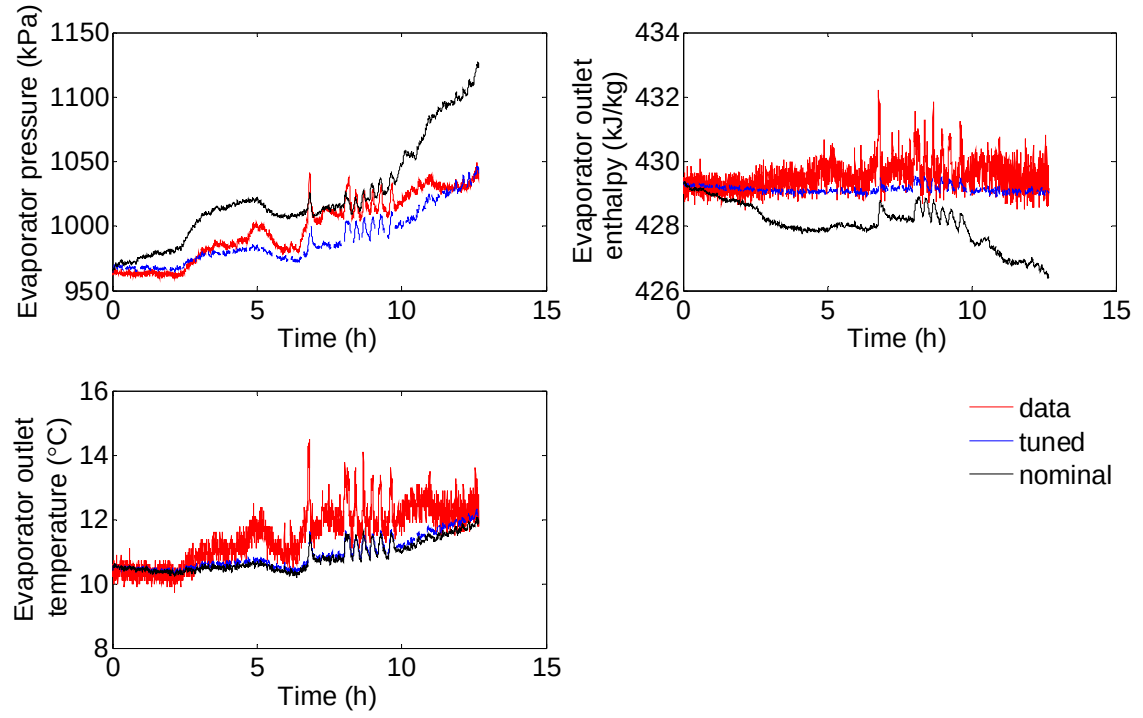
Figure 5.30: Evaporator FCV model validation using Rheem 5T field cooling data 08/01/2011.

In Figure 5.31, the compressor speed remains almost steady state. The inputs contain the gradual increase of the condenser pressure. These transients are also shown in the output Figure 5.31 (b). The black lines in Figure 5.31 (b) represent the nominal predicted outputs of the evaporator model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. The match of pressure is better than that of the temperature, because the weighting factor for the temperature is set lower in the objective function. The match of steady state temperature is still good regardless of the transient match.

Figure 5.32 shows the parameter tuning results using one Coleman data set for the evaporator model. The inputs contain numerous and frequent compressor startups and shutdowns. These drastic and hard transients are far more than that in Figure 5.29, posing a great challenge to parameter tuning and validation. The black lines in Figure 5.32 (b) represent the nominal predicted outputs of the evaporator model, which are distinct from the data shown in the red lines. In this validation, weighting factor for the pressure is set higher, thus the transient match in pressure is good. The steady states match for all the outputs is still simultaneously achieved.

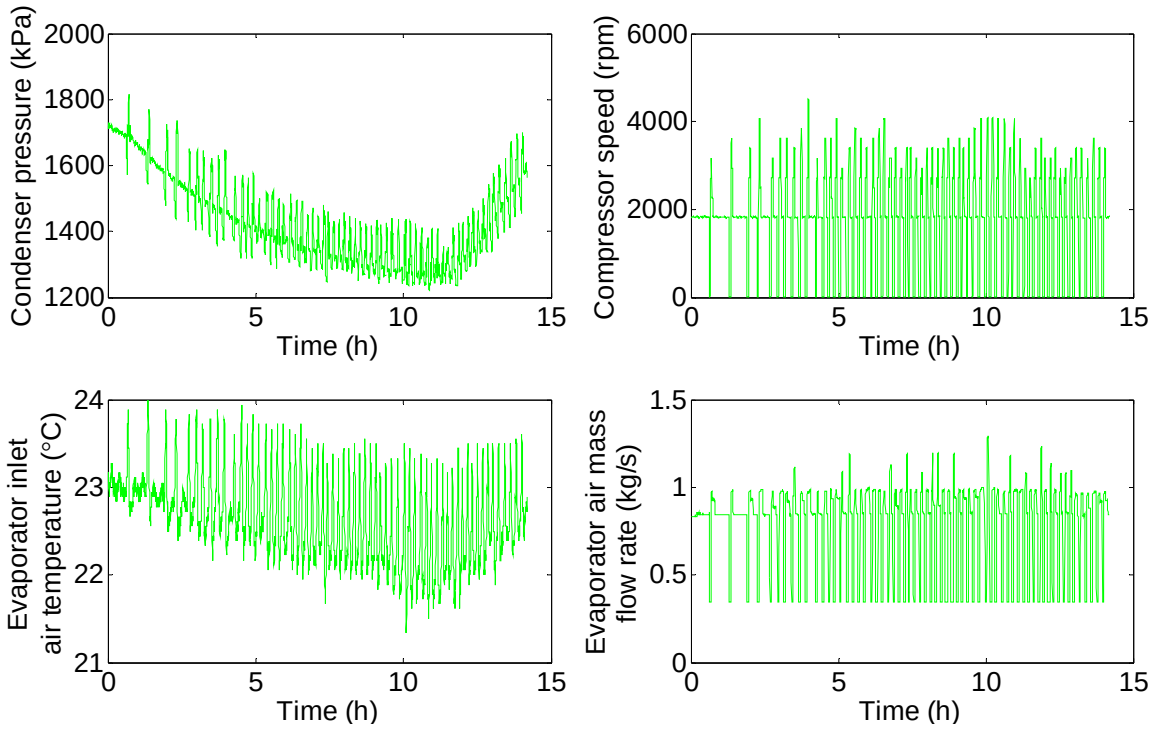


(a) Inputs

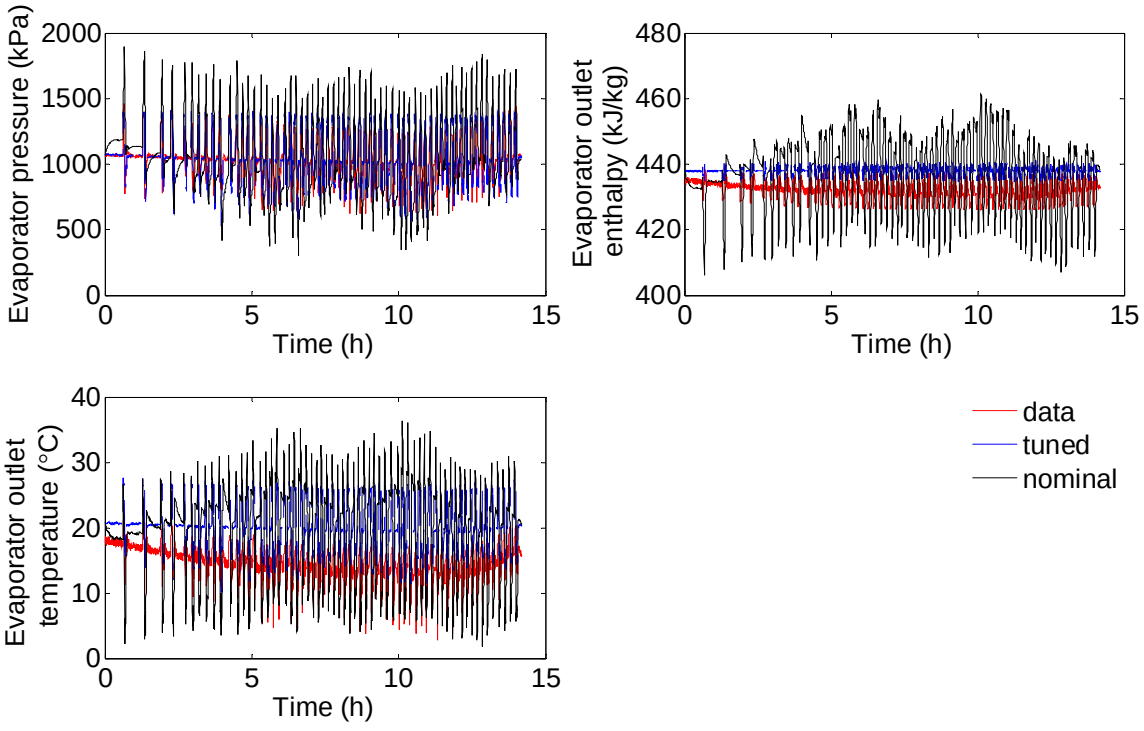


(b) Outputs

Figure 5.31: Evaporator FCV model validation using Rheem 5T field cooling data 07/19/2011.



(a) Inputs

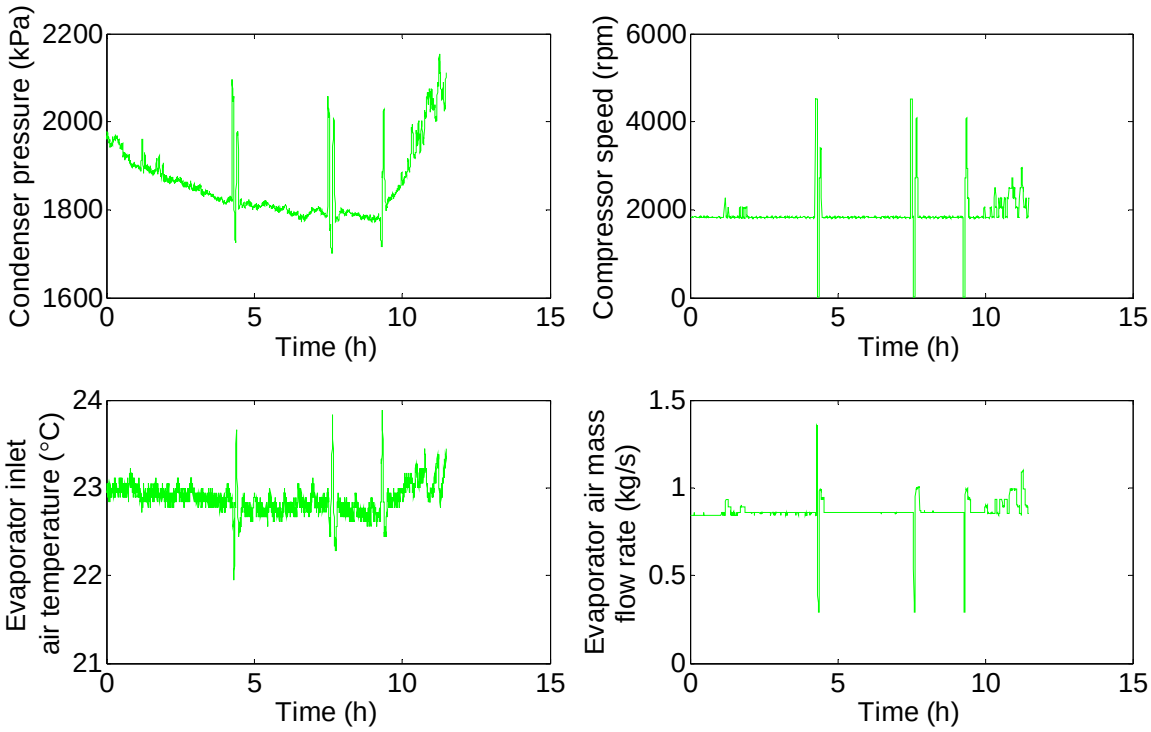


(b) Outputs

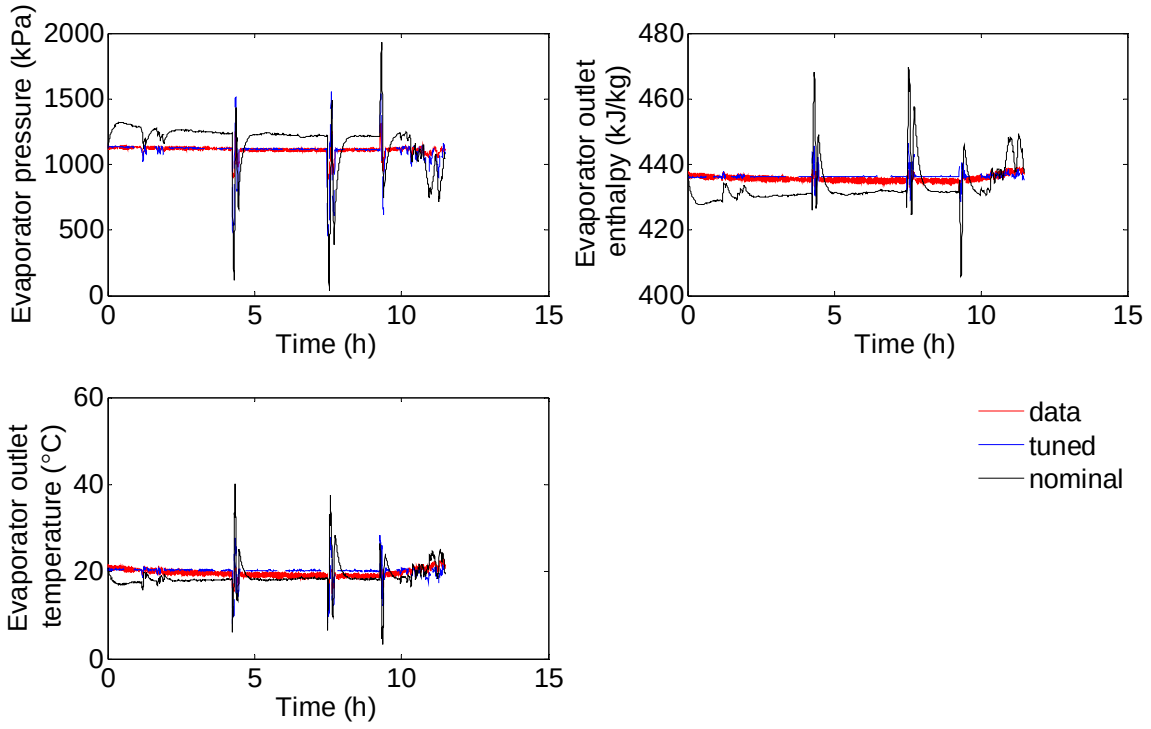
Figure 5.32: Evaporator FCV model validation using Coleman 5T field cooling data 05/17/2012.

In Figure 5.33, the inputs contain less frequent compressor startups and shutdowns; this may be helpful in tuning and validation. The transients are also seen in the output Figure 5.33 (b). The black lines in Figure 5.33 (b) represent the nominal predicted outputs of the evaporator model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. This can be seen more clearly in Figure 5.33 (c), which is a zoomed-in view of a small portion of the output transients. The match of the three outputs is equally good; in fact the individual relative errors for them are 1.26%, 1.52% and 1.66%, respectively. Note the match of steady states at the beginning of these outputs is also achieved simultaneously, which means simultaneous tuning is feasible for the FCV evaporator model using the proposed method.

All of these FCV evaporator model tuning and validation results indicate that the proposed parameter tuning and validation approach can indeed drive the predicted outputs toward the field data, even if the initial predicted outputs are distinct from the data, or the transients are drastic and frequent.

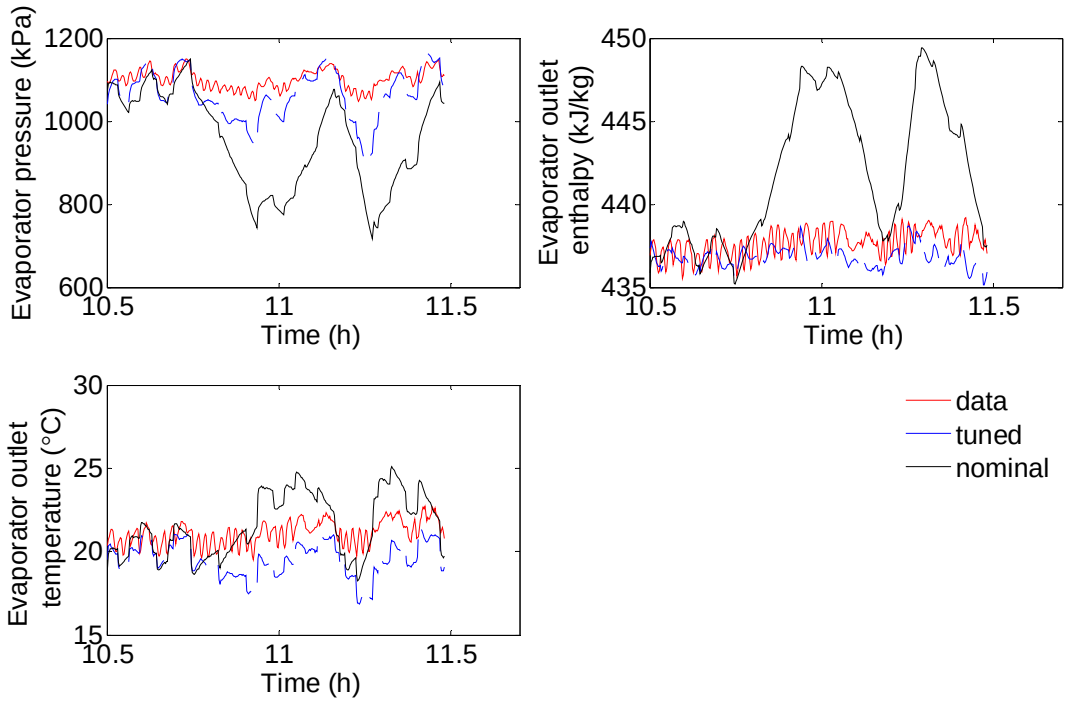


(a) Inputs



(b) Outputs

Figure 5.33: Evaporator FCV model validation using Coleman 5T field cooling data 07/15/2012.



(c) Outputs zoomed in view

Figure 5.33: Continued.

FCV Condenser Model

Several Rheem and Coleman 5T cooling field data sets were selected to perform the FCV condenser model validation. The tuning parameters for the condenser include the internal and cross area of the tube, diameter, length, TXV and compressor map coefficients. The outputs used in the objective function include condenser pressure and condenser refrigerant outlet temperature. The following are some model tuning and validation results. Table 5.3 lists the simulation errors.

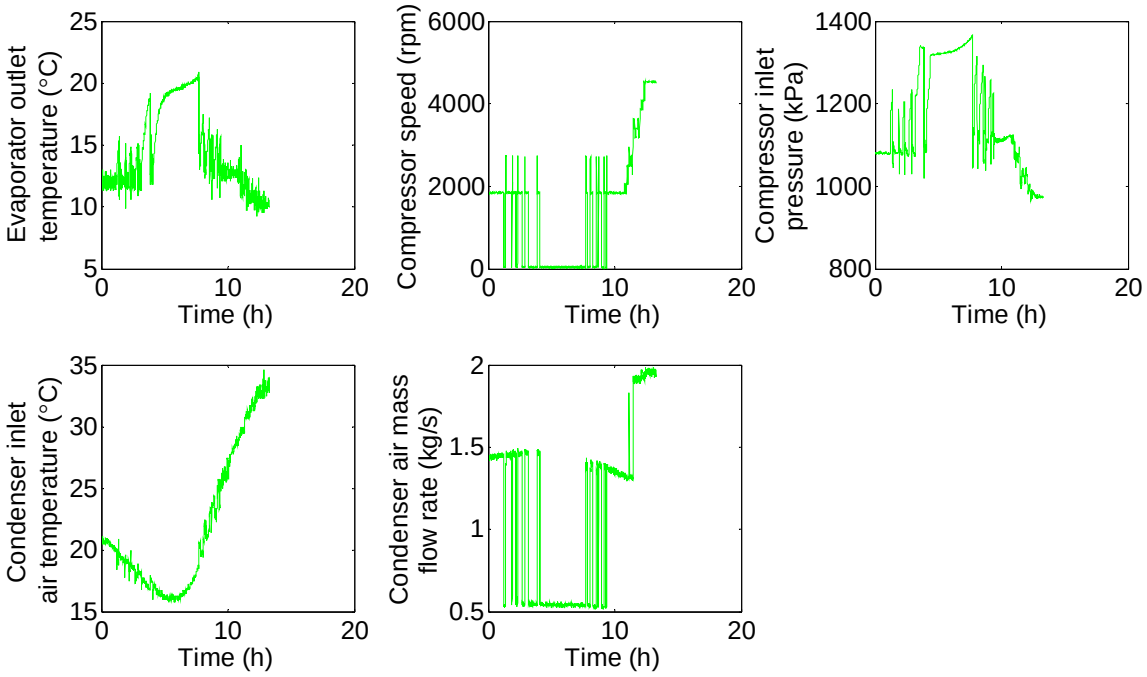
In Figure 5.34, the inputs contain frequent compressor startups and shutdowns, superheat changes, compressor inlet pressure changes, condenser inlet air temperature and mass flow rate changes. These transients are also shown in the output Figure 5.34

(b). The black lines in Figure 5.34 (b) represent the nominal predicted outputs of the condenser model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. The match of steady states for all the outputs is better than the corresponding transients, because the weighting factors for all the steady states are set higher in the objective function. Note the match of transients in these results is still achieved, which means simultaneous tuning is feasible for the FCV condenser model using the proposed method.

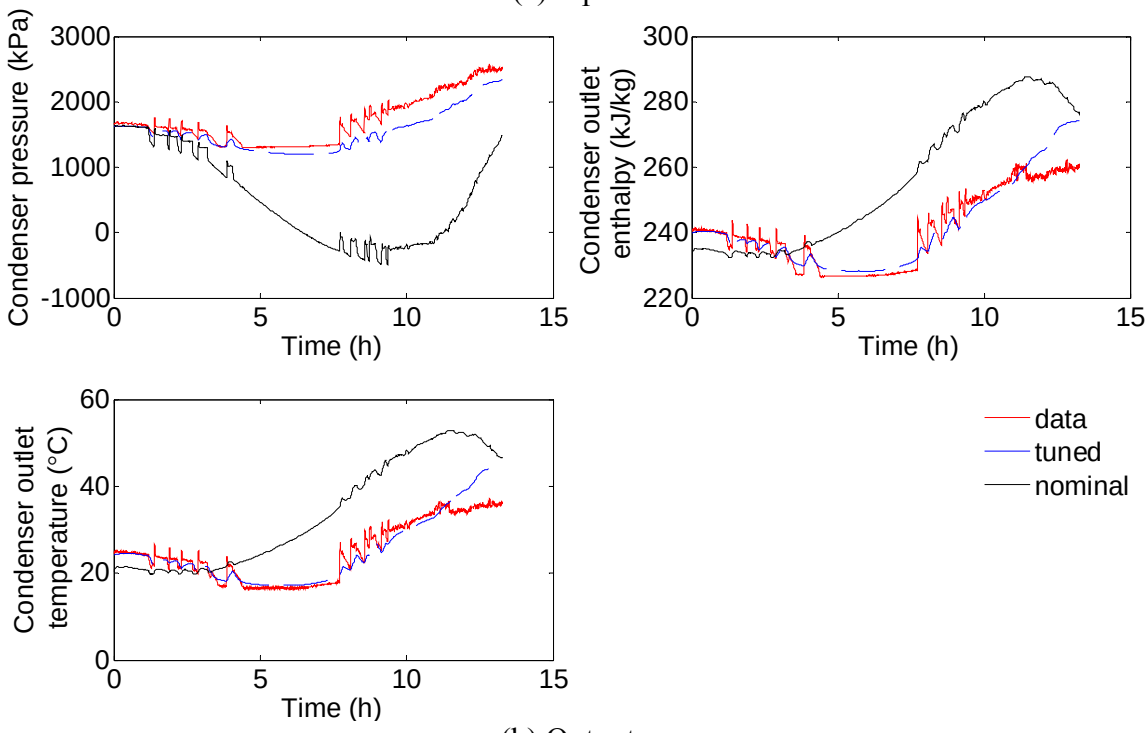
Table 5.3: Relative error of condenser FCV model validation using field data.

Data	Rheem 5T 06/06/2011	Rheem 5T 07/19/2011	Coleman 5T 05/17/2012	Coleman 5T 07/15/2012
Relative error (%)	8.38	2.84	4.67	3.67

In Figure 5.35, the compressor speed is kept almost constant. The inputs contain evaporator outlet temperature changes, compressor inlet pressure changes, condenser inlet air temperature and mass flow rate changes. These transients are also seen in the output Figure 5.35 (b). The black lines in Figure 5.35 (b) represent the nominal predicted outputs of the condenser model, which are distinct from the data shown in the red lines.

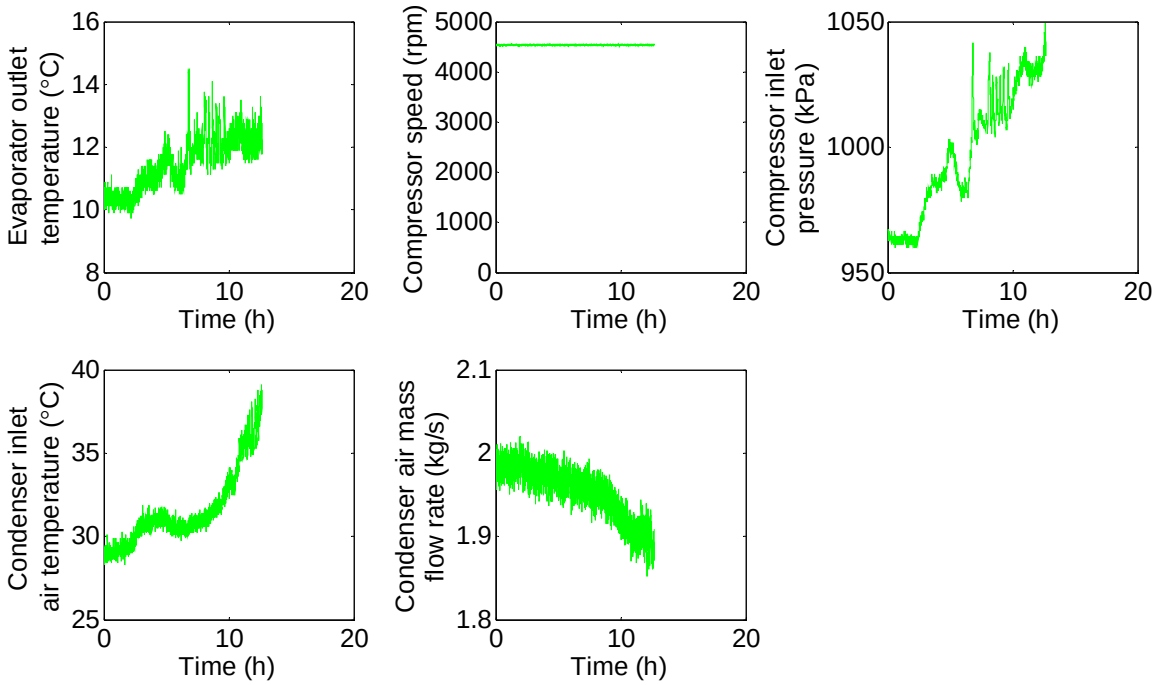


(a) Inputs

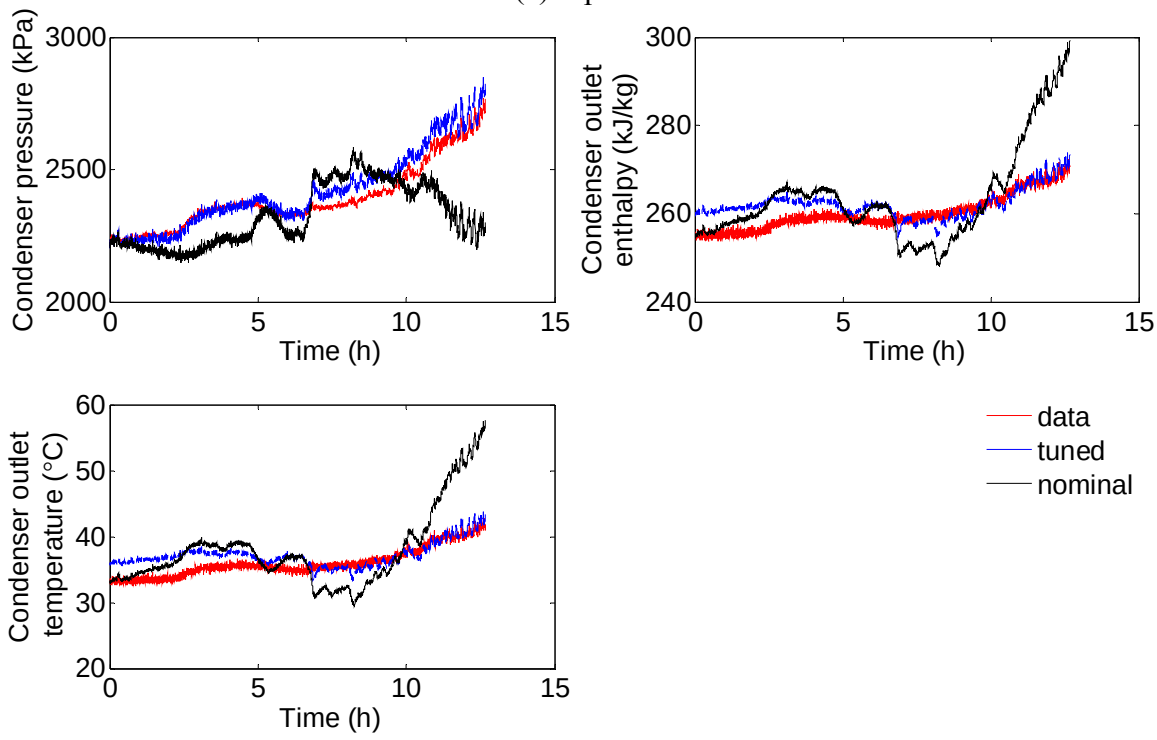


(b) Outputs

Figure 5.34: Condenser FCV model validation using Rheem 5T field cooling data 06/06/2011.



(a) Inputs

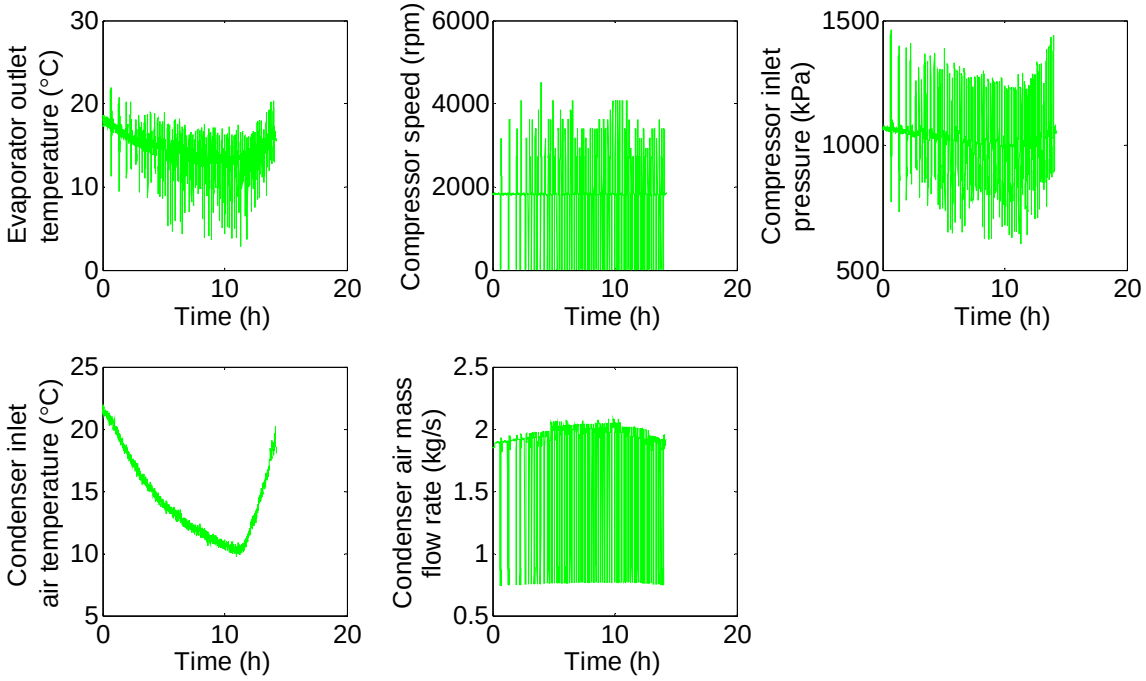


(b) Outputs

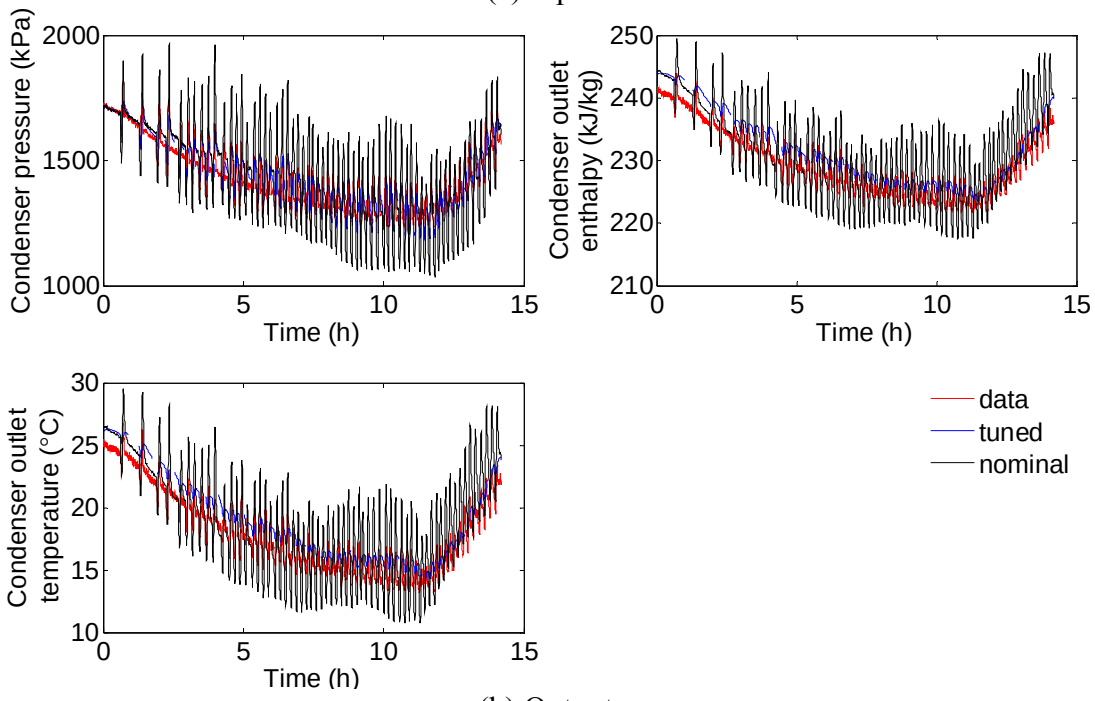
Figure 5.35: Condenser FCV model validation using Rheem 5T field cooling data 07/19/2011.

The proposed parameter tuning method drives the predicted outputs toward the field data, the final tuned outputs are shown in blue lines. The match of steady state for the pressure is better than those of the other two, because the weighting factors are set to emphasize more importance in transient match of all the outputs and steady state match of the pressure. Therefore, the match of transients in the temperature and enthalpy is still good, with a slight discrepancy in the steady state match.

Figure 5.36 shows the parameter tuning results using one Coleman data for the condenser model. The inputs contain numerous and frequent compressor startups and shutdowns. Other inputs also contain frequent and hard transients. The black lines in Figure 5.36 (b) represent the nominal predicted outputs of the heat pump model, which are distinct from the data shown in the red lines. In this validation, the weighting factors are again set to emphasize more importance in transient match of all the outputs and steady state match of the pressure. Therefore, the match of transients in the temperature and enthalpy is still good, with a slight discrepancy in the steady state match.



(a) Inputs

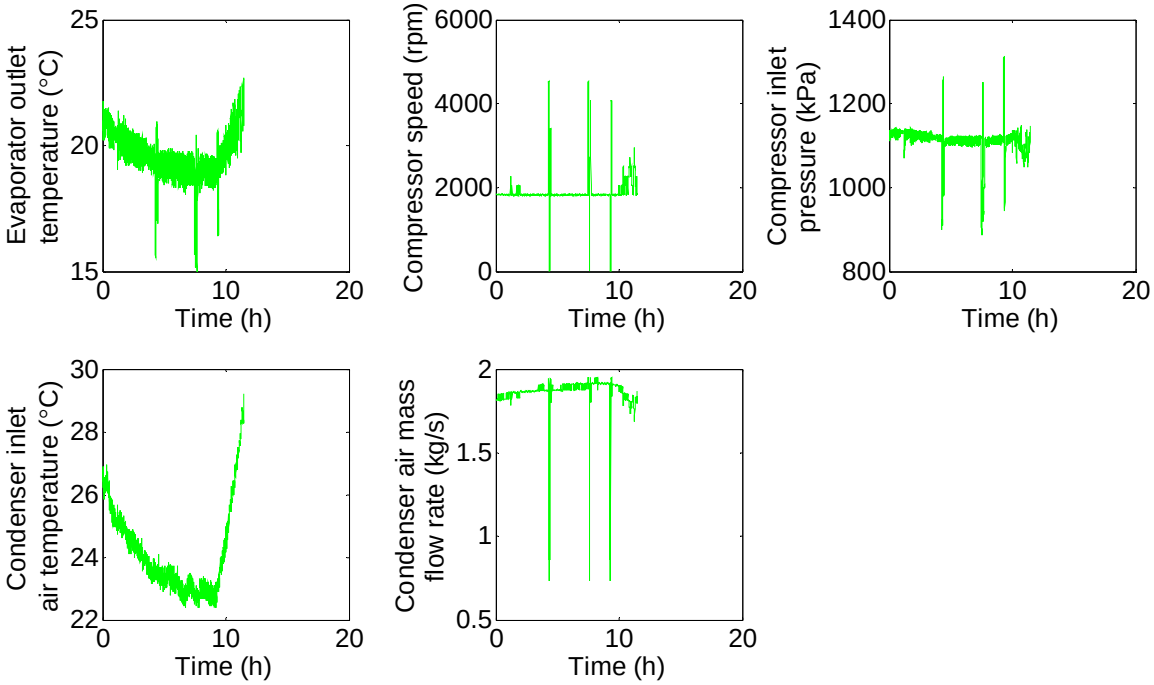


(b) Outputs

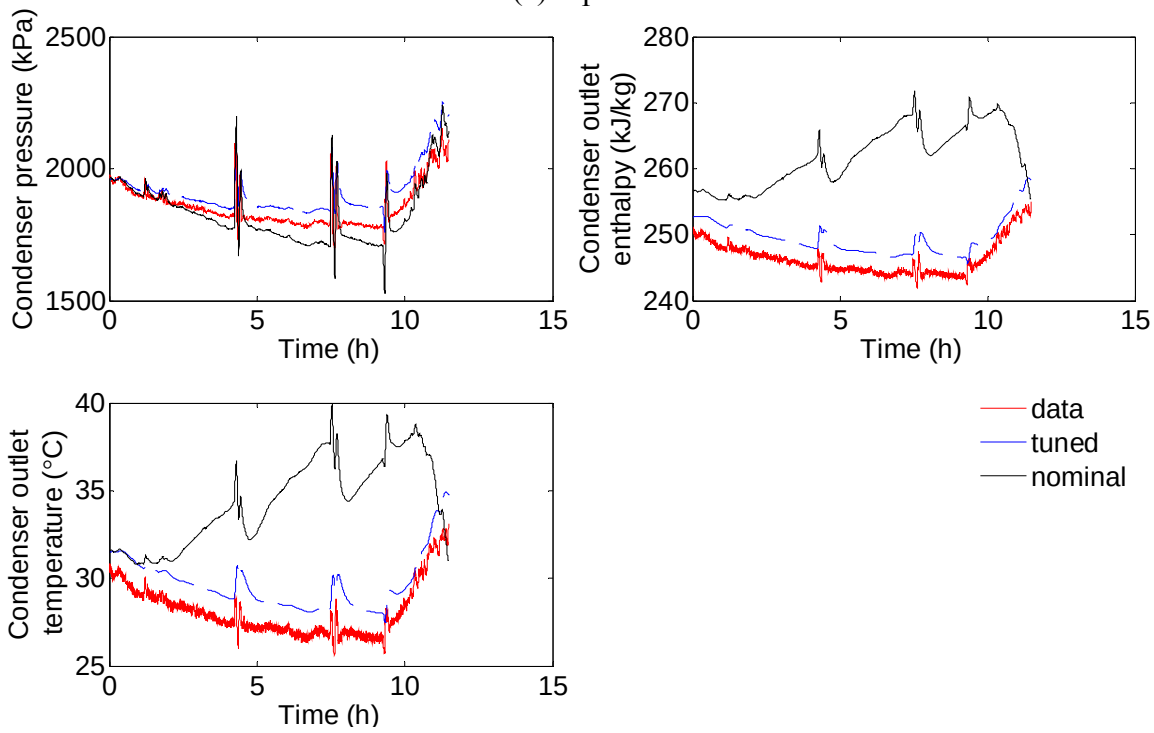
Figure 5.36: Condenser FCV model validation using Coleman 5T field cooling data 05/17/2012.

Figure 5.37 shows the parameter tuning results using another Coleman data for the condenser model. The inputs contain a few compressor startups and shutdowns. The inputs also include evaporator outlet temperature changes, compressor inlet pressure changes, condenser inlet air temperature and mass flow rate changes. The black lines in Figure 5.36 (b) represent the nominal predicted outputs of the heat pump model, which are distinct from the data shown in the red lines. In this validation, the weighting factors are again set to emphasize more importance in transient match of all the outputs and steady state match of the pressure.

All of these FCV condenser model tuning and validation results indicate that the proposed parameter tuning and validation approach can indeed drive the predicted outputs toward the field data, even if the initial predicted outputs are distinct from the data, or the transients are drastic and frequent.



(a) Inputs



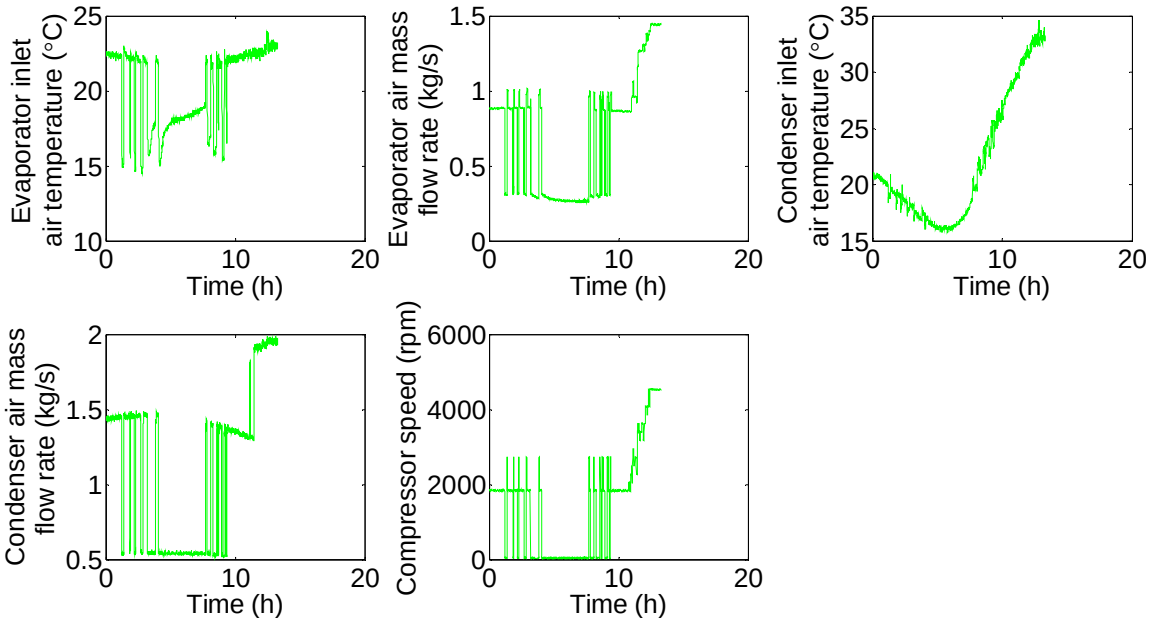
(b) Outputs

Figure 5.37: Condenser FCV model validation using Coleman 5T field cooling data 07/15/2012.

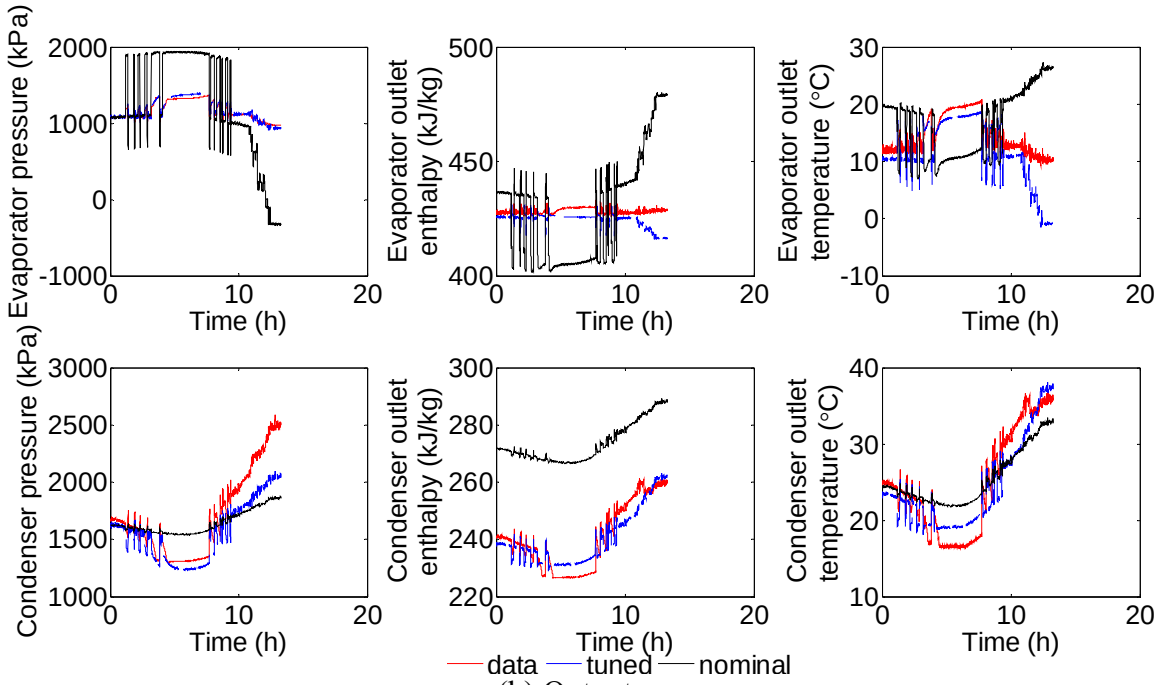
Complete Heat Pump Model

The complete heat pump model is also tuned and validated using Rheem 5T cooling and Coleman 5T cooling field data. The tuning parameters for the heat pump model include TXV coefficients, compressor map coefficients and heat exchanger internal and cross area of the tube, diameter, and length. The outputs used in the objective function include evaporator pressure, evaporator refrigerant outlet temperature, condenser pressure, and condenser refrigerant outlet temperature. The following are some model tuning and validation results. Table 5.4 lists the simulation errors.

In Figure 5.38, the inputs contain frequent compressor startups and shutdowns. Other transient inputs include the frequent changes in inlet air temperature and mass flow rates. These drastic and hard transients are also shown in the output Figure 5.38 (b). The black lines in Figure 5.38 (b) represent the nominal predicted outputs of the heat pump model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. The match is good except for the final portion of evaporator temperature, which also causes the slight mismatch of enthalpy in the end. The condenser outputs match is better than that of the evaporator, because the weighting factors (including the steady state and transients) for the condenser are set higher in the objective function. Note the match of steady states at the beginning of these outputs is also achieved simultaneously, which means simultaneous tuning is feasible for the complete heat pump model using the proposed method.



(a) Inputs



(b) Outputs

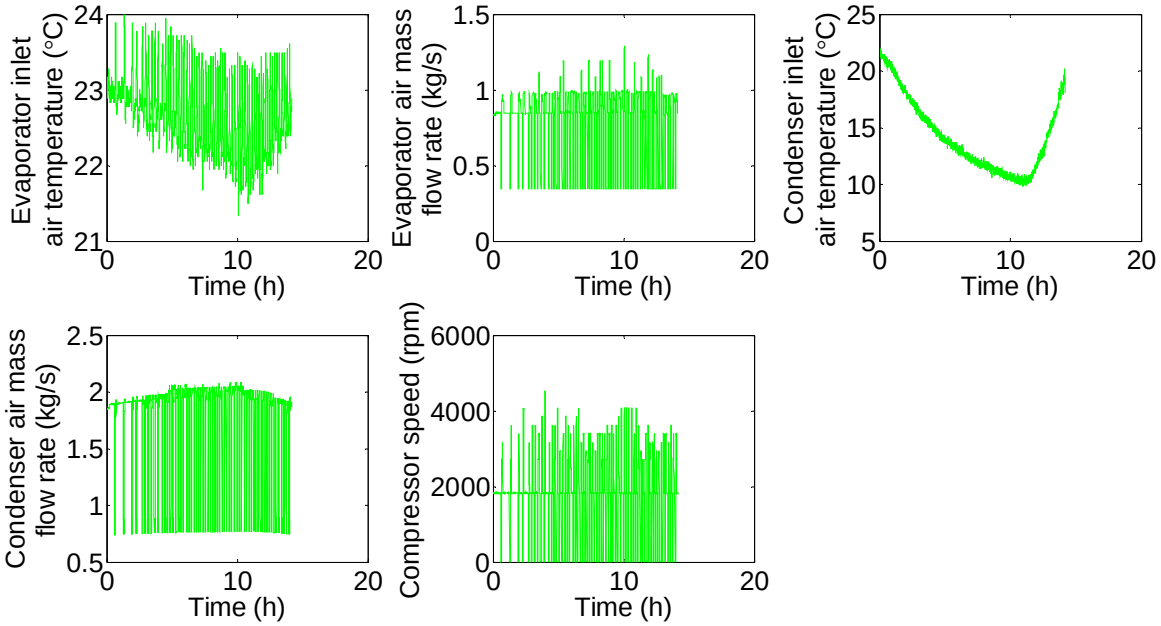
Figure 5.38: Complete HP model validation using Rheem 5T field cooling data 06/06/2011.

Table 5.4: Relative error of complete HP model validation using field data.

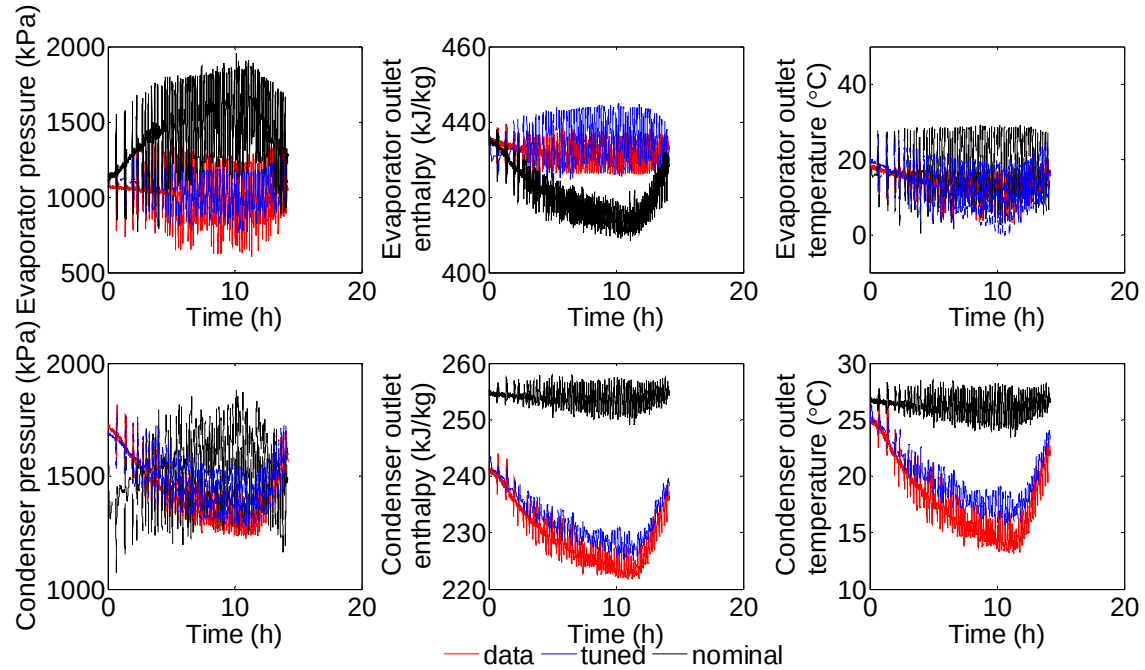
Data	Rheem 5T 06/06/2011	Coleman 5T 05/17/2012
Relative error (%)	12.73	10.37

In Figure 5.39, the inputs contain drastic and frequent compressor startups and shutdowns. This poses a challenge to parameter tuning. Other transient inputs include the drastic and frequent changes in inlet air temperature and mass flow rates. These drastic and hard transients are also seen in the output Figure 5.39 (b). The black lines in Figure 5.39 (b) represent the nominal predicted outputs of the heat pump model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data, and the final tuned outputs are shown in blue lines. The match is good for both the evaporator and condenser, as the weighting factors are equally distributed in the objective function. Note the match of steady states at the beginning of these outputs is also achieved simultaneously, which means simultaneous tuning is feasible for the complete heat pump model using the proposed method.

All of these complete heat pump model tuning and validation results indicate that the proposed parameter tuning and validation approach can indeed drive the predicted outputs toward the field data, even if the initial predicted outputs are distinct from the data, or the transients are drastic and frequent.



(a) Inputs



(b) Outputs

Figure 5.39: Complete HP model validation using Coleman 5T field cooling data 05/17/2012.

In each transient data set used in this section, it only contains one operating condition at the beginning. If multiple operating conditions are in the data set, it can be divided piecewise, with each piece containing one operating condition at the beginning. Parameter tuning and validation can then be applied on each of these pieces. Another way to solve this issue is to perform simultaneous tuning using multiple data sets with different operating conditions, to be shown in the following section.

Simultaneous Parameter Tuning Using Multiple Field Data Sets with Different Operating Conditions

The advantages of simultaneous parameter tuning and validation include:

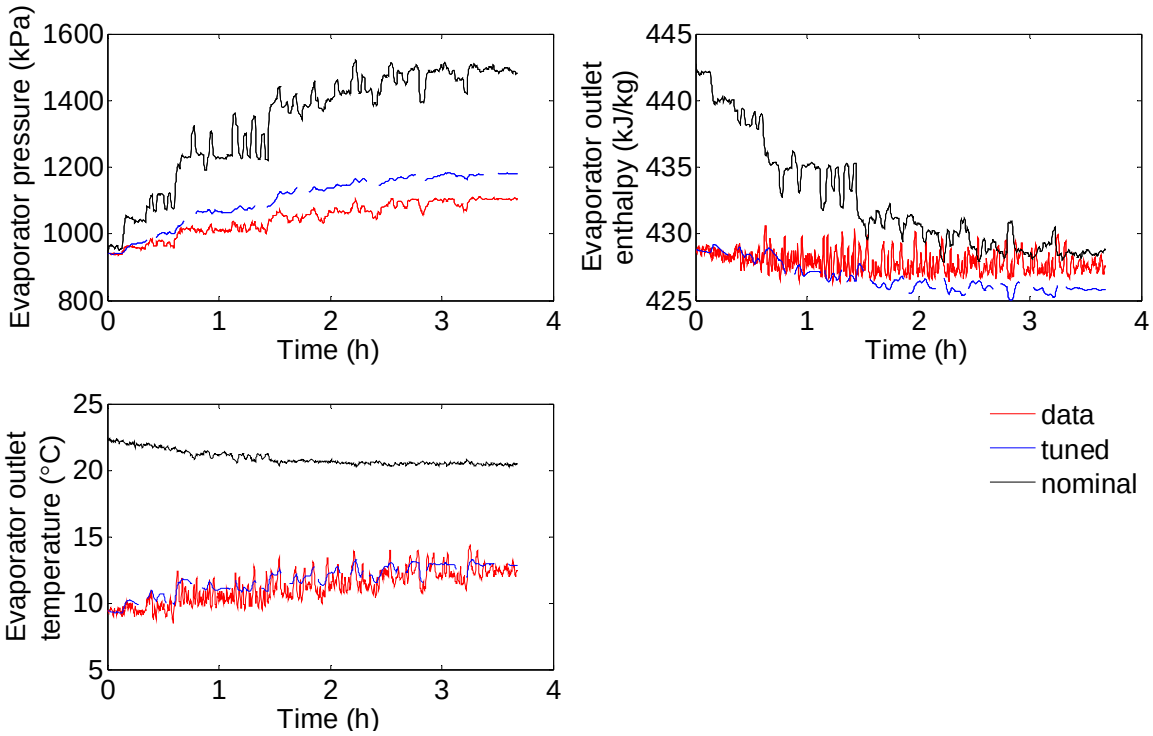
- Avoiding performing tedious individual data set tuning for multiple data sets with different operating conditions.
- Improving the robustness of tuned model. This means the same parameters obtained from tuning can also be used to predict accurate model outputs under other operating conditions.
- Reducing the number of tuning parameters by assuming some parameters are the same for different data sets.

However, simultaneous parameter tuning using multiple data sets is a challenge since it requires that both outputs start at their own steady state condition at the same time, and then matching the transients after that. This may be realized by having

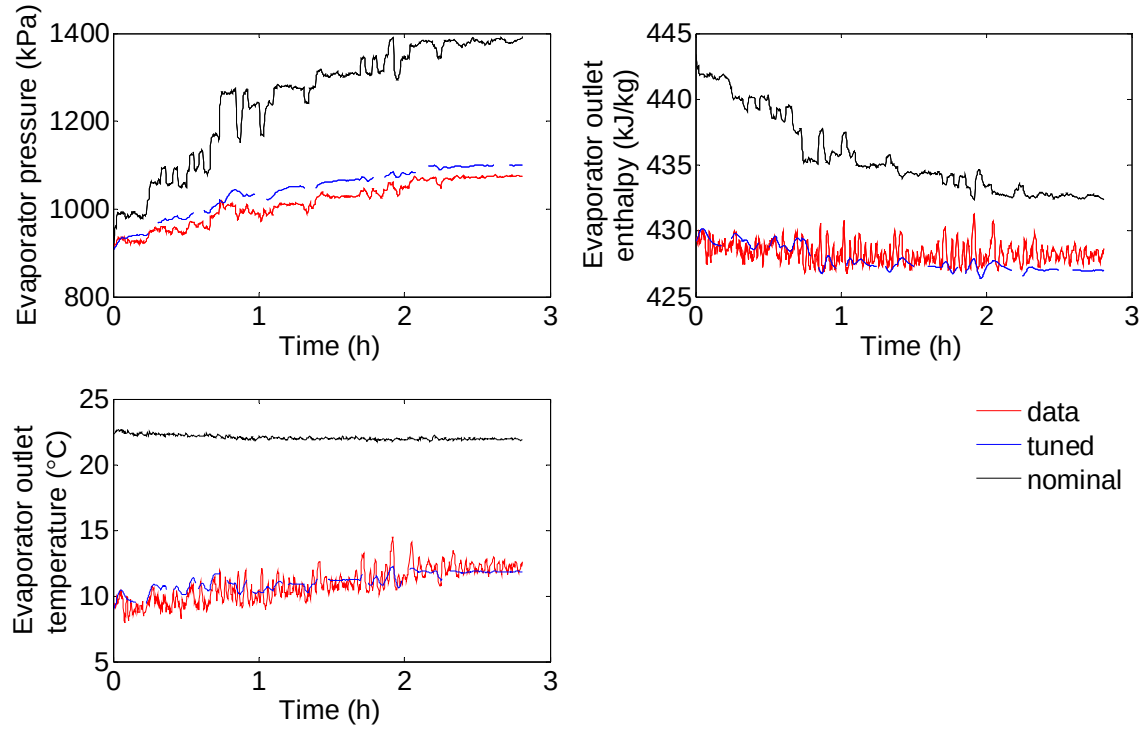
individual sets of tuning parameters for each data set, and then performing parameter tuning and optimization separately at the same time, i.e. multiple objective functions obtained by varying respective tuning parameters are optimized simultaneously. However, this approach is equivalent to performing the single data parameter tuning and validation simultaneously for multiple data sets. Therefore it will greatly increase the number of tuning parameters, as well as computation cost.

This section presents some good results of parameter tuning and validation by assuming tuning parameters to be the same for two data sets, which have distinct yet close operating conditions. The tuning parameters include the heat exchanger internal and cross area of the tube, diameter, length, TXV and compressor map coefficients. Among these parameters, the physical ones are assumed to be the same for the two data sets, and the different tuning parameters include the TXV coefficients and volumetric efficiency coefficients. This is because the prediction of mass flow rates, calculated by the coefficients, can be adjusted differently for the two data sets. In this scenario, the objective function described in Equation (3.16) is modified to include both field data sets.

The parameter tuning results are shown in Figure 5.40. The averaged error is 4.7% for data 05/30/2011, and 5.3% for 06/26/2011.



(a) Tuning results for Rheem 5T transient cooling data 05/30/2011



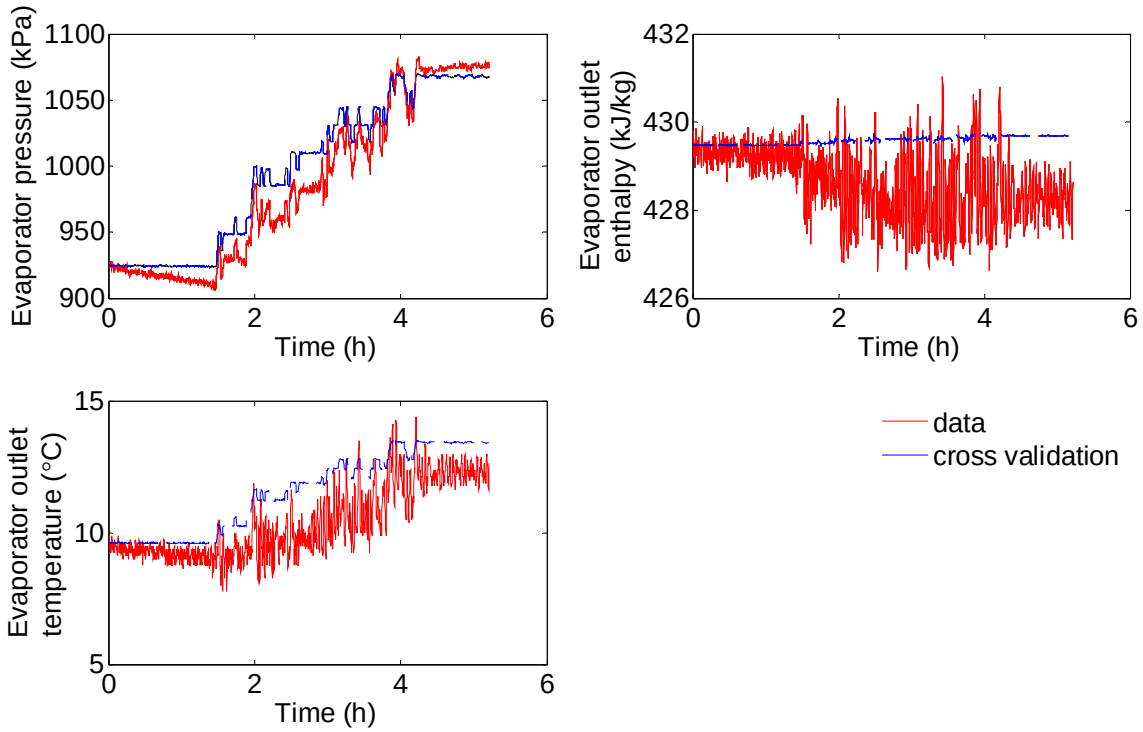
(b) Tuning outputs for Rheem 5T transient cooling data 06/26/2011

Figure 5.40: Evaporator FCV model multiple data validation using Rheem 5T transient cooling data 05/30/2011 and 06/26/2011.

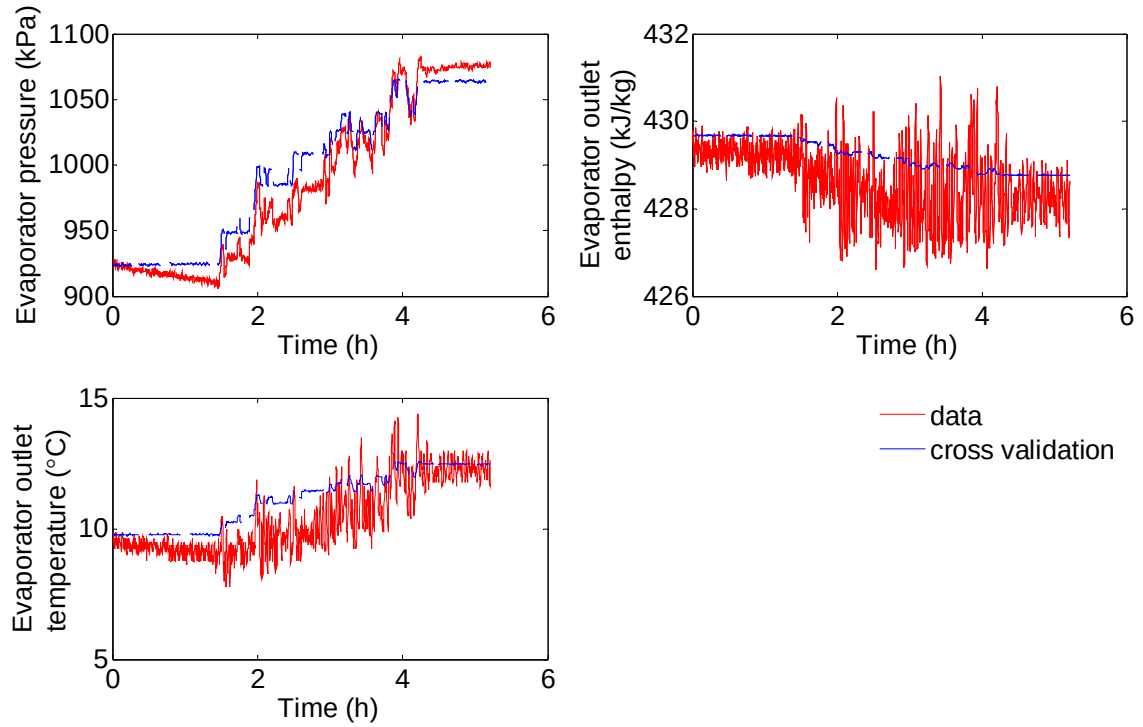
The black lines represent the nominal predicted outputs, which are distinct from the field data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the field data simultaneously for both data sets, and the final tuned outputs are shown in blue lines. The match is good for both data sets. Note the match of steady states at the beginning of these outputs is also achieved simultaneously for both data sets. This signifies that simultaneous tuning for multiple field data sets is feasible using the proposed method.

Cross Validation

In addition, the tuned solution obtained from the previous section was validated using another data set (Rheem 5T cooling data 20110705) with a different operating condition (yet close to the operating conditions of the two data sets used for parameter tuning). The results are shown in Figure 5.41. From the description of the previous section, there are two sets of tuned parameters, in which the physical parameters are the same for both data sets (05/30/2011 and 06/26/2011), while the TXV and compressor map coefficients are different. Therefore, the cross validation for another data set can use either of the two sets of tuned parameters.



(a) Using the tuned coefficients from 05/30/2011



(b) Using the tuned coefficients from 06/26/2011

Figure 5.41: Evaporator FCV model cross validation for Rheem 5T transient cooling data 07/05/2011 with tuned parameters obtained from Figure 5.40.

In Figure 5.41, (a) use the tuned parameters with the TXV and compressor map coefficients from transient data 201105030, and (b) use those from transient data 06/26/2011. The predictions are really close for either set of tuned parameters, as the averaged relative error is 5.6% and 7.8%, respectively. These results indicate that the tuned FCV evaporator model is effective in predicting the outputs with different operating conditions, implying the good robustness of tuned model.

CHAPTER VI

APPLICATION TO RESIDENTIAL CONDITIONED SPACE MODEL

The proposed parameter tuning and validation method is also applied on the residential conditioned space model, as described in Chapter II. In order to validate this model without actual data, a comparative model was created using EnergyPlus in order to validate the residential model created in MATLAB/Simulink.

This chapter first introduces the model established in EnergyPlus. The use of EnergyPlus with SketchUp and OpenStudio will be described. HVAC System Templates used in this model are also included.

Second, parameter tuning and validation using pseudo-steady state data validation is presented. Using the simulation outputs obtained from EnergyPlus, pseudo-steady state data is generated similarly to the method mentioned in Chapter V. The tuning and validation results are then discussed.

Finally, the transient data from EnergyPlus is used to validate the residential conditioned space model in detail. These validations include simultaneous steady state and transient data parameter tuning with and without a heat pump system.

Establishing a House Model Using EnergyPlus

The house model used in these tuning and validations was created in a program available from the U.S. Department of Energy called EnergyPlus [101]. This program allows users to model a house with or without an HVAC system, and thoroughly analyze

the performance and characteristic for a wide range of conditions. EnergyPlus is a software engine without a user-friendly interface, thus other programs must be used in conjunction with it. In this dissertation, the freeware software SketchUp was used to model the conditioned house in 3-D, including fenestrations [102]. The software suite OpenStudio connects SketchUp and EnergyPlus to each other, enabling an efficient workflow in setting up the house model [103]. Figure 6.1 illustrates the use of EnergyPlus, SketchUp and OpenStudio to establish a house model.

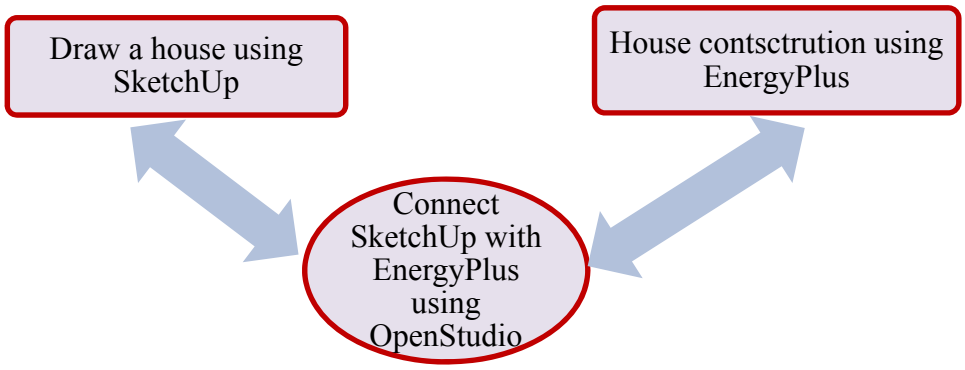


Figure 6.1: Setting up a house model in EnergyPlus.

The house model is assumed to have one zone, one room. The view of the residential conditioned space house is shown in Figure 6.2.

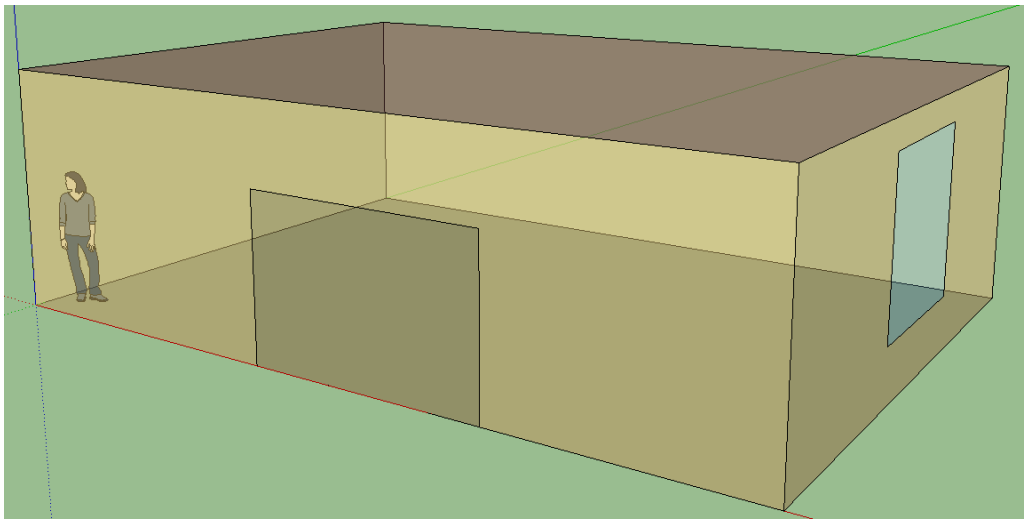


Figure 6.2: View of the residential conditioned space house.

The HVAC system is modeled using the HVAC Thermostat Template in EnergyPlus. The HVAC system is a single zone air-conditioning system. The templates allow the user to quickly assemble a basic system for simulation rather than define each and every component in the HVAC system. While this may cause some limitations on the complexity of the system, the templates are sufficient for the presentation in this dissertation. On the other hand, since the HVAC templates do not allow the control of the room humidity, the template file was ran first to generate an expidf file, which enables the users to make modifications to the house settings. The weather data used in this model is from Dayton International Airport. Table 6.1 shows some settings and dimensions of the house model in EnergyPlus.

Table 6.1: Inputs of EnergyPlus model (EnergyPlus configuration).

Variables	Room area	Window area	Fenestration	Infiltration	HVAC Control	Air Units
Values /Settings	70 m^2	10 m^2	Detailed	Design flow rate	Thermostat and humidistat	Ideal load air system

Parameter Tuning and Validation Using Steady State Data

Transient data were generated using the EnergyPlus model using a step size of 10 minutes for a whole year simulation. Steady state data were then generated from these transient data using the method similar to that described in Chapter V. A couple of steady state data points were selected to tune the model, including data in winter and summer. In these data points, the room temperature and the humidity are set to 21 ºC and 38%, which are controlled by the thermostat and humidistat in the EnergyPlus model.

In this validation, the tuning parameters include wall surface area, wall thickness, ceiling surface area, ceiling thickness, room width, floor surface area, total surface area of the windows and doors and room height. All the steady state data are assumed to have the same set of tuning parameters, thus they are tuned simultaneously to find the optimal tuning parameters. The results are shown below. These figures show that the predicted data match very well with the EnergyPlus data. The relative errors of the outputs are 4.91% for the cooling mode and 6.21% for the heating mode.

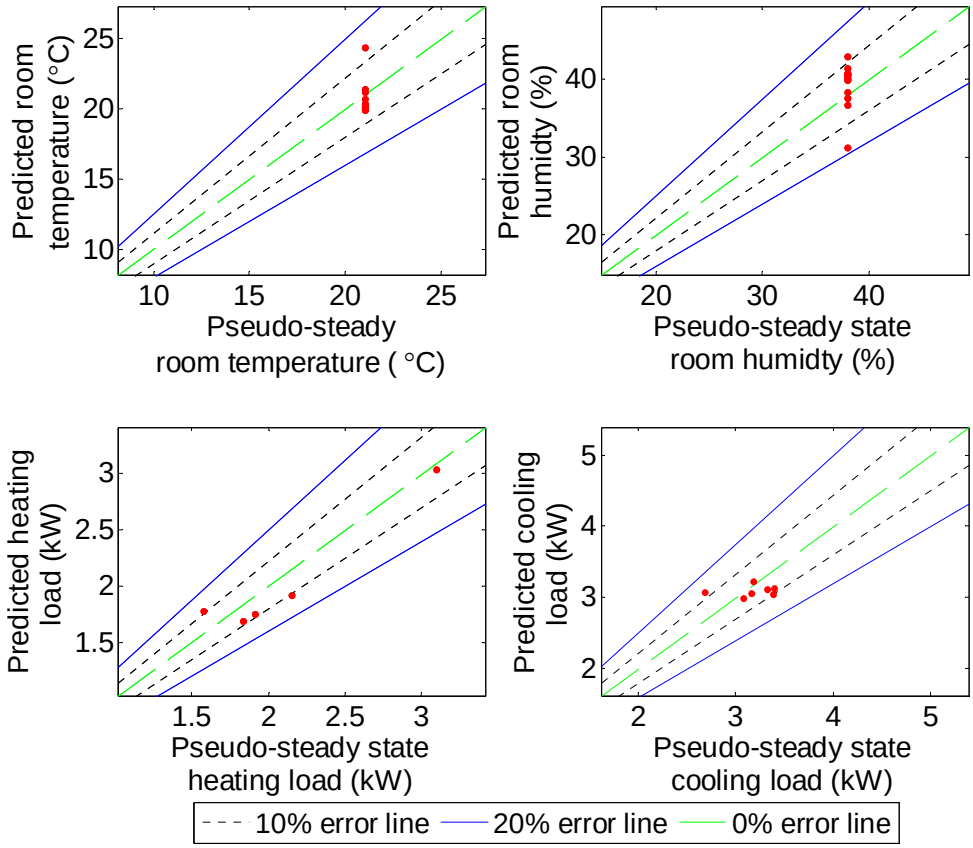


Figure 6.3: Residential model steady state validation.

Simultaneous Parameter Tuning and Validation Using EnergyPlus Data without Heat Pump System

Data with no heat pump system for the EnergyPlus model was generated in order to investigate the basic prediction of the residential model. Figure 6.4 and Figure 6.5 show the model inputs and the parameter tuning and validation results using ten days weather data in January.

In this parameter tuning and validation, physical parameters including wall surface area, wall thickness, ceiling surface area, ceiling thickness, room width, floor

surface area, total surface area of the windows and doors and room height were tuned. The averaged relative error is 8.7%.

In Figure 6.4, the inputs contain frequent transients. These transients are also shown in the output Figure 6.5. The black lines represent the nominal predicted outputs of the residence model, which are distinct from the data shown in the red lines. The proposed parameter tuning method drives the predicted outputs toward the transient data, and the final tuned outputs are shown in blue lines. The residential model is able to predict the room air temperature and humidity. Note the match of steady states at the beginning of these outputs is also achieved simultaneously, which means simultaneous tuning is feasible for the lumped parameter residence model using the proposed method.

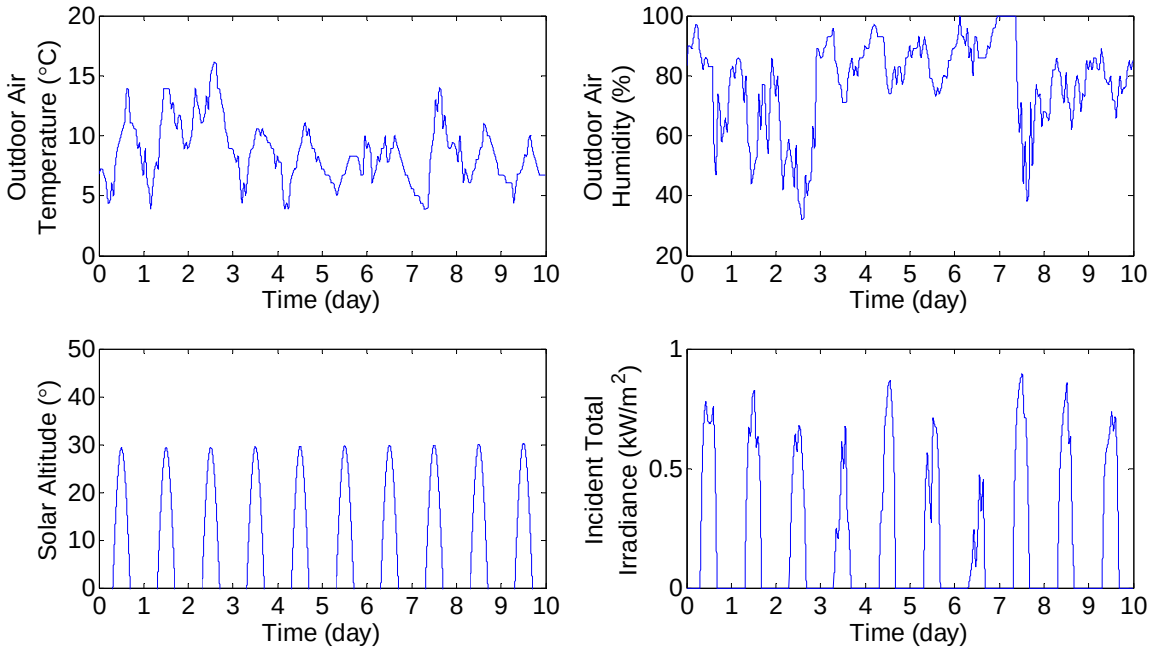


Figure 6.4: Ten days weather inputs to the residential model in January.

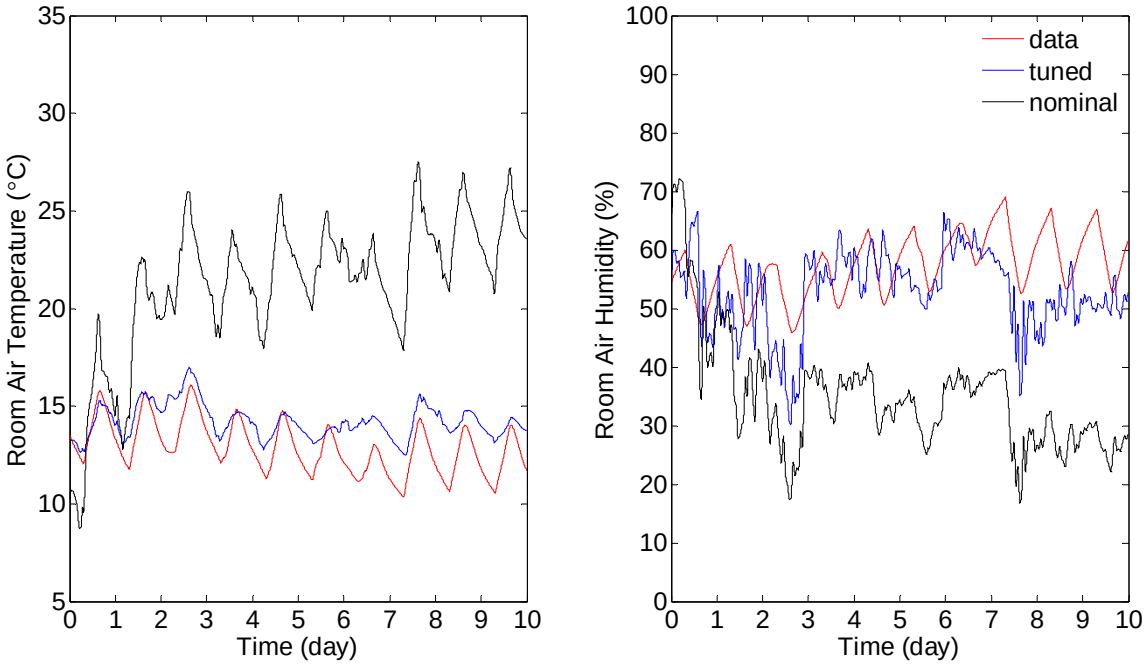


Figure 6.5: Residence model validation results without heat pump.

Simultaneous Parameter Tuning and Validation Using EnergyPlus Data with Heat Pump System

Two EnergyPlus data sets, i.e. ten days data in January and July, were selected to validate the residential model. In this validation, steady state and transient data were tuned together. The tuning parameters include wall surface area, wall thickness, ceiling surface area, ceiling thickness, room width, floor surface area, total surface area of the windows and doors and room height. The following are the tuning and validation results. In these validation results, the room temperature and the humidity are set to 21 ºC and 38%, which are controlled by the thermostat and humidistat in the EnergyPlus model, and a PID controller in the residential model.

For the heating mode, Figure 6.4 shows the inputs to the residential model. Figure 6.6 shows the corresponding validation results. The averaged relative error is 6.6%. In Figure 6.6, the proposed parameter tuning method drives the controlled outputs toward the data, and the final tuned outputs are shown in blue lines. The residential model is able to control the room air temperature and humidity.

For the cooling mode, Figure 6.7 shows the inputs to the residential model. Figure 6.8 shows the corresponding validation results. The averaged relative error is 4.2%. In Figure 6.8, the proposed parameter tuning method drives the controlled outputs toward the data, and the final tuned outputs are shown in blue lines. The residential model is able to control the room air temperature and humidity.

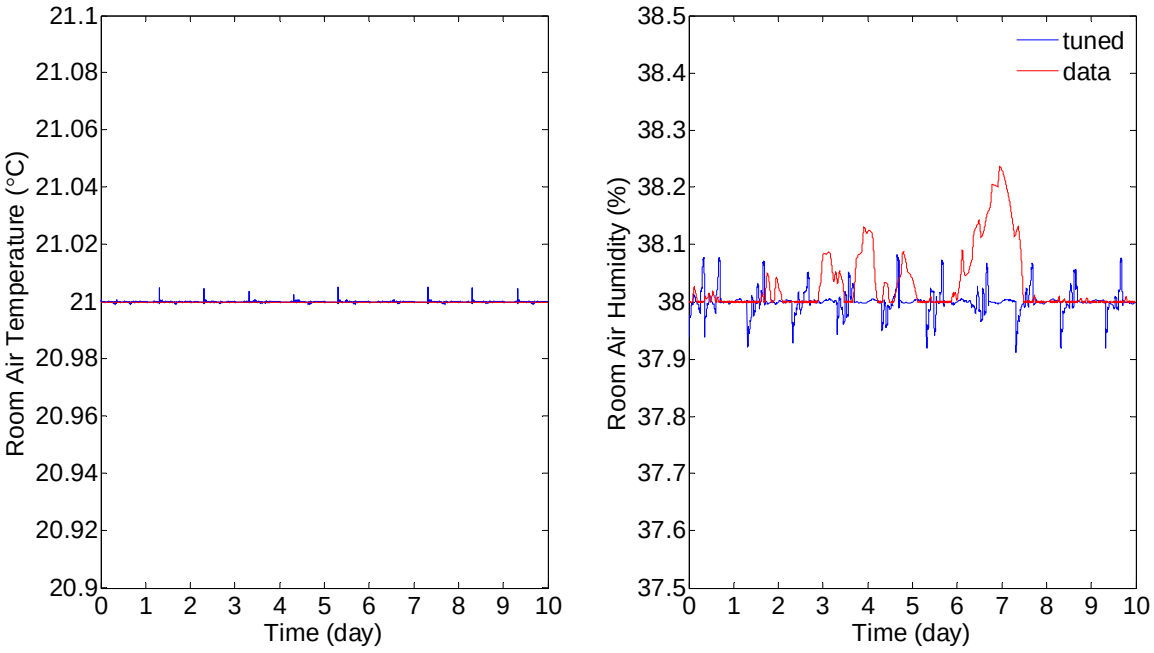


Figure 6.6: Residence model validation results with heat pump (heating).

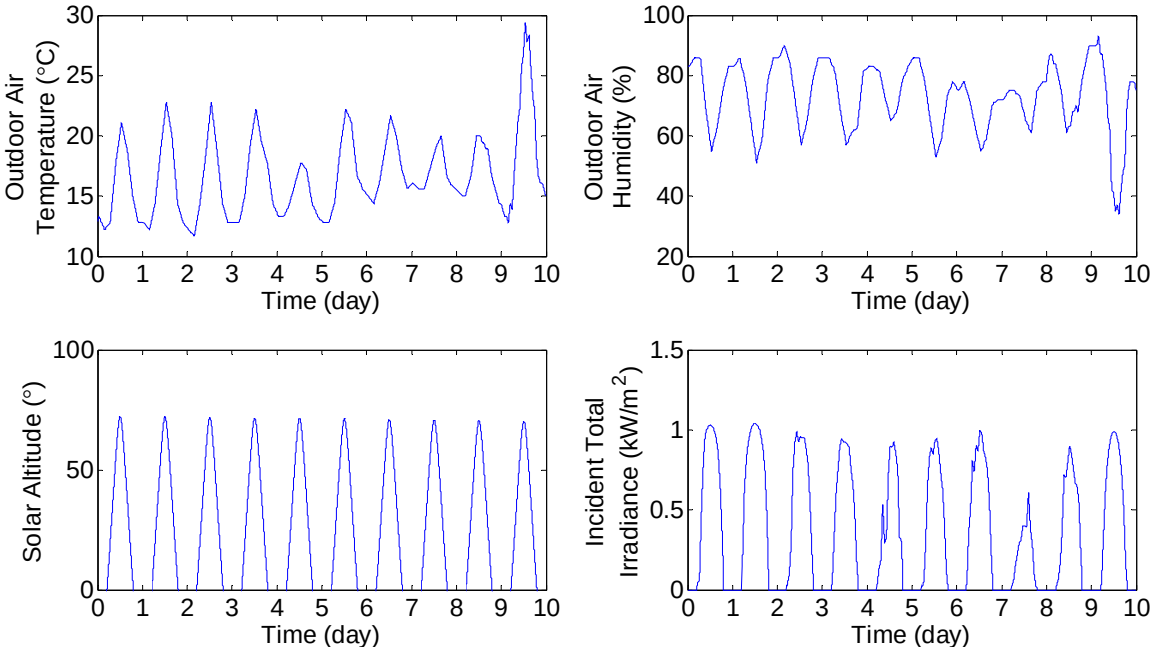


Figure 6.7: Ten days weather inputs to the residential model in July.

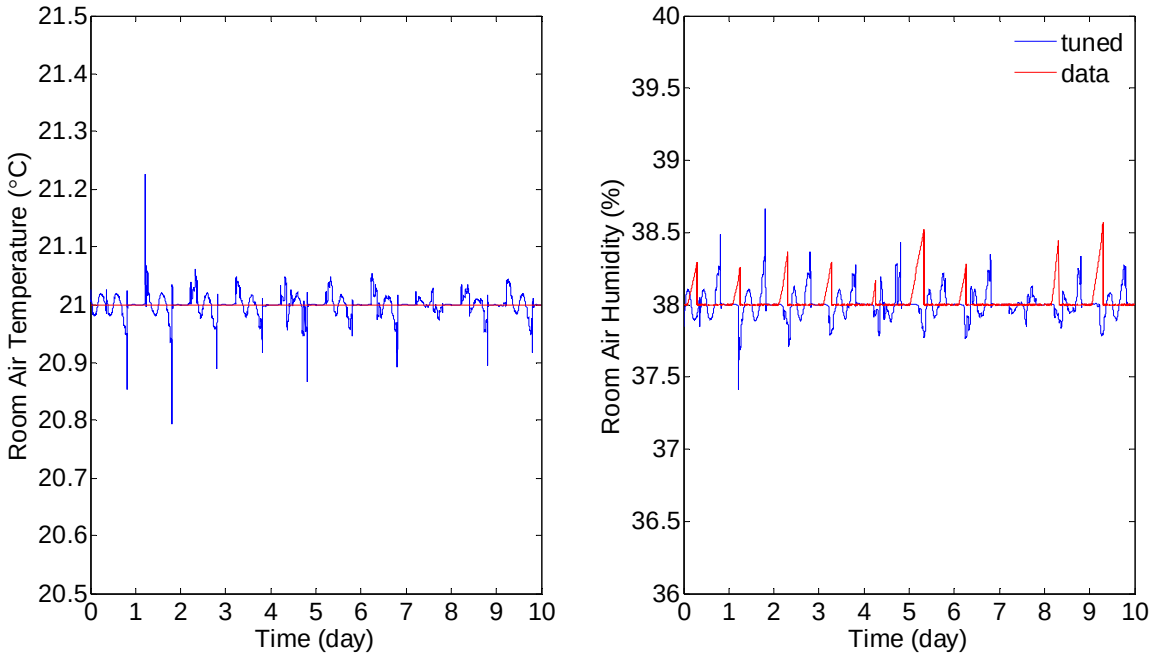


Figure 6.8: Residence model validation results with heat pump (cooling).

CHAPTER VII

CONCLUSION

The research presented in this dissertation has the ultimate goal of providing accurate dynamic vapor compression system models while reducing tedious effort in parameter tuning and model validation. While manual tuning is mainly used in this area to minimize the difference between model prediction and experimental measurements, developing effective automatic tuning of dynamic vapor compression system models can improve model accuracy, reduce computation cost and avoid tedious manual tuning.

The work presented herein explored this problem from three different perspectives. The first is to generate model outputs more quickly. This is critical since repeated simulations of the model are required during parameter tuning. The solution proposed in this dissertation is to use linearization and the discrete-time convolution to generate model outputs. This approach allows for an accurate prediction of the outputs as well as shortening the computation time. The second idea is to use wavelet decomposition to establish the objective function from the model outputs and data. Wavelets are capable of capturing the drastic transients in the data, retaining the information hidden in the time series. The last approach is to use a hybrid optimization algorithm to improve the tuning accuracy and further reduce the computation cost. Traditional gradient search method usually causes local convergence problems. If this issue is neglected, the parameter tuning and validation results can potentially imply incorrect and misleading conclusions. Therefore, a hybrid stochastic-deterministic

method is adopted to improve the tuned model robustness as well as reducing computation cost.

The proposed parameter tuning and model validation method has been verified and applied on several different dynamic models. These models include an HVAC system model with moving boundary (MB) heat exchanger models, a heat pump model with finite control volume (FCV) heat exchanger models, and a residential conditioned space model. The parameter tuning and validation results on the component and complete system model levels are promising, indicating that the proposed approach is effective in finding the optimal values of multiple parameters. The linearized model with discrete-time convolution is fast and accurate, increasing the total computation speed. This method is also an efficient means to tune the models using multiple data sets with different operating conditions simultaneously. Cross validation of the tuned parameters using other data sets further proves the robustness of the tuned models.

Future Work

The proposed method would be an instrumental tool in tuning the models in areas of aerospace, biochemical or automobile systems. When it is applied to different nonlinear models, the approximation of the nonlinear models is necessary. The linear discrete-time model proposed in this dissertation provides a cost-efficient and accurate way to represent the nonlinear model. Nevertheless, more accurate approximations of nonlinear models should be developed for parameter tuning and model validation. The use of reduced-order nonlinear or finite control volume models, for example, is possible.

The use of hybrid optimization algorithm is promising in this dissertation, but it requires an efficient transition point from the global search algorithm to the local search algorithm. Future studies should explore more efficient and automatic methods to determine this point. Wavelets are useful in capturing the full transient information in the data, and there are also some possibilities to develop less weighting-factor-dependent objective functions. This is especially true when simultaneously tuning models with distinct outputs using both steady state and transient data.

Finally, parameter tuning using multiple data sets can improve the model robustness within certain operating conditions. Future work may find the should seek methods to increase the range of the operating conditions in which the tuned results can be used for that model, i.e. improving the robustness of the tuned model to include more operating conditions.

REFERENCES

[1] Energy Information Administration, "International Energy Outlook 2013," Report # DOE/EIA-0484(2013), 2013. Available at: http://www.eia.doe.gov/oiaf/ieo/world.html/.

[2] Rasmussen, B. P., "Review Article: Dynamic Modeling of Vapor Compression Systems - Part II: Simulation Tutorial," *HVAC&R Research*, vol. 18(5), pp. 956-973, 2012.

[3] Rasmussen, B. P., "Review Article: Dynamic Modeling of Vapor Compression Systems - Part I: Literature Review," *HVAC&R Research*, vol. 18(5), pp. 934-955, 2012.

[4] Grald, E. W. and MacArthur, J. W., "A Moving-Boundary Formulation for Modeling Time-Dependent Two-Phase Flows," *International Journal of Heat & Fluid Flow*, vol. 13(3), pp. 266-272, 1992.

[5] He, X. D., Liu, S., and Asada, H., "Modeling of Vapor Compression Cycles for Multivariable Feedback Control of HVAC Systems," *American Society of Mechanical Engineers Journal of Dynamic Systems Measurement & Control*, vol. 119(2), pp. 183-191, 1997.

[6] Rasmussen, B. P. and Alleyne, A., "Control-Oriented Modeling of Transcritical Vapor Compression Systems," *American Society of Mechanical Engineers Journal of Dynamic Systems Measurement & Control*, vol. 126(1), pp. 54-64, 2004.

[7] Wedekind, G. L. and Stoecker, W. F., "Theoretical Model for Predicting Transient Response of Mixture-Vapor Transition Point in Horizontal Evaporating Flow," *Journal of Heat Transfer*, vol. 90, pp. 165, 1968.

[8] Narayanan, S., Srinivas, B., Pushpavanam, S., and Bhallamudi, S. M., "Nonlinear Dynamics of a Two-Phase Flow System in an Evaporator: The Effects of (i) A Time Varying Pressure Drop (ii) An Axially Varying Heat Flux," *Nuclear Engineering and Design*, vol. 178, pp. 279-294, 1997.

[9] Willatzen, M., Pettit, N. B. O. L., and Ploug-Sorensen, L., "A General Dynamic Simulation Model for Evaporators and Condensers in Refrigeration. Part I: Moving Boundary Formulation of Two-Phase Flows with Heat Exchange," *International Journal of Refrigeration*, vol. 21(5), pp. 398-403, 1998.

[10] Mithraratne, P., Wijeysundera, N. E., and Bong, T. Y., "Dynamic Simulation of a Thermostatically Controlled Counter-Flow Evaporator," *International Journal of Refrigeration*, vol. 23(3), pp. 174-189, 2000.

[11] Tummescheit, H. and Eborn, J., "Design of a Thermao-Hydraulic Model Library in Modelica^{TM} ," from proceedings of the 1998 European Simulation Multiconference, Manchester, UK, 1998.

[12] Bendapudi, S. and Braun, J. E., "A Review of Literature on Dynamic Models of Vapor Compressino Equipment," Report # 4036-5, American Society of Heating, Refrigerating and Air Conditioning Engineers, 2002.

[13] Bendapudi, S., "Development and Evaluation of Modeling Approaches for Transients in Centrifugal Chillers," Ph.D. Thesis, Department of Mechanical Engineering, Purdue University, West Lafayette, IN, 2004.

[14] Bendapudi, S., Braun, J. E., and Groll, E. A., "A Comparison of Moving-Boundary and Finite-Volume Formulations for Transients in Centrifugal Chillers," *International Journal of Refrigeration*, vol. 31, pp. 1437-1452, 2008.

[15] Hemami, T. L. and Dunn, W. E., "Development of a Transient System Model of Mobile Air-Conditioning Systems," Master Thesis, Department of Mechanical Science and Engineering, University of Illinois, Urbana Champaign, IL, 1998.

[16] Gupta, A., "Reduced Order Modeling of Heat Exchangers Using High Order Finite Control Volume Models," Master Record of Study, Department of Mechanical Engineering, Texas A&M University, College Station, TX, 2007.

[17] Dhar, M., "Transient Analysis of Refrigeration System," Ph.D. Dissertation, Department of Mechanical Engineering, Purdue University, West Lafayette, IN, 1978.

[18] Chi , J. and Didion, D., "A Simulation Model of the Transient Performance of a Heat Pump," *International Journal of Refrigeration*, vol. 5, pp. 176-184, 1982.

[19] Gruhle, W. D. and Isermann, R., "Modeling and Control of a Refrigerant Evaporator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 235-240, 1985.

[20] Wedekind, G. L., Bhatt, B. L., and Beck, B. T., "A System Mean Void Fraction Model for Predicting Various Transient Phenomena Associated with Two-Phase

Evaporating and Condensing Flows," *International Journal of Multiphase Flow*, vol. 4, pp. 97-114, 1978.

[21] Super Radiator Coils™, Products, 2014. Available at: http://www.srcoils.com/products/fluid/water-coils/.

[22] Wattelet, J. P., "Heat Transfer Flow Regimes of Refrigerants in a Horizontal-Tube Evaporator," Technical Report # 55, Air-Conditioning & Refrigeration Center, University of Illinois, Urbana-Champaign, IL, 1994.

[23] Dittus, F. W. and Boelter, L. M. K., "Heat transfer in automobile radiators of the tubular type," *International Communications in Heat and Mass Transfer*, vol. 12(1), pp. 3-22, 1985.

[24] Gnielinski, V., "New Equations for Heat and Mass Transfer in Turbulent Pipe and Channel Flow," *International Chemical Engineering*, vol. 16(2), pp. 359-367, 1976.

[25] Incropera, F. P. and Dewitt, D. P., *Fundamentals of Heat Transfer*, John Wiley and Sons, New York, pp. 399-407, 1981.

[26] Kandlikar, S. G., "An Improved Correlation for Predicting Two-Phase Flow Boiling Heat Transfer Coefficient in Horizontal and Vertical Tubes," from proceedings of the 1983 American Society of Mechanical Engineers National Heat Transfer Conference, Seattle, MA, 1983.

[27] Chen, J. C., "A Correlation for Boiling Heat Transfer to Saturated Fluids in Convective Flow," *Industrial and Engineering Chemistry, Process Design and Development*, vol. 5(3), pp. 322-329, 1966.

[28] Jallouk, P. A., "Two-Phase Flow Pressure Drop and Heat Transfer Characteristics of Refrigerants in Vertical Tubes," Ph.D. Dissertation, Department of Mechanical Engineering, University of Tennessee, Knoxville, TN, 1974.

[29] Mohr, V. and Runge, R., "Forced Convection Boiling of Neon in Horizontal Tubes," *Heat Transfer in Boiling*, Hahne, E. and Grigull, U., Eds., Hemisphere, Washington, D.C., pp. 307, 1977.

[30] Kandlikar, S. G., "A General Correlation for Saturated Two-Phase Flow Boiling Heat Transfer Inside Horizontal and Vertical Tubes," *Journal of Heat Transfer*, vol. 112, pp. 219-228, 1990.

[31] McCullagh, K. R., Green, G. H., and Chandra, S. S., "An Analysis of Chill Water Cooling Dehumidifying Coils Using Dynamic Relationships," *American Society of Heating, Refrigeration and Air Conditioning Engineers Transactions*, vol. 75 (2), pp. 200-209, 1969.

[32] Clark, D. R., "Type 12: Cooling or Dehumidifying Coil," HVACSIM+ Reference Manual, National Bureau of Standards, pp. 63-68, 1985.

[33] Chow, T. T., "Chilled Water Cooling Coil Models from Empirical to Fundamental," *Numerical Heat Transfer Part A- Applications*, vol. 32(1), pp. 63-83, 1997.

[34] Keir, M. C., Rasmussen, B. P., and Alleyne, A. G., "Parameteric Sensitivity Analysis and Model Tuning Applied to Vapor Compression Systems," from proceedings of the 2005 American Society of Mechanical Engineers International

Mechanical Engineering Congress and Exposition, pp. 1203-1212, Orlando, FL, 2005.

[35] Frank, P. M., *Introduction to System Sensitivity Theory*, Academic Press, New York, 1978.

[36] Cruz, J. B. Jr., *Feedback Systems*, McGraw-Hill, New York, 1972.

[37] Wilkie, D. E. and Perkins, W. R., "Essential Parameters in Sensitivity Analysis," Automatica, vol. 5, pp. 191-197, 1969.

[38] Rabehl, R. J., Mitchell, J. W., and Beckman, W. A., "Parameter Estimation and the Use of Catalog Data in Modeling Heat Exchangers and Coils," *HVAC&R Research*, vol. 5(1), pp. 3-17, 1999.

[39] Helvoirt, J., Jager, B., Steinbuch, M., and Smeulers, J., "Modeling and Identification of Centrifugal Compressor Dynamics with Approximate Realizations," from proceedings of the 2005 Institute of Electrical and Electronics Engineers Conference on Control Applications, pp. 1441-1447, Toronto, Canada, 2005.

[40] Nassif, N., Moujaes, S., and Zaheeruddin, M., "Self-Tuning Dynamic Models of HVAC System Components," *Energy and Buildings*, vol. 40, pp. 1709-1720, 2008.

[41] Rasmussen, B. P., "Dynamic Modeling and Advanced Control of Air Conditioning and Refrigeration Systems," Ph.D. Thesis, Department of Mechanical Science and Engineering, University of Illinois, Urbana-Champaign, IL, 2005.

[42] Zhou, X., "Dynamic Modeling of Chilled Water Cooling Coils," Ph.D. Thesis, Department of Mechanical Engineering, Purdue University, West Lafayette, IN, 2005.

[43] McKinley, T. L. and Alleyne, A. G., "Modeling Secondary Coolant Systems in THERMOSYS 3.0," Department of Mechanical Science and Engineering, University of Illinois, Urbana-Champaign, IL, 2007.

[44] Grossman, A. and Morlet, J., "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape," *Society for Industrial and Applied Mathematics Journal on Mathematical Analysis*, vol. 15, pp. 723-736, 1984.

[45] Morlet, J., Arens, G., Fourgeau, E., and Giard, D., "Wave Propagation and Sampling Theory 1, Complex Signal and Scattering in Multilayered Media", *Geophysics*, vol. 47, pp. 203-221, 1982.

[46] Gabor, D., "Theory of Communications," *Journal of Institute Electrical Engineers*, vol. 93, pp. 429-457, 1946.

[47] Meyer, Y., *Ondelettes et opérateurs I: Ondelettes*, Herman: Paris, France, 1990.

[48] Mallat, S., "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation", *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, vol. 11(7), pp. 674-693, 1989.

[49] Daubechies, I., "Ten Lectures on Wavelets," from the 1992 Society for Industrial and Applied Mathematics/CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, PA, 1992.

[50] Sweldens, W., "The Lifting Scheme: A Construction of Second Generation Wavelets," Technical Report # 1995-6, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, Columbia, SC, 1995.

[51] Candès, E. J., "Ridgelets: Theory and Applications," Ph.D. Dissertation, Department of Statistics, Stanford University, Stanford, CA, 1998.

[52] Do, M. N. and Vetterli, M., "The Finite Ridgelet Transform for Image Representation", *Institute of Electrical and Electronics Engineers Transaction on Image Processing*, vol. 12(1), pp. 16-28, 2003.

[53] Donoho, D. L. and Duncan, M. R., "Digital Curvelet Transform for Image Denoising", from proceedings of the 2000 International Society for Optics and Photonics Conference, vol. 4056, pp. 12-29, 2000.

[54] Leducq, D., "Hydraulic Noise Diagnostics Using Wavelet Analysis," from proceedings of the 1990 International Conference on Noise Control Engineering, pp. 997-1000, Goteborg, Sweden, 1990.

[55] Wang, W. J. and McFadden, P. D., "Application of the Wavelet Transform to Gearbox Vibration Analysis," *American Society of Mechanical Engineers, Petroleum Division*, vol. 52, pp. 13-20, 1993.

[56] Wang, W. J. and McFadden, P. D., "Application of Wavelets to Gearbox Vibration Signals for Fault Detection," *Journal of Sound and Vibration*, vol. 192, pp. 927-939, 1996.

[57] Wang, W. J. and McFadden, P. D., "Application of Orthogonal Wavelets to Early Gear Damage Detection," *Mechanical Systems and Signal Processing*, vol. 9, pp. 497-507, 1995.

[58] Newland, D. E., "Wavelet Analysis of Vibration, Part I: Theory," *Journal of Vibration and Acoustics, Transactions of the American Society of Mechanical Engineers*, vol. 116, pp. 409-416, 1994.

[59] Newland, D. E., "Wavelet Analysis of Vibration, Part 2: Wavelet Maps," *Journal of Vibration and Acoustics, Transactions of the American Society of Mechanical Engineers*, vol. 116, pp. 417-425, 1994.

[60] Newland, D. E., "Some Properties of Discrete Wavelet Maps," *Probabilistic Engineering Mechanics*, vol. 9, pp. 59-69, 1994.

[61] Newland, D. E., "Progress in the Application of Wavelet Theory to Vibration Analysis," *American Society of Mechanical Engineers, Design Engineering Division*, vol. 84, pp. 1313-1322, 1995.

[62] Newland, D. E., "Ridge and Phase Identification in the Frequency Analysis of Transient Signals by Harmonic Wavelets," *Journal of Vibration and Acoustics, Transactions of the American Society of Mechanical Engineers*, vol. 121, pp. 149-155, 1999.

[63] Gurley, K. and Kareem, A., "Applications of Wavelet Transforms in Earthquake, Wind and Ocean Engineering," *Energy Structures*, vol. 21, pp. 149-167, 1999.

[64] Chancey, V. C. and Flowers, G. T., "Identification of Transient Vibration Characteristics Using Absolute Harmonic Wavelet Coefficients," *Journal of Vibration and Control*, vol. 7, pp. 1175-1193, 2001.

[65] Kang, P. and Birtwhistle, D., "Condition Assessment of Power Transformer On-Load Tap-Changers Using Wavelet Analysis," *Institute of Electrical and Electronics Engineers Transactions on Power Delivery*, vol. 16, pp. 394-400, 2001.

[66] Wang, W. J., "Wavelets for Detecting Mechanical Faults with High Sensitivity," *Mechanical Systems and Signal Processing*, vol. 15, pp. 685-696, 2001.

[67] Gaberson, H. A., "Wavelet Use for Transient Event Detection in Machinery Vibration Signals," from proceedings of the 2001 International Modal Analysis Conference, vol. 2, pp. 998-1004, Orlando, FL, 2001.

[68] Chen, B. H., Wang, X. Z., Yang, S. H., and McGreavy, C., "Application of Wavelets and Neural Networks to Diagnostic System Development, 1, Feature Extraction," *Computers and Chemical Engineering*, vol. 23, pp. 899-906, 1999.

[69] Wang, X. Z., Chen, B. H., Yang, S. H., and McGreavy, C., "Application of Wavelets and Neural Networks to Diagnostic System Development, 2, An Integrated Framework and Its Application," *Computers and Chemical Engineering*, vol. 23, pp. 945-954, 1999.

[70] Lin, J. and Qu, L. S., "Feature Extraction Based on Morlet Wavelet and Its Application for Mechanical fault Diagnostics," *Journal of Sound and Vibration*, vol. 234, pp. 135-148, 2000.

[71] Yen, G. G. and Lin, K. C., "Conditional Health Monitoring Using Vibration Signatures," from proceedings of the 1999 Institute of Electrical and Electronics Engineers Conference on Decision and Control, vol. 5, pp. 4493-4498, Phoenix, AZ, 1999.

[72] Goumas, S., Zervakis, M., Pouliezos, A., and Stavrakakis, G. S., "Intelligent On-line Quality Control Using Discrete Wavelet Analysis Features and Likelihood Classification," from proceedings of the 2000 International Society for Optics and Photonics Conference, vol. 4072, pp. 500-511, 2000.

[73] Stavrakakis, G. S., Goumas, S. K., and Zervakis, M. E., "Classification of Washing Machines Vibration Signals Using Discrete Wavelet Analysis for Feature Extraction," *Institute of Electrical and Electronics Engineers Transactions on Instrumentation and Measurement,* vol. 51, pp. 497-508, 2002.

[74] Lu, C. J. and Hsu, Y. T., "Application of Wavelet Transform to Structural Damage Detection," *Shock and Vibration Digest*, vol. 32, pp. 50, 2000.

[75] Liu, B., Ling, S. F., and Meng, Q. F., "Machinery Diagnostics Based on Wavelet Packets," *Journal of Vibration and Control*, vol. 3, pp. 5-17, 1997.

[76] Dhar, A., Reddy, T. A., and Claridge, D., "An Artificial Neural Network with Wavelet Basis Functions to Predict Hourly Heating and Cooling Energy Use in Commercial Buildings with Temperature as the Only Weather Variable," from proceedings of the 1995 American Society of Mechanical Engineers International Solar Energy Conference, , vol. 1, pp. 145-156, Mawuii, 1995.

[77] Fugal, D. L., *Conceptual Wavelets in Digital Signal Processing*, Space & Signals Technical Publishing: San Diego, CA, 2009.

[78] Chui, C. K., *Wavelets: A Mathematical Tool for Signal Analysis*, 3rd ed., Society for Industrial and Applied Mathematics: Philadelphia, PA, 1997.

[79] Daubechies I., "Orthonormal Basis of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, pp. 909-996, 1988.

[80] Stromberg, J. O., "A Modified Haar System and Higher Order Spline Systems," from proceedings of the 1982 Conference on Harmonic Analysis in Honor of Antoni Zygmund (The Wadsworth Mathematics Series), vol. 2, pp. 475-493, Cortana, Italy, 1982.

[81] Meyer, Y., "Principe D'incertitude, Bases Hilbertiennes et Algkbres D'operateurs," *Seminuire Bourbaki*, vol. 28, pp. 209-223, 1985.

[82] Lemarie, P. G., "Une Nouvelle Base E'ondelettes de L^2(IR)," *Journal of Pure and Applied Mathematical Sciences*, vol. 67, pp. 227-238, 1988.

[83] Battle, G., "A Block Spin Construction of Wavelets. Part I Lemarie Functions," *Communications in Mathematical Physics*, vol. 110, pp. 601-615, 1987.

[84] Cohen, A., Daubechies, I., and Feauveau, J. C., "Biorthogonal Bases of Compactly Supported Wavelets," Technical Report # 1217-900529-07, AT&T Bell Laboratory, Murray Hill, NJ, 1990.

[85] Vetterli, M. and Herley, C., "Wavelets and Filter Banks: Relationships and New Results," from proceedings of the 1990 Institute of Electrical and Electronics

Engineers International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, 1990.

[86] Strang, G., "Wavelets and Dilation Equations: A Brief Introduction," *Society for Industrial and Applied Mathematics Review*, vol. 31(4), pp. 614-627, 1989.

[87] Swann, W. H., "A Survey of Nonlinear Optimization Techniques," *Federation of European Biochemical Societies Letters*, vol. 2, pp. S40-S55, 1969.

[88] Lewis, R. M., Torczon V., and Trosset, M. W., "Direct Search Methods: Then and Now," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191-207, 2000.

[89] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press: Cambridge, UK, 1992.

[90] Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed., Addison-Wesley Longman Publishing: Boston, MA, 1989.

[91] Caldas, L. G. and Norford, L. K., "Genetic Algorithms for Optimization of Building Envelopes and the Design and Control of HVAC Systems," *Journal of Solar Energy Engineering*, vol. 125, pp. 343-351, 2003.

[92] Chelouah, R. and Siarry, P., "Genetic and Nelder-Mead Algorithms Hybridized for a More Accurate Global Optimization of Continuous Multi-minima Functions," *European Journal of Operational Research*, vol. 148(2), pp. 335-348, 2003.

[93] Yen, J., Liao, J. C., Lee, B. J., and Randolph, D., "A Hybrid Approach to Modeling Metabolic Systems Using a Genetic Algorithm and Simplex Method," *Institute of Electrical and Electronics Engineers Transactions on System, Man and Cybernetics. Part B: Cybernet*, vol. 28(2), pp. 173-191, 1998.

[94] Zhang, H., Zhang, L. Z., and Meng, F., " Reactive Power Optimization Based on Genetic Algorithm," from proceedings of the 1998 International Conference on Power System Technology, vol. 2, pp, 1448-1453, Beijing, China, 1998.

[95] Hibbert, D. B., "Genetic Algorithms in Chemistry," *Chemometrics and Intelligent Laboratory Systems*, vol. 19, pp. 277-293, 1993.

[96] Leardi, R., "Genetic Algorithms in Chemometrics and Chemistry: A review," *Journal of Chemometrics*, vol. 15, pp. 559-569, 2001.

[97] Lucasius, C. B. and Kateman, G., "Understanding and Using Genetic Algorithms: Part 1. Concepts, Properties and Context," *Chemometrics and Intelligent Laboratory Systems*, vol. 19, pp. 1-33, 1992.

[98] Lucasius, C. B. and Kateman, G., "Understanding and Using Genetic Algorithms: Part 2. Representation, Configuration and Hybridization," *Chemometrics and Intelligent Laboratory Systems*, vol. 25, pp. 99-145, 1994.

[99] Fernandez, M. R., Mendes, P., and Banga, J. R., "A Hybrid Approach for Efficient and Robust Paramter Estimation in Biochemical Pathways," *BioSystems*, vol. 83, pp. 248-265, 2008.

[100] Maeder, M., Neuhold, Y. M., and Puxty, G., "Application of A Genetic Algorithm: Near Optimal Etimation of the Rate and Equilibrium Constants of

Complex Reaction Mechanisms," *Chemometrics and Intelligent Laboratory Systems*, vol. 70, pp. 193-203, 2003.

[101] U. S. Dept. of. Energy. EnergyPlus Energy Simulation Software, 2013. Available at: http://apps1.eere.energy.gov/buildings/energyplus/.

[102] Trimble Buildings. SketchUp Software, 2014. Available at: http://www.sketchup.com/.

[103] U. S. Dept. of. Energy. OpenStudio Software, 2013. Available at: http://apps1.eere.energy.gov/buildings/energyplus/openstudio.cfm/.

APPENDIX A

One Sample Tuning m-file in Matlab/Simulink

```
clear all
close all
tStart=tic;
matlabpool open 2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.0 Load data and define parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global RefProp CompProp
global DataSSCooling
load('Outputs_EVAP_ACC_AllData_SS_TuningTogether_AllVariables',
'Parameter_tuned','Operating_Conditions_store')
load DataSSCooling_SSFilename
filename='20110719';
load (filename)
load ([filename,'_SSCoolingRecord'])

load DataSSCooling_95
load RefProp_R410A
load CompProp_Cooling
load v_TXV
load m_air_coefficient
index = 10; %%SS validation index
```

```
N =  length(Data.rpm);
directory
={'C:\Users\Shuangshuang\Documents\MATLAB\Emerson\5TValidation\ModelValidation\Components_
EMERSON_5T\EVAP_ACC\TransientValidation\Figures\';

'C:\Users\Shuangshuang\Documents\MATLAB\Emerson\5TValidation\ModelValidation\Components_E
MERSON_5T\EVAP_ACC\TransientValidation\OutputData\'};
Name_outputs = {'Pe_{ro} (kPa)' 'He_{ro} (kJ/kg)' 'Te_{ro} (C)' 'm_valve (kg/s)' 'm_comp (kg/s)' 'He_ri
(kJ/kg)'};
Name_inputs = {'Pc_{ro} (kPa)' 'rpm' 'Te_{ai} (C)' 'me_{air} (kg/s)'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2.0 Inputs to the evaporator from the data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ts = 10;
tf  = (N - (lb(1)))*10;

Pc_ro_u = Data.Pc(lb(1):N);
Hc_ro_u = Data.Hc_ro(lb(1):N);

rpm_u  = Data.rpm(lb(1):N);
Hk_ri_u = Data.Hk_ri(lb(1):N);

Te_ai_u = Data.Te_ai(lb(1):N);
me_air_u = a_e_cooling*Data.me_air(lb(1):N);
```

```
Pe_ro_data   = Data.Pe(lb(1):N);

He_ro_data   = Data.He_ro(lb(1):N);

Te_ro_data   = Data.Te_ro(lb(1):N);

Te_ao_data   = Data.Te_ao(lb(1):N);

He_ri_data   = Data.Hc_ro(lb(1):N);

FieldData = [Pe_ro_data He_ro_data Te_ro_data He_ri_data];

Inputs = [Pc_ro_u rpm_u Te_ai_u me_air_u];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 2.0 Load operating conditions and physical parameters

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 2.1 EVAP

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Cpe_ext      = 1.005;                                  %Specific heat of external fluid

Slip_e         = 4;                                  %slip ratio in the evaporator

Te_a_mu      = 0.5;                                  %weighting factor for calculating average external fluid

temperature

n_regions = 20;

Mass_e        = 60.759/n_regions      *ones(n_regions,1);%Total mass of heat exchanger

Cpw_e         = 0.7008              *ones(n_regions,1); %Specific heat of heat exchanger material

Ae_o          =273.966/n_regions *ones(n_regions,1);      %Total external surface area of the heat exchanger

Diameter_e    = 0.023968            *ones(n_regions,1);  %Hydraulic diameter of refrigerant passage

Ae_i          = 2.943/n_regions      *ones(n_regions,1);  %Total Internal surface area of the heat exchanger

Ae_cs         =4.425e-004          *ones(n_regions,1);   %Cross sectional area of refrigerant passage

Le_total      = 13.1784;                            %Total length of refrigerant passage
```

```
%   Colburn J-factor data
CJF.Re_data  = [500 600 800 1000 1200 1500 2000 2500 3000 4000 5000 6000 8000 10000];
CJF.jH_data  = [0.014 0.013 0.012 0.0105 0.0099 0.009 0.008 0.0073 0.0068 0.006 0.0055 0.005 0.0046
0.0041];
CJF.sigma    = 0.500;
CJF.Dh       = 0.023968;%0.015461;
CJF.Afr      = 2.862;%1.6025;

%   Air
PGW.mu_T    = [296.65; 313.05; 333.1; 353.05; 363.3; 373.45; 384.15; 394.55; 411.75; 429.55; 449.55];
% air  temperature in Kelvin
PGW.mu_data = 1e-5*[1.838; 1.916; 2.01; 2.1; 2.146; 2.191; 2.238; 2.282; 2.355; 2.429; 2.511];          %
air  viscosity
PGW.k_T     = [299.6; 322.1; 347.2; 372.1; 396.4; 420.4; 440.4];                    % air temperature in
Kelvin
PGW.k_data  = 1e-2*[2.635; 2.801; 2.981; 3.155; 3.321; 3.482; 3.614];                % air conductivity
PGW.Cp_T    = [288.6; 299.7; 310.8; 321.9; 333.0; 344.1; 355.2; 366.3; 377.4; 388.5];     % air
temperature in Kelvin
PGW.Cp_data = [1.006; 1.007; 1.007; 1.008; 1.008; 1.009; 1.01; 1.011; 1.012; 1.013];      % air specific
heat
Ext_Fluid    = PGW;                                %external fluid air

%Accumulator Conditions
V_acc        = 15.25*0.0254*pi*(5*0.0254/2)^2;                          %Accumulator volume
UA_acc       = 0.1;                              %Heat transfer coefficient for Accumulator
minv_acc     = 0.2922;                              %Refrigerant mass in the Accumulator
```

```
ext_flow_var = 1;
HX_var = 1;

me_air       =Operating_Conditions_store(index, 1); %0.1;%2.335;                    %mass flow rate
of external fluid at evaporator
Pe           = Operating_Conditions_store(index, 3);                     %pressure of refrigerant at
evaporator
He_ri        = Operating_Conditions_store(index, 4);                       %enthalpy of the refrigerant at
the evaporator inlet
Te_ai        = Operating_Conditions_store(index, 5);                       %temperature of the air at the
evaporator inlet
Te_ao        =Operating_Conditions_store(index, 6);                      %temperature of the air at the
evaporator outlet

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2.2 TXV
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c1           = v_TXV(1);      %Operating range vector of percentage valve opening
c2            = v_TXV(2);     %Operating range vector of valve pressure difference
tau_b =10;
Slip_v = 4;
Pv_ri        = DataSSCooling.Pc_l(index);%DataSSCooling.Pc_l(index);
%valve inlet pressure
Hv_ri        = DataSSCooling.Hc_ro(index);                               %valve inlet enthalpy
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 2.3 Comp
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Vk            = 3.83457E-05;          %Minimum percentage valve opening
tau_k          = 200;                %Time constant for evolution of the refrigerant enthalpy at
compressor outlet
RPM_vector     = CompProp.rpm;           %Operating range vector of compressor speed
P_ratio_vector  = CompProp.Pr;            %Operating range vector of compressor pressure ratio
eta_v_matrix    = CompProp.eta_v;         %Interpolation matrix for volumetric efficiency,
?_vol=f(rpm,P_ratio)
eta_a_matrix    = CompProp.eta_a;         %Interpolation matrix for adiabatic efficiency,
?_k=f(rpm,P_ratio)

% Compressor %outlet condition is unreasonable
Pk_ro    = DataSSCooling.Pc(index);%mean(Data.Pc(lb(1):lb(1)+95));
%compressor outlet pressure
Tk_ri    = DataSSCooling.Tk_ri(index);                    %compressor inlet temperature
rpm      = DataSSCooling.rpm(index); %mean(Data.rpm(lb(1):lb(1)+95));%
mean(Data.Tk_ri(lb(1):lb(1)+95));%                    %compressor rpm
Hk_ri    = DataSSCooling.Hk_ri(index); %mean(Data.Hk_ri(lb(1):lb(1)+95));%
%refrigerant enthalpy at the inlet of compressor

%  Estimation of operating mass flow rate
Pk_ri_u = Pe_ro_data;
Pk_ro_u = Pc_ro_u;
Hf      = interp1(RefProp.Psat,RefProp.Hf,Pk_ri_u);           % Enthalpy for saturated liquid inlet
conditions
```

```
Hg      = interp1(RefProp.Psat,RefProp.Hg,Pk_ri_u);                    % Enthalpy for saturated vapor inlet
conditions
Rhof    = interp1(RefProp.Psat,RefProp.Rhof,Pk_ri_u);                  % Density for saturated liquid inlet
conditions
Rhog    = interp1(RefProp.Psat,RefProp.Rhog,Pk_ri_u);                  % Density for saturated vapor inlet
conditions

% Assign refrigerant density based on fluid phase
if Hk_ri_u < Hf                                                        % For sub-cooled inlet refrigerant
   rho_k = interp2(RefProp.Hl,RefProp.P,RefProp.Rhol_ph,Hk_ri_u,Pk_ri_u);
elseif Hk_ri_u > Hg                                                    % For superheated inlet refrigerant
   rho_k = interp2(RefProp.Hv,RefProp.P,RefProp.Rhov_ph,Hk_ri_u,Pk_ri_u);
else                                                                   % For two-phase inlet refrigerant
   quality = (Hk_ri_u-Hf)./(Hg-Hf);                                    % Refrigerant quality
   rho_k = inv( (1-quality)*inv(Rhof) + (quality).*inv(Rhog) );
end

P_ratio     = Pk_ro_u./Pk_ri_u;
eta_v       = interp2(CompProp.rpm,CompProp.Pr,CompProp.eta_v',rpm_u,P_ratio);
% plot(isnan(eta_v))
P_ratio(P_ratio<1)          = 1;
eta_v       = interp2(CompProp.rpm,CompProp.Pr,CompProp.eta_v',rpm_u,P_ratio);
mdot_k      = rpm_u./60.*Vk.*rho_k.*eta_v;
mdot_e      = DataSSCooling.mdot_e(index);                             %mass flow rate of refrigerant at
evaporator
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 3.0 Call Optimization Function, find parameters that minimize the error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Physical = [  Parameter_tuned(1) Parameter_tuned(2) Parameter_tuned(3) Parameter_tuned(4)*0.01
Parameter_tuned(5) Vk Cpe_ext c2*1e9 ];%Parameter_tuned(1) Parameter_tuned(2) Parameter_tuned(3)
Parameter_tuned(4) Parameter_tuned(5)Ae_cs(1) Ae_i(1) Ae_o(1) Le_total Diameter_e(1)
Parameter_guess = [c1*1e6 ];%];% load tuned parameters from SS tuning results
ub_P             = [Inf  ];
lb_P             = [-Inf  ];
hybridopts = optimset('Display','off','Algorithm','interior-point');
OPTIONS        =
gaoptimset('MutationFcn',@mutationadaptfeasible,'HybridFcn',{@fmincon,hybridopts});

% %TEST
[y_var_nom,sss_fval_nom] =
fun_solution_FCVEvapAcc_Transient(Parameter_guess,Physical,n_regions,Pe,me_air,Te_ai,Te_ao,Slip_e
,Mass_e,Cpw_e,Te_a_mu,He_ri,CJF,Ext_Fluid,mdot_e,ext_flow_var,HX_var,minv_acc,V_acc,ts,tf,Te_ai
_u,me_air_u,...
rpm,Pk_ro,Hk_ri,rpm_u,Hk_ri_u,RPM_vector,P_ratio_vector,eta_v_matrix,eta_a_matrix,...
tau_b,Slip_v,Pv_ri,Hv_ri,Pc_ro_u,Hc_ro_u);
figure
for i=1:1:3
subplot(2,2,i)
plot(FieldData(:,i),'r')
hold on
plot(y_var_nom(:,i),'k')
legend('data', 'nominal',15);
```

```
legend('boxoff');
xlabel('Time (10s)','fontsize',16);
ylabel(Name_outputs{i},'FontSize',15)
set(gca,'FontSize',15);
end
hold off

figure
plot(y_var_nom(:,4))
hold on
plot(y_var_nom(:,5),'r')
hold off
% % TEST

% % Using Hybrid Algorithm to find the parameters
% Pass parameters that do not need to be tuned using anonymous functions
f =
@(ParameterTuning)fun_FCVEvapAcc_Transient(ParameterTuning,Physical,n_regions,Pe,me_air,Te_ai,
Te_ao,Slip_e,Mass_e,Cpw_e,Te_a_mu,He_ri,CJF,Ext_Fluid,mdot_e,ext_flow_var,HX_var,minv_acc,V_a
cc,ts,tf,Te_ai_u,me_air_u,...
rpm,Pk_ro,Hk_ri,rpm_u,Hk_ri_u,RPM_vector,P_ratio_vector,eta_v_matrix,eta_a_matrix,...
tau_b,Slip_v,Pv_ri,Hv_ri,Pc_ro_u,Hc_ro_u,...
    FieldData);
[Parameter_tuned, ObjFcn] = ga(f,1,[],[],[],[],lb_P,ub_P,[],OPTIONS);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % 4.0 Obtain the simulation results with new parameters, compare it with data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 [y_var,sss_fval] =
fun_solution_FCVEvapAcc_Transient(Parameter_tuned,Physical,n_regions,Pe,me_air,Te_ai,Te_ao,Slip_e
,Mass_e,Cpw_e,Te_a_mu,He_ri,CJF,Ext_Fluid,mdot_e,ext_flow_var,HX_var,minv_acc,V_acc,ts,tf,Te_ai
_u,me_air_u,...
rpm,Pk_ro,Hk_ri,rpm_u,Hk_ri_u,RPM_vector,P_ratio_vector,eta_v_matrix,eta_a_matrix,...
tau_b,Slip_v,Pv_ri,Hv_ri,Pc_ro_u,Hc_ro_u);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % 5.0 Outputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
error_temp = zeros(1,3);
for i=1:3
    error_temp(i) = norm(FieldData(:,i)-y_var(:,i))/norm(FieldData(:,i));
end
error = mean(error_temp);
h1 = figure('Name',['Outputs_EVAP_ACC Transient Tuning Results1_Cooling Data ' ,filename]);
for i=1:1:3
subplot(2,2,i)
plot(FieldData(:,i),'r')
hold on
plot(y_var(:,i),'b--')
plot(y_var_nom(:,i),'k')
legend('data', 'tuned model','nominal',15);
legend('boxoff');
```

```
xlabel('Time (10s)','fontsize',15);
ylabel(Name_outputs{i},'FontSize',15)
set(gca,'FontSize',15);
end
hold off
saveas(h1,[directory{1},['Outputs_EVAP_ACC Transient Tuning Results1_Cooling Data ' ,filename]]);

h2 = figure('Name',['Outputs_EVAP_ACC Transient Tuning Results2_Cooling Data ' ,filename]);
for i=4:1:5
subplot(2,2,i-3)
plot(y_var(:,i),'b--')
hold on
plot(y_var_nom(:,i),'k')
legend('tuned model','nominal',15);
legend('boxoff');
xlabel('Time (10s)','fontsize',15);
ylabel(Name_outputs{i},'FontSize',15)
set(gca,'FontSize',15);
end
hold off
saveas(h2,[directory{1},['Outputs_EVAP_ACC Transient Tuning Results2_Cooling Data ' ,filename]]);

h3 = figure('Name',['Inputs_EVAP_ACC Transient Tuning Results2_Cooling Data ' ,filename]);
for i=1:1:4
subplot(2,2,i)
plot(Inputs(:,i),'g')
```

```
xlabel('Time (10s)','fontsize',15);
ylabel(Name_inputs{i},'FontSize',15)
set(gca,'FontSize',15);
end
hold off
saveas(h3,[directory{1},['Inputs_EVAP_ACC Transient Tuning Results2_Cooling Data ' ,filename]]);

figure
plot(FieldData(:,4),'r')
hold on
plot(y_var(:,6),'b--') % matlabpool close
xlabel('Time (10s)','fontsize',16);
ylabel('He_ri','FontSize',15)
legend('Data','tuned model',15);
hold off

tElapsed=toc(tStart);
save([directory{2},['Outputs_EVAP_ACC Transient Tuning Results_AllVariables_Cooling Data '
,filename]]);
matlabpool close
```

APPENDIX B

Input File of EnergyPlus

```
!-Generator IDFEditor 1.44
!-Option SortedOrder
!-NOTE: All comments with '!-' are ignored by the IDFEditor and are generated automatically.
!-      Use '!' comments if they need to be retained when using the IDFEditor.
!-   ===========  ALL OBJECTS IN CLASS: VERSION ===========
Version,
    8.0;                    !- Version Identifier
!-   ===========  ALL OBJECTS IN CLASS: SIMULATIONCONTROL ===========
SimulationControl,
    Yes,                    !- Do Zone Sizing Calculation
    No,                     !- Do System Sizing Calculation
    No,                     !- Do Plant Sizing Calculation
    No,                     !- Run Simulation for Sizing Periods
    Yes;                    !- Run Simulation for Weather File Run Periods
!-   ===========  ALL OBJECTS IN CLASS: BUILDING ===========
Building,
    ResidenceHouse,         !- Name
    0.0,                    !- North Axis {deg}
    City,                   !- Terrain
    0.04,                   !- Loads Convergence Tolerance Value
    0.4,                    !- Temperature Convergence Tolerance Value {deltaC}
    FullExterior,           !- Solar Distribution
    25,                     !- Maximum Number of Warmup Days
    6;                      !- Minimum Number of Warmup Days
!-   ===========  ALL OBJECTS IN CLASS: TIMESTEP ===========
Timestep,
    60;                     !- Number of Timesteps per Hour
!-   ===========  ALL OBJECTS IN CLASS: SITE:LOCATION ===========
Site:Location,
    DaytonIntlAP,           !- Name
```

```
   39.9,                !- Latitude {deg}
   -84.22,              !- Longitude {deg}
   -5.0,                !- Time Zone {hr}
   305;                 !- Elevation {m}
!-   ==========  ALL OBJECTS IN CLASS: SIZINGPERIOD:WEATHERFILEDAYS ==========
SizingPeriod:WeatherFileDays,
   ResidenceSizing,         !- Name
   1,                   !- Begin Month
   1,                   !- Begin Day of Month
   12,                  !- End Month
   31,                  !- End Day of Month
   Monday,                !- Day of Week for Start Day
   Yes,                 !- Use Weather File Daylight Saving Period
   Yes;                 !- Use Weather File Rain and Snow Indicators
!-   ==========  ALL OBJECTS IN CLASS: RUNPERIOD ==========
RunPeriod,
   ResidenceRunPeriod,      !- Name
   7,                   !- Begin Month
   24,                  !- Begin Day of Month
   8,                   !- End Month
   23,                  !- End Day of Month
   UseWeatherFile,          !- Day of Week for Start Day
   Yes,                 !- Use Weather File Holidays and Special Days
   Yes,                 !- Use Weather File Daylight Saving Period
   No,                  !- Apply Weekend Holiday Rule
   Yes,                 !- Use Weather File Rain Indicators
   Yes,                 !- Use Weather File Snow Indicators
   1,                   !- Number of Times Runperiod to be Repeated
   Yes;                 !- Increment Day of Week on repeat
!-   ==========  ALL OBJECTS IN CLASS: RUNPERIODCONTROL:SPECIALDAYS ==========
RunPeriodControl:SpecialDays,
   New Years Day,           !- Name
   January 1,             !- Start Date
   1,                   !- Duration {days}
   Holiday;               !- Special Day Type
```

```
RunPeriodControl:SpecialDays,
   Veterans Day,           !- Name
   November 11,            !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Christmas,              !- Name
   December 25,            !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Independence Day,       !- Name
   July 4,                 !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type

RunPeriodControl:SpecialDays,
   MLK Day,                !- Name
   3rd Monday in January,  !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Presidents Day,         !- Name
   3rd Monday in February, !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Memorial Day,           !- Name
   Last Monday in May,     !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Labor Day,              !- Name
   1st Monday in September, !- Start Date
   1,                      !- Duration {days}
```

```
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Columbus Day,           !- Name
   2nd Monday in October,  !- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
RunPeriodControl:SpecialDays,
   Thanksgiving,           !- Name
   4th Thursday in November,!- Start Date
   1,                      !- Duration {days}
   Holiday;                !- Special Day Type
!-   ====== ALL OBJECTS IN CLASS: RUNPERIODCONTROL:DAYLIGHTSAVINGTIME ======
! Daylight Saving Period in US
RunPeriodControl:DaylightSavingTime,
   2nd Sunday in March,    !- Start Date
   1st Sunday in November; !- End Date
!-   ===========  ALL OBJECTS IN CLASS: SCHEDULETYPELIMITS ===========
ScheduleTypeLimits,
   Any Number;             !- Name
ScheduleTypeLimits,
   Fraction,               !- Name
   0.0,                    !- Lower Limit Value
   1.0,                    !- Upper Limit Value
   CONTINUOUS;             !- Numeric Type
ScheduleTypeLimits,
   Temperature,            !- Name
   -60,                    !- Lower Limit Value
   200,                    !- Upper Limit Value
   CONTINUOUS;             !- Numeric Type
ScheduleTypeLimits,
   On/Off,                 !- Name
   0,                      !- Lower Limit Value
   1,                      !- Upper Limit Value
   DISCRETE;               !- Numeric Type
ScheduleTypeLimits,
```

```
   Control Type,           !- Name
   0,                      !- Lower Limit Value
   4,                      !- Upper Limit Value
   DISCRETE;               !- Numeric Type
ScheduleTypeLimits,
   Humidity,               !- Name
   10,                     !- Lower Limit Value
   90,                     !- Upper Limit Value
   CONTINUOUS;             !- Numeric Type
ScheduleTypeLimits,
   Number;                 !- Name
! -------------------------------------------------------------
! New objects created from ExpandObjects
! -------------------------------------------------------------
ScheduleTypeLimits,
   HVACTemplate Any Number; !- Name
!-   ===========  ALL OBJECTS IN CLASS: SCHEDULE:COMPACT ===========
Schedule:Compact,
   Office Lights Schedule,  !- Name
   Fraction,               !- Schedule Type Limits Name
   Through: 12/31,         !- Field 1
   For: Weekdays,          !- Field 2
   Until: 05:00,           !- Field 3
   0.05,                   !- Field 4
   Until: 07:00,           !- Field 5
   0.1,                    !- Field 6
   Until: 08:00,           !- Field 7
   0.3,                    !- Field 8
   Until: 17:00,           !- Field 9
   0.9,                    !- Field 10
   Until: 18:00,           !- Field 11
   0.5,                    !- Field 12
   Until: 20:00,           !- Field 13
   0.3,                    !- Field 14
   Until: 22:00,           !- Field 15
```

```
    0.2,                    !- Field 16
    Until: 23:00,           !- Field 17
    0.1,                    !- Field 18
    Until: 24:00,           !- Field 19
    0.05,                   !- Field 20
    For: SummerDesignDay,   !- Field 21
    Until: 24:00,           !- Field 22
    1.0,                    !- Field 23
    For: Saturday,          !- Field 24
    Until: 06:00,           !- Field 25
    0.05,                   !- Field 26
    Until: 08:00,           !- Field 27
    0.1,                    !- Field 28
    Until: 12:00,           !- Field 29
    0.3,                    !- Field 30
    Until: 17:00,           !- Field 31
    0.15,                   !- Field 32
    Until: 24:00,           !- Field 33
    0.05,                   !- Field 34
    For: WinterDesignDay,   !- Field 35
    Until: 24:00,           !- Field 36
    0.0,                    !- Field 37
    For: Sunday Holidays AllOtherDays,  !- Field 38
    Until: 24:00,           !- Field 39
    0.05;                   !- Field 40
Schedule:Compact,
    Office Equipment Schedule,  !- Name
    Fraction,               !- Schedule Type Limits Name
    Through: 12/31,         !- Field 1
    For: Weekdays,          !- Field 2
    Until: 08:00,           !- Field 3
    0.40,                   !- Field 4
    Until: 12:00,           !- Field 5
    0.90,                   !- Field 6
    Until: 13:00,           !- Field 7
```

```
    0.80,                   !- Field 8
    Until: 17:00,           !- Field 9
    0.90,                   !- Field 10
    Until: 18:00,           !- Field 11
    0.50,                   !- Field 12
    Until: 24:00,           !- Field 13
    0.40,                   !- Field 14
    For: SummerDesignDay,   !- Field 15
    Until: 24:00,           !- Field 16
    1.0,                    !- Field 17
    For: Saturday,          !- Field 18
    Until: 06:00,           !- Field 19
    0.30,                   !- Field 20
    Until: 08:00,           !- Field 21
    0.4,                    !- Field 22
    Until: 12:00,           !- Field 23
    0.5,                    !- Field 24
    Until: 17:00,           !- Field 25
    0.35,                   !- Field 26
    Until: 24:00,           !- Field 27
    0.30,                   !- Field 28
    For: WinterDesignDay,   !- Field 29
    Until: 24:00,           !- Field 30
    0.0,                    !- Field 31
    For: Sunday Holidays AllOtherDays,  !- Field 32
    Until: 24:00,           !- Field 33
    0.30;                   !- Field 34
Schedule:Compact,
    Office Occupancy Schedule,  !- Name
    Fraction,               !- Schedule Type Limits Name
    Through: 12/31,         !- Field 1
    For: Weekdays,          !- Field 2
    Until: 06:00,           !- Field 3
    0.0,                    !- Field 4
    Until: 07:00,           !- Field 5
```

0.1, !- Field 6
Until: 08:00, !- Field 7
0.2, !- Field 8
Until: 12:00, !- Field 9
0.95, !- Field 10
Until: 13:00, !- Field 11
0.5, !- Field 12
Until: 17:00, !- Field 13
0.95, !- Field 14
Until: 18:00, !- Field 15
0.3, !- Field 16
Until: 20:00, !- Field 17
0.1, !- Field 18
Until: 24:00, !- Field 19
0.05, !- Field 20
For: SummerDesignDay, !- Field 21
Until: 06:00, !- Field 22
0.0, !- Field 23
Until: 22:00, !- Field 24
1.0, !- Field 25
Until: 24:00, !- Field 26
0.05, !- Field 27
For: Saturday, !- Field 28
Until: 06:00, !- Field 29
0.0, !- Field 30
Until: 08:00, !- Field 31
0.1, !- Field 32
Until: 12:00, !- Field 33
0.3, !- Field 34
Until: 17:00, !- Field 35
0.1, !- Field 36
Until: 19:00, !- Field 37
0.0, !- Field 38
Until: 24:00, !- Field 39
0.0, !- Field 40

```
   For: WinterDesignDay,    !- Field 41
   Until: 24:00,            !- Field 42
   0.0,                     !- Field 43
   For: Sunday Holidays AllOtherDays,  !- Field 44
   Until: 06:00,            !- Field 45
   0.0,                     !- Field 46
   Until: 18:00,            !- Field 47
   0.0,                     !- Field 48
   Until: 24:00,            !- Field 49
   0.0;                     !- Field 50
Schedule:Compact,
   Infiltration Schedule,   !- Name
   Fraction,                !- Schedule Type Limits Name
   Through: 12/31,          !- Field 1
   For: Weekdays SummerDesignDay,  !- Field 2
   Until: 06:00,            !- Field 3
   1.0,                     !- Field 4
   Until: 22:00,            !- Field 5
   0.0,                     !- Field 6
   Until: 24:00,            !- Field 7
   1.0,                     !- Field 8
   For: Saturday WinterDesignDay,  !- Field 9
   Until: 06:00,            !- Field 10
   1.0,                     !- Field 11
   Until: 18:00,            !- Field 12
   0.0,                     !- Field 13
   Until: 24:00,            !- Field 14
   1.0,                     !- Field 15
   For: Sunday Holidays AllOtherDays,  !- Field 16
   Until: 24:00,            !- Field 17
   1.0;                     !- Field 18

Schedule:Compact,
   Infiltration Half On Schedule,  !- Name
   Fraction,                !- Schedule Type Limits Name
```

```
    Through: 12/31,          !- Field 1
    For: Weekdays SummerDesignDay,  !- Field 2
    Until: 06:00,            !- Field 3
    1.0,                     !- Field 4
    Until: 22:00,            !- Field 5
    0.5,                     !- Field 6
    Until: 24:00,            !- Field 7
    1.0,                     !- Field 8
    For: Saturday WinterDesignDay,  !- Field 9
    Until: 06:00,            !- Field 10
    1.0,                     !- Field 11
    Until: 18:00,            !- Field 12
    0.5,                     !- Field 13
    Until: 24:00,            !- Field 14
    1.0,                     !- Field 15
    For: Sunday Holidays AllOtherDays,  !- Field 16
    Until: 24:00,            !- Field 17
    1.0;                     !- Field 18
Schedule:Compact,
    Infiltration Quarter On Schedule,  !- Name
    Fraction,                !- Schedule Type Limits Name
    Through: 12/31,          !- Field 1
    For: Weekdays SummerDesignDay,  !- Field 2
    Until: 06:00,            !- Field 3
    1.0,                     !- Field 4
    Until: 22:00,            !- Field 5
    0.25,                    !- Field 6
    Until: 24:00,            !- Field 7
    1.0,                     !- Field 8
    For: Saturday WinterDesignDay,  !- Field 9
    Until: 06:00,            !- Field 10
    1.0,                     !- Field 11
    Until: 18:00,            !- Field 12
    0.25,                    !- Field 13
    Until: 24:00,            !- Field 14
```

```
    1.0,                    !- Field 15
    For: Sunday Holidays AllOtherDays,  !- Field 16
    Until: 24:00,           !- Field 17
    1.0;                    !- Field 18
Schedule:Compact,
    Hours of Operation Schedule,  !- Name
    On/Off,                 !- Schedule Type Limits Name
    Through: 12/31,         !- Field 1
    For: Weekdays SummerDesignDay,  !- Field 2
    Until: 06:00,           !- Field 3
    0.0,                    !- Field 4
    Until: 22:00,           !- Field 5
    1.0,                    !- Field 6
    Until: 24:00,           !- Field 7
    0.0,                    !- Field 8
    For: Saturday WinterDesignDay,  !- Field 9
    Until: 06:00,           !- Field 10
    0.0,                    !- Field 11
    Until: 18:00,           !- Field 12
    1.0,                    !- Field 13
    Until: 24:00,           !- Field 14
    0.0,                    !- Field 15
    For: Sunday Holidays AllOtherDays,  !- Field 16
    Until: 24:00,           !- Field 17
    0.0;                    !- Field 18
Schedule:Compact,
    Always On,              !- Name
    Fraction,               !- Schedule Type Limits Name
    Through: 12/31,         !- Field 1
    For: AllDays,           !- Field 2
    Until: 24:00,           !- Field 3
    1.0;                    !- Field 4
Schedule:Compact,
    Always Off,             !- Name
    Fraction,               !- Schedule Type Limits Name
```

```
   Through: 12/31,          !- Field 1
   For: AllDays,            !- Field 2
   Until: 24:00,            !- Field 3
   0.0;                     !- Field 4
Schedule:Compact,
   Heating Setpoint Schedule,  !- Name
   Temperature,             !- Schedule Type Limits Name
   Through: 12/31,          !- Field 1
   For: Weekdays,           !- Field 2
   Until: 05:00,            !- Field 3
   15.6,                    !- Field 4
   Until: 19:00,            !- Field 5
   21.0,                    !- Field 6
   Until: 24:00,            !- Field 7
   15.6,                    !- Field 8
   For SummerDesignDay,     !- Field 9
   Until: 24:00,            !- Field 10
   15.6,                    !- Field 11
   For: Saturday,           !- Field 12
   Until: 06:00,            !- Field 13
   15.6,                    !- Field 14
   Until: 17:00,            !- Field 15
   21.0,                    !- Field 16
   Until: 24:00,            !- Field 17
   15.6,                    !- Field 18
   For: WinterDesignDay,    !- Field 19
   Until: 24:00,            !- Field 20
   21.0,                    !- Field 21
   For: Sunday Holidays AllOtherDays,  !- Field 22
   Until: 24:00,            !- Field 23
   15.6;                    !- Field 24
Schedule:Compact,
   Cooling Setpoint Schedule,  !- Name
   Temperature,             !- Schedule Type Limits Name
   Through: 12/31,          !- Field 1
```

```
   For: Weekdays SummerDesignDay,  !- Field 2
   Until: 06:00,          !- Field 3
   30.0,                  !- Field 4
   Until: 22:00,          !- Field 5
   24.0,                  !- Field 6
   Until: 24:00,          !- Field 7
   30.0,                  !- Field 8
   For: Saturday,         !- Field 9
   Until: 06:00,          !- Field 10
   30.0,                  !- Field 11
   Until: 18:00,          !- Field 12
   24.0,                  !- Field 13
   Until: 24:00,          !- Field 14
   30.0,                  !- Field 15
   For WinterDesignDay,    !- Field 16
   Until: 24:00,          !- Field 17
   30.0,                  !- Field 18
   For: Sunday Holidays AllOtherDays,  !- Field 19
   Until: 24:00,          !- Field 20
   30.0;                  !- Field 21
Schedule:Compact,
   Office Activity Schedule,!- Name
   Any Number,            !- Schedule Type Limits Name
   Through: 12/31,        !- Field 1
   For: AllDays,          !- Field 2
   Until: 24:00,          !- Field 3
   120.;                  !- Field 4
Schedule:Compact,
   Office Work Eff. Schedule,  !- Name
   Fraction,              !- Schedule Type Limits Name
   Through: 12/31,        !- Field 1
   For: AllDays,          !- Field 2
   Until: 24:00,          !- Field 3
   0.0;                   !- Field 4
Schedule:Compact,
```

```
   Office Clothing Schedule,!- Name
   Any Number,              !- Schedule Type Limits Name
   Through: 04/30,          !- Field 1
   For: AllDays,            !- Field 2
   Until: 24:00,            !- Field 3
   1.0,                     !- Field 4
   Through: 09/30,          !- Field 5
   For: AllDays,            !- Field 6
   Until: 24:00,            !- Field 7
   0.5,                     !- Field 8
   Through: 12/31,          !- Field 9
   For: AllDays,            !- Field 10
   Until: 24:00,            !- Field 11
   1.0;                     !- Field 12
Schedule:Compact,
   ResidenceHeatingSCH,     !- Name
   Temperature,             !- Schedule Type Limits Name
   Through: 12/31,          !- Field 1
   For: AllDays,            !- Field 2
   Until: 24:00,            !- Field 3
   21;                      !- Field 4
Schedule:Compact,
   Residence Activity Schedule,  !- Name
   Any Number,              !- Schedule Type Limits Name
   Through: 12/31,          !- Field 1
   For: AllDays,            !- Field 2
   Until: 24:00,            !- Field 3
   2;                       !- Field 4
Schedule:Compact,
   ResidenceInfiltrationSCH,!- Name
   Any Number,              !- Schedule Type Limits Name
   Through: 12/31,          !- Field 1
   For: AllDays,            !- Field 2
   Until: 24:00,            !- Field 3
   1.66666666666667E-03;    !- Field 4
```

```
Schedule:Compact,
   Residence Activity Level Schedule,  !- Name
   Any Number,            !- Schedule Type Limits Name
   Through: 12/31,        !- Field 1
   For: AllDays,          !- Field 2
   Until: 24:00,          !- Field 3
   70;                    !- Field 4
Schedule:Compact,
   Residence HumidiStat SCH,!- Name
   Humidity,              !- Schedule Type Limits Name
   Through: 12/31,        !- Field 1
   For: AllDays,          !- Field 2
   Until: 24:00,          !- Field 3
   38;                    !- Field 4
Schedule:Compact,
   Residence HumidiRate SCH,!- Name
   Any Number,            !- Schedule Type Limits Name
   Through: 12/31,        !- Field 1
   For: AllDays,          !- Field 2
   Until: 24:00,          !- Field 3
   .00166667;             !- Field 4
Schedule:Compact,
   HVACTemplate-Always 4,   !- Name
   HVACTemplate Any Number, !- Schedule Type Limits Name
   Through: 12/31,        !- Field 1
   For: AllDays,          !- Field 2
   Until: 24:00,          !- Field 3
   4;                     !- Field 4
!-   ===========  ALL OBJECTS IN CLASS: MATERIAL ===========
Material,
   F08 Metal surface,       !- Name
   Smooth,                  !- Roughness
   0.0008,                  !- Thickness {m}
   45.28,                   !- Conductivity {W/m-K}
   7824,                    !- Density {kg/m3}
```

```
   500;                    !- Specific Heat {J/kg-K}
Material,
   I01 25mm insulation board,  !- Name
   MediumRough,            !- Roughness
   0.0254,                 !- Thickness {m}
   0.03,                   !- Conductivity {W/m-K}
   43,                     !- Density {kg/m3}
   1210;                   !- Specific Heat {J/kg-K}
Material,
   I02 50mm insulation board,  !- Name
   MediumRough,            !- Roughness
   0.0508,                 !- Thickness {m}
   0.03,                   !- Conductivity {W/m-K}
   43,                     !- Density {kg/m3}
   1210;                   !- Specific Heat {J/kg-K}
Material,
   G01a 19mm gypsum board,  !- Name
   MediumSmooth,           !- Roughness
   0.019,                  !- Thickness {m}
   0.16,                   !- Conductivity {W/m-K}
   800,                    !- Density {kg/m3}
   1090;                   !- Specific Heat {J/kg-K}
Material,
   M11 100mm lightweight concrete,  !- Name
   MediumRough,            !- Roughness
   0.1016,                 !- Thickness {m}
   0.53,                   !- Conductivity {W/m-K}
   1280,                   !- Density {kg/m3}
   840;                    !- Specific Heat {J/kg-K}
Material,
   F16 Acoustic tile,      !- Name
   MediumSmooth,           !- Roughness
   0.0191,                 !- Thickness {m}
   0.06,                   !- Conductivity {W/m-K}
   368,                    !- Density {kg/m3}
```

```
   590;                    !- Specific Heat {J/kg-K}
Material,
   M01 100mm brick,        !- Name
   MediumRough,            !- Roughness
   0.1016,                 !- Thickness {m}
   0.89,                   !- Conductivity {W/m-K}
   1920,                   !- Density {kg/m3}
   790;                    !- Specific Heat {J/kg-K}

Material,
   M15 200mm heavyweight concrete,  !- Name
   MediumRough,            !- Roughness
   0.2032,                 !- Thickness {m}
   1.95,                   !- Conductivity {W/m-K}
   2240,                   !- Density {kg/m3}
   900;                    !- Specific Heat {J/kg-K}
Material,
   M05 200mm concrete block,!- Name
   MediumRough,            !- Roughness
   0.2032,                 !- Thickness {m}
   1.11,                   !- Conductivity {W/m-K}
   800,                    !- Density {kg/m3}
   920;                    !- Specific Heat {J/kg-K}
Material,
   G05 25mm wood,          !- Name
   MediumSmooth,           !- Roughness
   0.0254,                 !- Thickness {m}
   0.15,                   !- Conductivity {W/m-K}
   608,                    !- Density {kg/m3}
   1630;                   !- Specific Heat {J/kg-K}
!-   ===========  ALL OBJECTS IN CLASS: MATERIAL:AIRGAP ===========
Material:AirGap,
   F04 Wall air space resistance,  !- Name
   0.15;                   !- Thermal Resistance {m2-K/W}
Material:AirGap,
```

```
   F05 Ceiling air space resistance,  !- Name
   0.18;                 !- Thermal Resistance {m2-K/W}
!-   ===========  ALL OBJECTS IN CLASS: WINDOWMATERIAL:GLAZING ===========
WindowMaterial:Glazing,
   Clear 3mm,            !- Name
   SpectralAverage,      !- Optical Data Type
   ,                     !- Window Glass Spectral Data Set Name
   0.003,                !- Thickness {m}
   0.837,                !- Solar Transmittance at Normal Incidence
   0.075,                !- Front Side Solar Reflectance at Normal Incidence
   0.075,                !- Back Side Solar Reflectance at Normal Incidence
   0.898,                !- Visible Transmittance at Normal Incidence
   0.081,                !- Front Side Visible Reflectance at Normal Incidence
   0.081,                !- Back Side Visible Reflectance at Normal Incidence
   0,                    !- Infrared Transmittance at Normal Incidence
   0.84,                 !- Front Side Infrared Hemispherical Emissivity
   0.84,                 !- Back Side Infrared Hemispherical Emissivity
   0.9;                  !- Conductivity {W/m-K}
!-   ===========  ALL OBJECTS IN CLASS: WINDOWMATERIAL:GAS ===========
WindowMaterial:Gas,
   Air 13mm,             !- Name
   Air,                  !- Gas Type
   0.0127;               !- Thickness {m}
!-   ===========  ALL OBJECTS IN CLASS: CONSTRUCTION ===========
Construction,
   Exterior Floor,       !- Name
   I02 50mm insulation board,  !- Outside Layer
   M15 200mm heavyweight concrete;  !- Layer 2
Construction,
   Interior Floor,       !- Name
   F16 Acoustic tile,    !- Outside Layer
   F05 Ceiling air space resistance,  !- Layer 2
   M11 100mm lightweight concrete;  !- Layer 3
Construction,
   Exterior Wall,        !- Name
```

```
    M01 100mm brick,         !- Outside Layer
    M15 200mm heavyweight concrete,  !- Layer 2
    I02 50mm insulation board,  !- Layer 3
    F04 Wall air space resistance,  !- Layer 4
    G01a 19mm gypsum board;  !- Layer 5
Construction,
    Interior Wall,           !- Name
    G01a 19mm gypsum board,  !- Outside Layer
    F04 Wall air space resistance,  !- Layer 2
    G01a 19mm gypsum board;  !- Layer 3
Construction,
    Exterior Roof,           !- Name
    M11 100mm lightweight concrete,  !- Outside Layer
    F05 Ceiling air space resistance,  !- Layer 2
    F16 Acoustic tile;       !- Layer 3
Construction,
    Interior Ceiling,        !- Name
    M11 100mm lightweight concrete,  !- Outside Layer
    F05 Ceiling air space resistance,  !- Layer 2
    F16 Acoustic tile;       !- Layer 3
Construction,
    Exterior Window,         !- Name
    Clear 3mm,               !- Outside Layer
    Air 13mm,                !- Layer 2
    Clear 3mm;               !- Layer 3
Construction,
    Interior Window,         !- Name
    Clear 3mm;               !- Outside Layer
Construction,
    Exterior Door,           !- Name
    F08 Metal surface,       !- Outside Layer
    I01 25mm insulation board;  !- Layer 2
Construction,
    Interior Door,           !- Name
    G05 25mm wood;           !- Outside Layer
```

```
!-   ===========  ALL OBJECTS IN CLASS: GLOBALGEOMETRYRULES ===========
GlobalGeometryRules,
    LowerLeftCorner,         !- Starting Vertex Position
    Counterclockwise,        !- Vertex Entry Direction
    Relative;                !- Coordinate System
!-   ===========  ALL OBJECTS IN CLASS: ZONE ===========
Zone,
    ResidenceHouseZone,      !- Name
    0.0,                     !- Direction of Relative North {deg}
    0.0,                     !- X Origin {m}
    0.0,                     !- Y Origin {m}
    0.0,                     !- Z Origin {m}
    1,                       !- Type
    1,                       !- Multiplier
    autocalculate,           !- Ceiling Height {m}
    autocalculate,           !- Volume {m3}
    autocalculate,           !- Floor Area {m2}
    TARP;                    !- Zone Inside Convection Algorithm
!-   ===========  ALL OBJECTS IN CLASS: BUILDINGSURFACE:DETAILED ===========
BuildingSurface:Detailed,
    Floor,                   !- Name
    Floor,                   !- Surface Type
    Exterior Floor,          !- Construction Name
    ResidenceHouseZone,      !- Zone Name
    Adiabatic,               !- Outside Boundary Condition
    ,                        !- Outside Boundary Condition Object
    NoSun,                   !- Sun Exposure
    NoWind,                  !- Wind Exposure
    0.0,                     !- View Factor to Ground
    4,                       !- Number of Vertices
    10.000000000000,         !- Vertex 1 X-coordinate {m}
    0.000000000000,          !- Vertex 1 Y-coordinate {m}
    0.000000000000,          !- Vertex 1 Z-coordinate {m}
    0.000000000000,          !- Vertex 2 X-coordinate {m}
    0.000000000000,          !- Vertex 2 Y-coordinate {m}
```

```
    0.000000000000,          !- Vertex 2 Z-coordinate {m}
    0.000000000000,          !- Vertex 3 X-coordinate {m}
    7.002288955332,          !- Vertex 3 Y-coordinate {m}
    0.000000000000,          !- Vertex 3 Z-coordinate {m}
    10.000000000000,         !- Vertex 4 X-coordinate {m}
    7.002288955332,          !- Vertex 4 Y-coordinate {m}
    0.000000000000;          !- Vertex 4 Z-coordinate {m}
BuildingSurface:Detailed,
    Back-Wall,               !- Name
    Wall,                    !- Surface Type
    Exterior Wall,           !- Construction Name
    ResidenceHouseZone,      !- Zone Name
    Outdoors,                !- Outside Boundary Condition
    ,                        !- Outside Boundary Condition Object
    SunExposed,              !- Sun Exposure
    WindExposed,             !- Wind Exposure
    ,                        !- View Factor to Ground
    4,                       !- Number of Vertices
    10.000000000000,         !- Vertex 1 X-coordinate {m}
    7.002288955332,          !- Vertex 1 Y-coordinate {m}
    0.000000000000,          !- Vertex 1 Z-coordinate {m}
    0.000000000000,          !- Vertex 2 X-coordinate {m}
    7.002288955332,          !- Vertex 2 Y-coordinate {m}
    0.000000000000,          !- Vertex 2 Z-coordinate {m}
    0.000000000000,          !- Vertex 3 X-coordinate {m}
    7.002288955332,          !- Vertex 3 Y-coordinate {m}
    3.000000000000,          !- Vertex 3 Z-coordinate {m}
    10.000000000000,         !- Vertex 4 X-coordinate {m}
    7.002288955332,          !- Vertex 4 Y-coordinate {m}
    3.000000000000;          !- Vertex 4 Z-coordinate {m}

BuildingSurface:Detailed,
    Left-Wall,               !- Name
    Wall,                    !- Surface Type
    Exterior Wall,           !- Construction Name
```

```
  ResidenceHouseZone,      !- Zone Name
  Outdoors,                !- Outside Boundary Condition
  ,                        !- Outside Boundary Condition Object
  SunExposed,              !- Sun Exposure
  WindExposed,             !- Wind Exposure
  ,                        !- View Factor to Ground
  4,                       !- Number of Vertices
  0.000000000000,          !- Vertex 1 X-coordinate {m}
  7.002288955332,          !- Vertex 1 Y-coordinate {m}
  0.000000000000,          !- Vertex 1 Z-coordinate {m}
  0.000000000000,          !- Vertex 2 X-coordinate {m}
  0.000000000000,          !- Vertex 2 Y-coordinate {m}
  0.000000000000,          !- Vertex 2 Z-coordinate {m}
  0.000000000000,          !- Vertex 3 X-coordinate {m}
  0.000000000000,          !- Vertex 3 Y-coordinate {m}
  3.000000000000,          !- Vertex 3 Z-coordinate {m}
  0.000000000000,          !- Vertex 4 X-coordinate {m}
  7.002288955332,          !- Vertex 4 Y-coordinate {m}
  3.000000000000;          !- Vertex 4 Z-coordinate {m}
BuildingSurface:Detailed,
  Right-Wall,              !- Name
  Wall,                    !- Surface Type
  Exterior Wall,           !- Construction Name
  ResidenceHouseZone,      !- Zone Name
  Outdoors,                !- Outside Boundary Condition
  ,                        !- Outside Boundary Condition Object
  SunExposed,              !- Sun Exposure
  WindExposed,             !- Wind Exposure
  ,                        !- View Factor to Ground
  4,                       !- Number of Vertices
  10.000000000000,         !- Vertex 1 X-coordinate {m}
  0.000000000000,          !- Vertex 1 Y-coordinate {m}
  0.000000000000,          !- Vertex 1 Z-coordinate {m}
  10.000000000000,         !- Vertex 2 X-coordinate {m}
  7.002288955332,          !- Vertex 2 Y-coordinate {m}
```

```
    0.000000000000,          !- Vertex 2 Z-coordinate {m}
    10.000000000000,         !- Vertex 3 X-coordinate {m}
    7.002288955332,          !- Vertex 3 Y-coordinate {m}
    3.000000000000,          !- Vertex 3 Z-coordinate {m}
    10.000000000000,         !- Vertex 4 X-coordinate {m}
    0.000000000000,          !- Vertex 4 Y-coordinate {m}
    3.000000000000;          !- Vertex 4 Z-coordinate {m}
BuildingSurface:Detailed,
    Roof,                    !- Name
    Roof,                    !- Surface Type
    Exterior Roof,           !- Construction Name
    ResidenceHouseZone,      !- Zone Name
    Outdoors,                !- Outside Boundary Condition
    ,                        !- Outside Boundary Condition Object
    SunExposed,              !- Sun Exposure
    WindExposed,             !- Wind Exposure
    ,                        !- View Factor to Ground
    4,                       !- Number of Vertices
    0.000000000000,          !- Vertex 1 X-coordinate {m}
    0.000000000000,          !- Vertex 1 Y-coordinate {m}
    3.000000000000,          !- Vertex 1 Z-coordinate {m}
    10.000000000000,         !- Vertex 2 X-coordinate {m}
    0.000000000000,          !- Vertex 2 Y-coordinate {m}
    3.000000000000,          !- Vertex 2 Z-coordinate {m}
    10.000000000000,         !- Vertex 3 X-coordinate {m}
    7.002288955332,          !- Vertex 3 Y-coordinate {m}
    3.000000000000,          !- Vertex 3 Z-coordinate {m}
    0.000000000000,          !- Vertex 4 X-coordinate {m}
    7.002288955332,          !- Vertex 4 Y-coordinate {m}
    3.000000000000;          !- Vertex 4 Z-coordinate {m}
BuildingSurface:Detailed,
    Front-Wall,              !- Name
    Wall,                    !- Surface Type
    Exterior Wall,           !- Construction Name
    ResidenceHouseZone,      !- Zone Name
```

```
   Outdoors,               !- Outside Boundary Condition
   ,                       !- Outside Boundary Condition Object
   SunExposed,             !- Sun Exposure
   WindExposed,            !- Wind Exposure
   ,                       !- View Factor to Ground
   4,                      !- Number of Vertices
   0.000000000000,         !- Vertex 1 X-coordinate {m}
   0.000000000000,         !- Vertex 1 Y-coordinate {m}
   0.000000000000,         !- Vertex 1 Z-coordinate {m}
   10.000000000000,        !- Vertex 2 X-coordinate {m}
   0.000000000000,         !- Vertex 2 Y-coordinate {m}
   0.000000000000,         !- Vertex 2 Z-coordinate {m}
   10.000000000000,        !- Vertex 3 X-coordinate {m}
   0.000000000000,         !- Vertex 3 Y-coordinate {m}
   3.000000000000,         !- Vertex 3 Z-coordinate {m}
   0.000000000000,         !- Vertex 4 X-coordinate {m}
   0.000000000000,         !- Vertex 4 Y-coordinate {m}
   3.000000000000;         !- Vertex 4 Z-coordinate {m}
!-  ==========  ALL OBJECTS IN CLASS: FENESTRATIONSURFACE:DETAILED ===========
FenestrationSurface:Detailed,
   Door,                   !- Name
   Door,                   !- Surface Type
   Exterior Door,          !- Construction Name
   Front-Wall,             !- Building Surface Name
   ,                       !- Outside Boundary Condition Object
   ,                       !- View Factor to Ground
   ,                       !- Shading Control Name
   ,                       !- Frame and Divider Name
   ,                       !- Multiplier
   4,                      !- Number of Vertices
   3.861256172497,         !- Vertex 1 X-coordinate {m}
   0.000000000000,         !- Vertex 1 Y-coordinate {m}
   0.000000000000,         !- Vertex 1 Z-coordinate {m}
   6.861256172497,         !- Vertex 2 X-coordinate {m}
   0.000000000000,         !- Vertex 2 Y-coordinate {m}
```

```
    0.000000000000,         !- Vertex 2 Z-coordinate {m}
    6.861256172497,         !- Vertex 3 X-coordinate {m}
    0.000000000000,         !- Vertex 3 Y-coordinate {m}
    2.000000000000,         !- Vertex 3 Z-coordinate {m}
    3.861256172497,         !- Vertex 4 X-coordinate {m}
    0.000000000000,         !- Vertex 4 Y-coordinate {m}
    2.000000000000;         !- Vertex 4 Z-coordinate {m}
FenestrationSurface:Detailed,
    Window,                 !- Name
    Window,                 !- Surface Type
    Exterior Window,        !- Construction Name
    Right-Wall,             !- Building Surface Name
    ,                       !- Outside Boundary Condition Object
    ,                       !- View Factor to Ground
    ,                       !- Shading Control Name
    ,                       !- Frame and Divider Name
    ,                       !- Multiplier
    4,                      !- Number of Vertices
    10.000000000000,        !- Vertex 1 X-coordinate {m}
    2.642768753792,         !- Vertex 1 Y-coordinate {m}
    0.652889645898,         !- Vertex 1 Z-coordinate {m}
    10.000000000000,        !- Vertex 2 X-coordinate {m}
    4.642768753792,         !- Vertex 2 Y-coordinate {m}
    0.652889645898,         !- Vertex 2 Z-coordinate {m}
    10.000000000000,        !- Vertex 3 X-coordinate {m}
    4.642768753792,         !- Vertex 3 Y-coordinate {m}
    2.652889645898,         !- Vertex 3 Z-coordinate {m}
    10.000000000000,        !- Vertex 4 X-coordinate {m}
    2.642768753792,         !- Vertex 4 Y-coordinate {m}
    2.652889645898;         !- Vertex 4 Z-coordinate {m}
!-  ========  ALL OBJECTS IN CLASS: ZONEINFILTRATION:DESIGNFLOWRATE =========
ZoneInfiltration:DesignFlowRate,
    ResidenceInfiltration,   !- Name
    ResidenceHouseZone,      !- Zone or ZoneList Name
    ResidenceInfiltrationSCH,!- Schedule Name
```

```
  Flow/Zone,              !- Design Flow Rate Calculation Method
  0.16667,                !- Design Flow Rate {m3/s}
  ,                       !- Flow per Zone Floor Area {m3/s-m2}
  ,                       !- Flow per Exterior Surface Area {m3/s-m2}
  ,                       !- Air Changes per Hour {1/hr}
  1,                      !- Constant Term Coefficient
  ,                       !- Temperature Term Coefficient
  ,                       !- Velocity Term Coefficient
  ;                       !- Velocity Squared Term Coefficient
!-  ============  ALL OBJECTS IN CLASS: SIZING:ZONE ============
Sizing:Zone,
  ResidenceHouseZone,     !- Zone or ZoneList Name
  SupplyAirTemperature,   !- Zone Cooling Design Supply Air Temperature Input Method
  10,                     !- Zone Cooling Design Supply Air Temperature {C}
  ,                       !- Zone Cooling Design Supply Air Temperature Difference {deltaC}
  SupplyAirTemperature,   !- Zone Heating Design Supply Air Temperature Input Method
  30,                     !- Zone Heating Design Supply Air Temperature {C}
  ,                       !- Zone Heating Design Supply Air Temperature Difference {deltaC}
  0.004,                  !- Zone Cooling Design Supply Air Humidity Ratio {kgWater/kgDryAir}
  0.004,                  !- Zone Heating Design Supply Air Humidity Ratio {kgWater/kgDryAir}
  ,                       !- Design Specification Outdoor Air Object Name
  ,                       !- Zone Heating Sizing Factor
  ,                       !- Zone Cooling Sizing Factor
  Flow/Zone,              !- Cooling Design Air Flow Method
  0.416666666666667,      !- Cooling Design Air Flow Rate {m3/s}
  0.000762,               !- Cooling Minimum Air Flow per Zone Floor Area {m3/s-m2}
  ,                       !- Cooling Minimum Air Flow {m3/s}
  ,                       !- Cooling Minimum Air Flow Fraction
  Flow/Zone,              !- Heating Design Air Flow Method
  0.416666666666667,      !- Heating Design Air Flow Rate {m3/s}
  0.002032,               !- Heating Maximum Air Flow per Zone Floor Area {m3/s-m2}
  0.1415762,              !- Heating Maximum Air Flow {m3/s}
  0.3;                    !- Heating Maximum Air Flow Fraction
```

```
!-   ===========  ALL OBJECTS IN CLASS: ZONECONTROL:HUMIDISTAT ===========
! HVACTemplate:Thermostat,
!    Residence Constant Setpoint Thermostat,  !- Name
!    ResidenceHeatingSCH,     !- Heating Setpoint Schedule Name
!    ,                        !- Constant Heating Setpoint {C}
!    ResidenceHeatingSCH,     !- Cooling Setpoint Schedule Name
!    ;                        !- Constant Cooling Setpoint {C}
! HVACTemplate:Zone:IdealLoadsAirSystem,
!    ResidenceHouseZone,      !- Zone Name
!    Residence Constant Setpoint Thermostat;  !- Template Thermostat Name
ZoneControl:Humidistat,
   Residence HumidiStat,    !- Name
   ResidenceHouseZone,      !- Zone Name
   Residence HumidiStat SCH;!- Humidifying Relative Humidity Setpoint Schedule Name
!-   ===========  ALL OBJECTS IN CLASS: ZONECONTROL:THERMOSTAT ===========
ZoneControl:Thermostat,
   ResidenceHouseZone Thermostat,  !- Name
   ResidenceHouseZone,      !- Zone or ZoneList Name
   HVACTemplate-Always 4,   !- Control Type Schedule Name
   ThermostatSetpoint:DualSetpoint,  !- Control 1 Object Type
   Residence Constant Setpoint Thermostat Dual SP Control;  !- Control 1 Name
!-   ========  ALL OBJECTS IN CLASS: THERMOSTATSETPOINT:DUALSETPOINT =========
ThermostatSetpoint:DualSetpoint,
   Residence Constant Setpoint Thermostat Dual SP Control,  !- Name
   ResidenceHeatingSCH,     !- Heating Setpoint Temperature Schedule Name
   ResidenceHeatingSCH;     !- Cooling Setpoint Temperature Schedule Name
!-   ==========  ALL OBJECTS IN CLASS: ZONEHVAC:IDEALLOADSAIRSYSTEM ==========
ZoneHVAC:IdealLoadsAirSystem,
   ResidenceHouseZoneZoneHVAC:IdealLoadsAirSystem,  !- Name
   ,                        !- Availability Schedule Name
   ResidenceHouseZone Supply Inlet,  !- Zone Supply Air Node Name
   ,                        !- Zone Exhaust Air Node Name
   50,                      !- Maximum Heating Supply Air Temperature {C}
   13,                      !- Minimum Cooling Supply Air Temperature {C}
   0.0156,                  !- Maximum Heating Supply Air Humidity Ratio {kgWater/kgDryAir}
```

```
   0.0077,                  !- Minimum Cooling Supply Air Humidity Ratio {kgWater/kgDryAir}
   NoLimit,                 !- Heating Limit
   ,                        !- Maximum Heating Air Flow Rate {m3/s}
   ,                        !- Maximum Sensible Heating Capacity {W}
   NoLimit,                 !- Cooling Limit
   ,                        !- Maximum Cooling Air Flow Rate {m3/s}
   ,                        !- Maximum Total Cooling Capacity {W}
   ,                        !- Heating Availability Schedule Name
   ,                        !- Cooling Availability Schedule Name
   Humidistat,              !- Dehumidification Control Type
   0.7,                     !- Cooling Sensible Heat Ratio {dimensionless}
   Humidistat,              !- Humidification Control Type
   ,                        !- Design Specification Outdoor Air Object Name
   ,                        !- Outdoor Air Inlet Node Name
   None,                    !- Demand Controlled Ventilation Type
   NoEconomizer,            !- Outdoor Air Economizer Type
   None,                    !- Heat Recovery Type
   0.7,                     !- Sensible Heat Recovery Effectiveness {dimensionless}
   0.65;                    !- Latent Heat Recovery Effectiveness {dimensionless}
!-  ============  ALL OBJECTS IN CLASS: ZONEHVAC:EQUIPMENTLIST ============
ZoneHVAC:EquipmentList,
   ResidenceHouseZone Equipment,  !- Name
   ZoneHVAC:IdealLoadsAirSystem,  !- Zone Equipment 1 Object Type
   ResidenceHouseZoneZoneHVAC:IdealLoadsAirSystem,  !- Zone Equipment 1 Name
   1,                       !- Zone Equipment 1 Cooling Sequence
   1;                       !- Zone Equipment 1 Heating or No-Load Sequence


!-  =========  ALL OBJECTS IN CLASS: ZONEHVAC:EQUIPMENTCONNECTIONS =========

ZoneHVAC:EquipmentConnections,
   ResidenceHouseZone,      !- Zone Name
   ResidenceHouseZone Equipment,  !- Zone Conditioning Equipment List Name
   ResidenceHouseZone Supply Inlet,  !- Zone Air Inlet Node or NodeList Name
   ,                        !- Zone Air Exhaust Node or NodeList Name
```

```
   ResidenceHouseZone Zone Air Node,  !- Zone Air Node Name
   ResidenceHouseZone Return Outlet;  !- Zone Return Air Node Name
!-   ===========  ALL OBJECTS IN CLASS: NODELIST ===========
NodeList,
   ResidenceHouseZone Supply Inlet,  !- Name
   ResidenceHouseZone Supply Inlet;  !- Node 1 Name
!-   ===========  ALL OBJECTS IN CLASS: OUTPUT:VARIABLEDICTIONARY ===========
Output:VariableDictionary,
   IDF;                  !- Key Field
!-   ==========  ALL OBJECTS IN CLASS: OUTPUT:TABLE:SUMMARYREPORTS ==========
Output:Table:SummaryReports,
   AllMonthly,           !- Report 1 Name
   AllSummary,           !- Report 2 Name
   ClimaticDataSummary;     !- Report 3 Name
!-   ===========  ALL OBJECTS IN CLASS: OUTPUT:TABLE:MONTHLY ===========
Output:Table:Monthly,
   ResidenceMonthly,       !- Name
   2,                    !- Digits After Decimal
   Zone Outdoor Air Drybulb Temperature,  !- Variable or Meter 1 Name
   SumOrAverageDuringHoursShown,  !- Aggregation Type for Variable or Meter 1
   Zone Air Relative Humidity,  !- Variable or Meter 2 Name
   SumOrAverageDuringHoursShown,  !- Aggregation Type for Variable or Meter 2
   Zone Air Temperature,    !- Variable or Meter 3 Name
   SumOrAverageDuringHoursShown,  !- Aggregation Type for Variable or Meter 3
   Surface Inside Face Temperature,  !- Variable or Meter 4 Name
   SumOrAverageDuringHoursShown,  !- Aggregation Type for Variable or Meter 4
   Surface Outside Face Outdoor Air Drybulb Temperature,  !- Variable or Meter 5 Name
   SumOrAverageDuringHoursShown,  !- Aggregation Type for Variable or Meter 5
   Surface Outside Face Temperature,  !- Variable or Meter 6 Name
   SumOrAverageDuringHoursShown;  !- Aggregation Type for Variable or Meter 6
!-   ===========  ALL OBJECTS IN CLASS: OUTPUTCONTROL:TABLE:STYLE ===========
OutputControl:Table:Style,
   CommaAndXML;           !- Column Separator
!-   ===========  ALL OBJECTS IN CLASS: OUTPUT:VARIABLE ===========
Output:Variable,
```

```
  *,                    !- Key Value
  Surface Inside Face Temperature,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Surface Outside Face Temperature,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Zone Outdoor Air Drybulb Temperature,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Site Outdoor Air Relative Humidity,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Site Horizontal Infrared Radiation Rate per Area,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Site Diffuse Solar Radiation Rate per Area,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Site Direct Solar Radiation Rate per Area,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Site Solar Altitude Angle,  !- Variable Name
  Timestep;             !- Reporting Frequency
Output:Variable,
  *,                    !- Key Value
  Zone Air Relative Humidity,  !- Variable Name
  Timestep;             !- Reporting Frequency
```

```
Output:Variable,
   *,                    !- Key Value
   Zone Air Temperature,    !- Variable Name
   Timestep;             !- Reporting Frequency

Output:Variable,
   *,                    !- Key Value
   Zone Ideal Loads Zone Sensible Heating Rate,  !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
   Zone Ideal Loads Zone Latent Heating Rate,  !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
   Zone Ideal Loads Zone Total Heating Rate,  !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
   Zone Ideal Loads Zone Sensible Cooling Rate,  !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
   Zone Ideal Loads Zone Latent Cooling Rate,  !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
   Zone Ideal Loads Zone Total Cooling Rate,  !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
   System Node Temperature, !- Variable Name
   Timestep;             !- Reporting Frequency
Output:Variable,
   *,                    !- Key Value
```

```
  System Node Mass Flow Rate,  !- Variable Name
  Timestep;                !- Reporting Frequency
Output:Variable,
  *,                       !- Key Value
  System Node Relative Humidity,  !- Variable Name
  Timestep;                !- Reporting Frequency
Output:Variable,
  *,                       !- Key Value
  Zone Predicted Moisture Load Moisture Transfer Rate,  !- Variable Name
  Timestep;                !- Reporting Frequency
```