MODELING AND CONTROL OF THERMAL MANAGEMENT SYSTEMS ON ELECTRIC VEHICLES

BY

KAYLA MICHELLE RUSSELL

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Adviser:

Professor Andrew Alleyne

Abstract

A growing emphasis on electrifying the transportation industry has led to an increase in the development and implementation of electric vehicles. The thermal management system on electric vehicles is one of the most critical subsystems that ensures safe and comfortable operation of these vehicles. Vapor compression systems are one of the most commonly used cooling methods in thermal management systems for electric vehicles. This thesis develops a vapor compression system model using graph-based modeling techniques. The graph-based modeling framework is extended in this work to capture coupled multi-timescale dynamics that exist in dynamic systems, such as the vapor compression system, while preserving the computational efficiency and modularity of the existing graph-modeling framework. This extended graph-based modeling framework is shown to accurately capture system dynamics within 1% of the current state of the art modeling approaches. Then the graph-based vapor compression system model is connected to a thermal cabin model to cool the interior cabin of an electric vehicle. This work also develops a model predictive control and a baseline control to manage the use of the vapor compression system to cool the cabin. The model predictive control and baseline control performed similarly in meeting the system constraints and objectives.

To Mom, Dad, and Brandon

Acknowledgments

First, I would like to thank my advisor, Dr. Andrew Alleyne, for his mentorship. I have greatly benefited from his valuable insight, guidance and advice. I look forward to working with him these next few years.

I would also like to thank past and present Alleyne Research Group (ARG) students. To the more senior members of ARG, Chris A., Cary, Spencer, and Mindy, thank you for fostering a welcoming environment and being willing to answer all of my questions. To the members of my cohort, Reid, Phil, Frank, Chris U., Dylan, and Kurt, thanks for being great coworkers that made the transition to graduate school more easy and enjoyable.

Additionally, I would like to acknowledge the financial support I received through the University of Illinois at Urbana-Champaign MechSE Distinguished Fellowship program. This support has allowed me to pursue exciting and meaningful research.

To my family, thank you for constantly supporting me and encouraging me throughout my academic journey. Your love and support have enabled me to pursue my goals and push limits that I did not think I could.

Lastly, I want to thank the friends I have made in graduate school. My experience would not be the same without them. Thank you especially to Kellie Halloran and Allisa Hastie for the friendship, support, and encouragement.

Table of contents

List of Abbreviations

BEV	Battery Electric Vehicle
CV	Control Volume
EEV	Electronic Expansion Valve
EV	Electric Vehicle
e-NTU	Effectiveness Number of Transfer Units
FLOP	Floating Points Operations
FV	Finite Volume
HTC	Heat Transfer Coefficient
HVAC	Heating, Ventilation, and Air Conditioning
ICE	Internal Combustion Engine
MB	Moving Boundary
MPC	Model Predictive Control
PHEV	Plug-in Hybrid Electric Vehicle
PID	Proportional-Integral-Derivative
SISO	Single-Input Single-Output
SMB	Switched Moving Boundary
TEV	Thermal Expansion Valve
TMS	Thermal Management System
VCS	Vapor Compression System

List of Symbols

Common Variables

A	Area
C	Capacitance
c_p	Specific Heat
D	Source flow matrix, Diameter
d	Disturbance
e	Graph edge
$\mathcal{E}$	Set of graph vertices
f	Friction Factor
h	Enthalpy
I	Identity Matrix
J	Objective Function
K	Kalman Gain
K_L	Minor Loss Coefficient
L	Length
$\dot{m}$	Mass flow rate
M	Incidence Matrix
$\dot{m}$	Mass Flow Rate
N_H	Horizon Length
P	Power Flow, Pressure, Covariance Matrix

Q	Heat Transfer
r	Reference
S	Adaptive Matrix
T	Temperature
t	time
u	Input
x	State variable
V	Volume
v	Graph vertex
$\mathcal{V}$	Set of graph vertices
α	Heat Transfer Coefficient
$\bar{\gamma}$	Mean Void Fraction
η	Efficiency
ζ	Normalized Lengths
Λ	Gain
ρ	Density
ω	Compressor Speed

Subscripts and Superscripts

a	Air
c	Condenser
cs	Cross Section
d	Dynamic variable
e	Evaporator
f	Fluid
head	Head state

k	Compressor, Time Step
r	Refrigerant
s	Graph source
SC	Subcool
SH	Superheat
t	Graph sink
tail	Tail state
TP	Two-Phase
v	Valve, Vapor
w	Wall

Chapter 1

Introduction

With a desire to reduce greenhouse gas emissions, there has been an emphasis on developing and implementing electric vehicles (EVs) around the world. While EVs have grown in popularity in the past decade, there are more improvements that must be made in order to ensure safe vehicle operation and to improve the performance of EVs. One of the main drawbacks of EVs is the relatively low driving range compared to internal combustion engine (ICE) vehicles. In order to improve the driving range of EVs, the auxiliary subsystems on an electric vehicle must be improved. As the thermal management system (TMS) is one of the subsystems that consumes the most energy, the TMS will be studied in this work. For this investigation, the vapor compression system (VCS) is at the core of the TMS onboard an EV. VCSs have been the main mode of cooling both conventional vehicles and EVs. However, VCSs are challenging to accurately model, as they contain complex nonlinear dynamics. This thesis aims to explore and improve the TMS through the use of different control strategies. A model predictive control (MPC) will be applied to the VCS on an EV and its performance will be compared with a baseline controller. In order to do effectively do this, a novel extension of the graph-based modeling framework is developed and utilized to mathematically model the VCS.

1.1 Motivation of Thesis

In the past decade there has been a significant increase in electrification of vehicles in the automotive industry. In 2010 there were less than 1 million battery electric vehicles (BEVs) and plug-in hybrid electric vehicles (PHEVs) on the roads. By the end of 2020 there were over 10 millions electric vehicles (EVs) on the roads worldwide. Figure 1.1 shows this trend over the past 10 years.

There are several reasons for the growing popularity of EVs. First, EV operation is

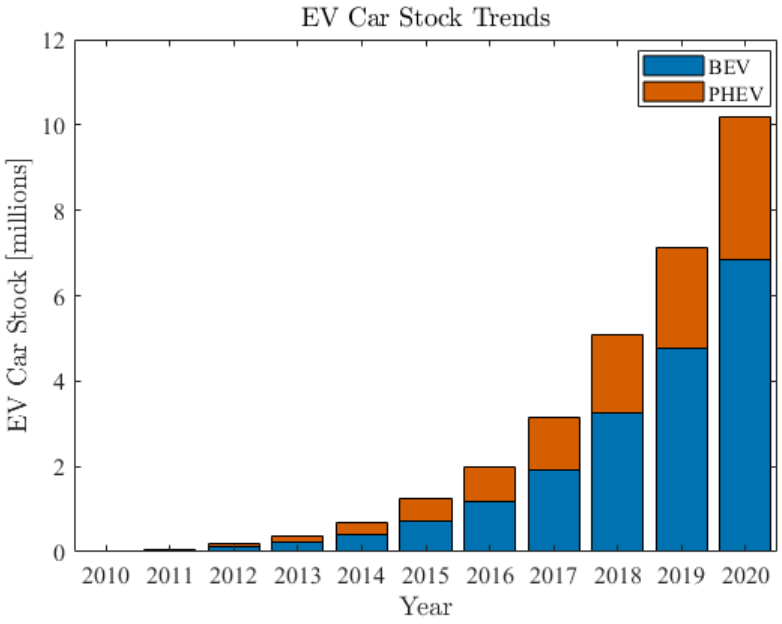


Figure 1.1: Electric vehicle trends. Based on data from the IEA Global EV Outlook 2021,https://www.iea.org/reports/global-ev-outlook-2021/trends-and-developments-in-electric-vehicle-markets. All rights reserved; as modified by Kayla Russell.

generally better than internal combustion engine (ICE) vehicle operation for the environment. The transportation sector accounted for 27% of the U.S. greenhouse gas emissions in 2020, with over half of those emissions generated from passenger cars, medium-, heavy-, and light-duty trucks [1]. Vehicles have become the main source of air pollution, especially in densely populated cities. For instance, it was found that vehicle emissions contributed to approximately 50% of the fine particulate matter air pollution in Shenzhen, China and Beijing, China [2]. Depending on the source of the electricity used to fuel the EV, EV operation can significantly reduce the amount of air pollution created by its operation. Additionally, EVs have better overall energy efficiency than conventional ICE vehicles. The total energy efficiency of a BEV is 60-70% while the total energy efficiency of ICE vehicle is 15-18% [2]. In addition to the environmental benefits, EVs also reduce the amount of noise pollution, especially in more urban areas, as the power train in a BEV is noticeably more quiet than an ICE.

While there are many benefits to EVs, there are a few main drawbacks that hinder their popularity. One drawback is the lack of existing infrastructure designed to charge EVs. While specific regions of the world are well-equipped for EVs, other areas lack the infrastructure to effectively implement EVs. Another drawback of EV use is the relatively limited driving range. Figure 1.2 shows the driving range of BEVs in the past 5 years. While the driving range of BEVs increased by 60% from 2015 to 2020, the average ICE vehicle driving range

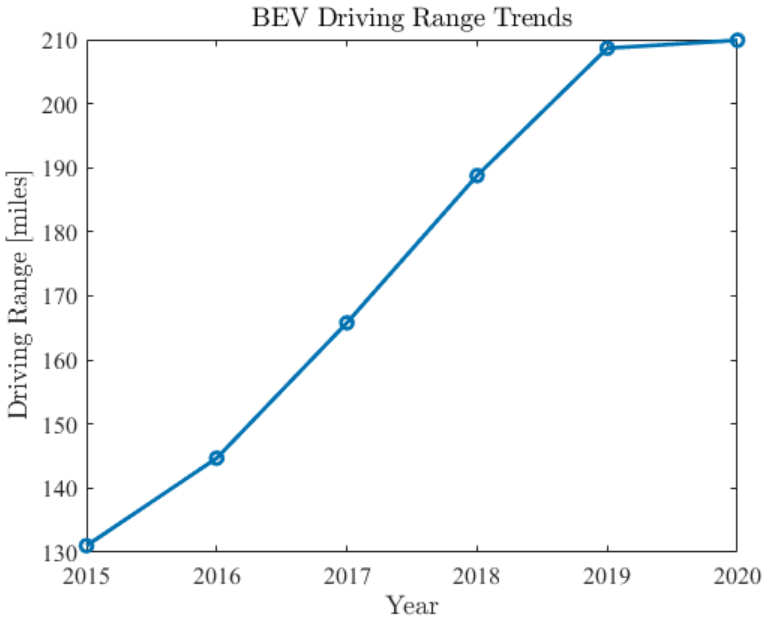


Figure 1.2: BEV driving range trends. Based on data from the IEA Global EV Outlook 2021,https://www.iea.org/reports/global-ev-outlook-2021/trends-and-developments-in-electric-vehicle-markets. All rights reserved; as modified by Kayla Russell.

was almost twice as long as that of a BEV in 2020.

When analyzing the fuel usage breakdown in a vehicle, the HVAC system consumes the most energy outside of the drive train [3]. The peak power demand of the TMS in an EV can increase by up to 5 kW when trying to maintain a comfortable cabin temperature and regulate the battery temperature to be within safe operating conditions. Generally, turning on the air conditioner can drop the fuel economy of a vehicle by about 18% [4]. This drop is even more pronounced in more advanced cars, with a drop of 25-40% [4]-[5]. Figure 1.3 shows how the Mitsubishi driving range changes based on operating conditions, where the orange "Max on" bars indicate that the vehicle is operating the air conditioning or heating system at its maximum capacity.

An improved TMS on EVs is needed to ensure safe operation of the vehicle while also improving the driving range of the EV. There are two main methods studied and implemented to improve the fuel efficiency and thermal management of electric vehicles: component optimization and controls strategies. Several different component optimization strategies have been outlined in [7]. There has been the most emphasis on optimizing the compressor as the compressor consumes about 65% of the total energy of a VCS on a vehicle [8]. One significant compressor improvement has been the implementation of a variable capacity compressor as opposed to a fixed capacity compressor [8].

Additionally, there have been some optimization studies on the heat exchangers. It was

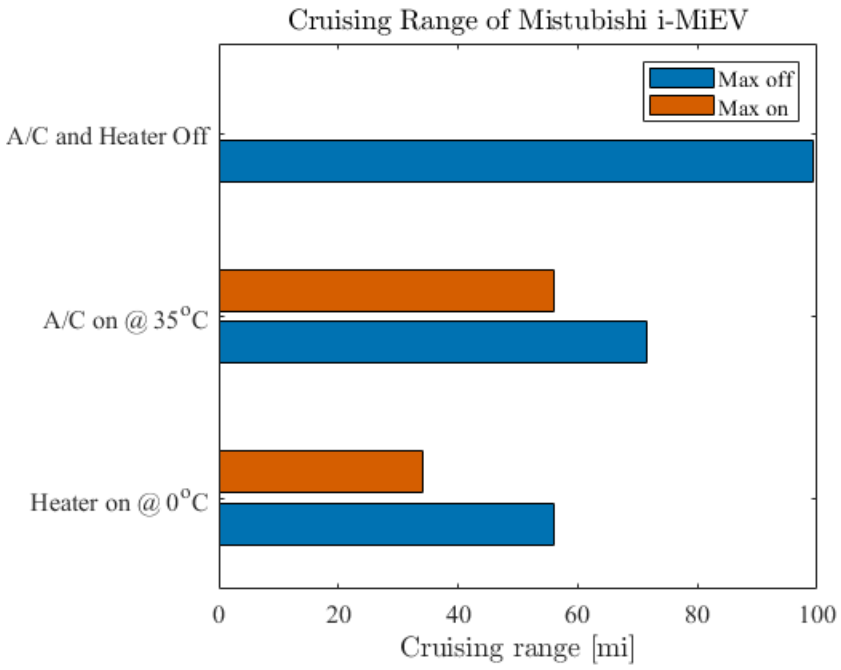


Figure 1.3: Mitsubishi i-MiEV driving range under different operating conditions. Adapted from data from [6]

found that micro-channeled heat exchangers have a greater benefit in their compactness, weight, and heat transfer characteristics when compared to more conventional heat exchangers. Also, work has been done to optimize the expansion valve. Traditionally, thermal expansion valves (TEVs) are used; however, these often have slow response times and low response precision. The electronic expansion valve (EEV) was first explored in [9]. Then [10] described the improvement of the control method using an EEV, as there is more precision and a faster response time with EEVs.

On the other hand, several controls strategies have been implemented with the aim to improve the EV TMS efficiency. Rule-based approaches are currently the most common controls approaches implemented, with the on/off control as a common choice during early implementation. After the on/off control, Proportional-Integral-Derivative (PID) control is another common approach, due to its improved performance over the on/off control while still maintaining ease of implementation. More recently model based approaches have been pursued, such as model predictive control (MPC).

1.2 Vapor Compression System Modeling and Control

Vapor compression systems are one of the most commonly implemented thermal management systems because they are efficient and reduce humidity. VCSs typically use a refrigerant as

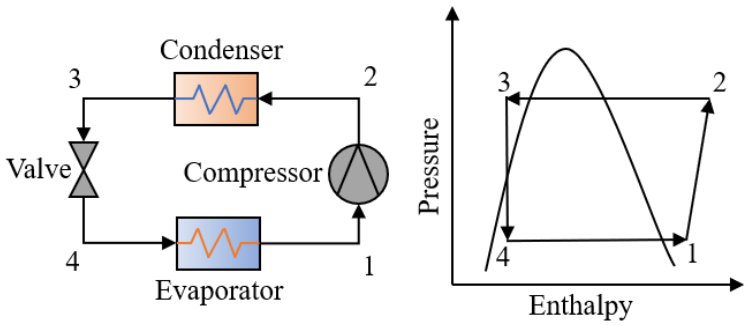


Figure 1.4: Subcritical VCS diagram.

the working fluid, with R-134a, R-22 and R-744 being the most commonly implemented. The VCS consists of four main components: compressor, condenser, valve, and evaporator. Figure 1.4 provides a diagram of how these components are connected and the P-h diagram of a VCS. The compressor provides work to the system by taking the refrigerant at low pressure and compressing it such that the fluid exits as a high pressure vapor. Then the high pressure vapor enters the condenser where it rejects heat to the secondary fluid (typically air). The refrigerant usually exits the condenser as a high pressure two-phase fluid. Then the refrigerant enters an expansion valve where it exits as a low pressure two-phase fluid. While there are many types of expansion valves, this thesis will focus on the electronic expansion valve (EEV), at it has a faster response than the TEV. Lastly, the refrigerant absorbs heat from the secondary fluid in the evaporator, exiting as a vapor. The heat absorbed in the evaporator by the working fluid is considered the cooling capacity of the system.

One common variation to the VCS is the addition of a receiver at the exit of the evaporator. When a receiver is used in the VCS, the working fluid exiting the evaporator is usually in the two-phase region. This helps maximize the efficiency of the evaporator, as the most heat transfer occurs when the fluid is in the two-phase region. The two-phase fluid then enters the receiver from which the compressor draws its fluid. The compressor pulls the fluid from the top portion of the receiver, ensuring only vapor enters the compressor. It is critical that only vapor enters the compressor, as any liquid could damage the component. For systems without a receiver, it is imperative that fluid exits as only vapor. Thus, there is usually a minimum goal of at least 3^oC of superheat, which slightly reduces its efficiency compared to a system with a receiver.

1.2.1 Modeling Challenges in Vapor Compression Systems

VCS modeling has long been studied as these systems contain complicated multi-timescale hydraulic and thermal dynamics, resulting in stiff systems. These challenging to model

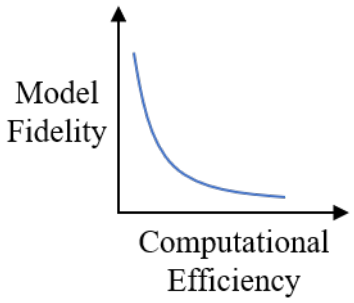


Figure 1.5: Model fidelity versus computational efficiency trade-off curve.

dynamics are due to the two-phase fluid present in the heat exchangers, as can be seen in Figure 1.4. Because of these dynamics, several different models with varying degrees of model fidelity have been derived in the literature. As with any type of modeling, there is a trade-off between the VCS model fidelity and the computational efficiency, with more complex models resulting in less computationally efficient models and less complex models resulting in higher computational efficiency (Figure 1.5). Often the model fidelity and computational efficiency depends on the modelling application. If the model is implemented in a real-time feedback control loop, then some fidelity might be sacrificed in order to improve computationally efficiency so that the controller can solve within the desired time frame. However, if the model is being implemented off-line, then higher model fidelity could be appropriate. This thesis will provide a model that breaks off this model fidelity - computational efficiency trade-off curve and creates a model that is both computationally efficient and has high model fidelity, as described in chapter 2.

1.2.2 Control Strategies for Vapor Compression Systems

The controls strategies for a VCS on a conventional ICE vehicle and that on an EV can look different. In an ICE vehicle the expansion valve and evaporator fan speed (when the system is in auto mode) are the main means of regulating the cabin temperature.

Historically, the compressor speed was not considered an actuator in the system, as the compressor was connected to the engine through a belt, making it challenging to accurately control the compressor speed. However, in modern systems and in EVs the compressor is controlled by an electric motor, which can easily be adjusted. As the compressor is the main component providing work to the refrigerant, the ability to control the compressor speed provides the ability to control the system more precisely.

There have been several different controls strategies implemented on VCSs in general. One of the most simple and commonly used control strategies for VCSs is the on/off control.

In this method the system runs until it reaches a desired temperature range, at which point the system will turn off. Once the temperature is outside the desired range, the system will turn back on again. While this is a very simple strategy that is easy to implement, it can be very inefficient. It also causes the system to go through unnecessarily large fluctuations and transients that could be avoided with a different control strategy. Another common control strategy is a PID control. This controller is popular as it maintains a similar ease of implementation, while providing an accurate and fast response. More recently, model based controllers have been explored in the literature as means of regulating the TMS. This thesis will explore the use of model predictive control as a means of controlling the VCS on an EV.

1.3 Organization of Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents the different mathematical models implemented in this thesis, including a novel extension to the existing graph-based modeling framework. Chapter 3 outlines the baseline PI controller and provides results from its implementation. Chapter 4 introduces model predictive control (MPC) and presents its implementation and results on the cabin thermal management system of an electric vehicle. It also includes a comparison and discussion on the results from the MPC and PI controls. Chapter 5 concludes the work and presents areas for future work.

Chapter 2

Modeling

This chapter presents the physics-based models used to simulate a VCS for an EV and those implemented in the feedback control loop. The system will be formulated through component-based modeling, in which each component model is derived individually. Component-based modeling allows components to easily be interchanged and rearranged within the system architecture, without having to reformulate the entire system model. There are several existing frameworks, toolboxes, and applications that are conducive to component-based modeling. One such framework that has been developed for thermofluid, electrical, and mechanical systems is Thermosys. Additionally, system models have been implemented in object-oriented programming languages, such as Modelica [11]. Also, a graph-based modeling framework has been developed for thermal, electrical, mechanical and fluid systems [12], [13]. Graph-based models have proven to be particularly useful for conservation-based systems and result in efficient models that improve on the computational efficiency. Because of these benefits, graph-based modeling will be the focus of this modeling chapter.

The remainder of this chapter is organized as given. Section 2.1 provides an overview of the single-state vertex graph-based modeling approach. Section 2.2 extends this graph-based modeling framework to include multi-state vertices, allowing more complicated system dynamics to be modeled in the graph-based modeling framework. Section 2.3 provides a summary of different common modeling approaches for dynamic, nonlinear heat exchangers and presents a multi-state vertex graph-based model for these heat exchangers. Section 2.4 overviews the modeling methods for the remaining components in an MPC. Section 2.5 presents the complete VCS graph-based model and verifies its accuracy and computational complexity against existing models. Section 2.6 briefly describes the cabin model. Section 2.7 provides the complete system model. Finally, Section 2.8 describes the power consumption of the VCS.

2.1 Single-State Graph-Based Modeling

The graph-based modeling framework is a visual and mathematical method of representing models [13]. Graph models are particularly useful in modeling conservation-based systems [12], such as the thermal management system on an electric vehicle. In graph models the dynamics are contained in sets of edges and vertices. The vertices represent capacitive elements while the edges represent transport between two vertices. Edges are given an orientation to indicate the direction of transport. Additionally, sources and sinks are used to add and reject quantities to the system, respectively. Figure 2.1 shows an example of a graph model.

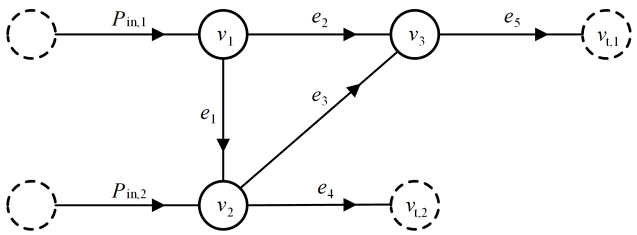


Figure 2.1: Notional single-state graph model with two sources, three internal vertices, and two sinks.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of order N_v and size N_e is composed of the set of vertices $\mathcal{V} = v_i, i \in [1 : N_v]$ connected by a set of oriented edges $\mathcal{E} = e_j, j \in [1 : N_e]$. Each vertex, v_i, is assigned a state, x_i, associated with the storage of a conserved quantity. Traditionally these conserved quantities are either energy or mass. Each edge, e_j, is assigned a value, P_j, describing the rate of transfer of the conserved quantity, such as power flow or mass flow rate. Each edge is oriented from its tail vertex, x_j^{tail}, to its head vertex, x_j^{head}. While flow can move in either direction, the edge orientation defines the convention of positive flow.

The dynamics of every vertex must satisfy either the conservation of energy or conservation of mass, depending on the state's domain. The dynamic conservation equation is given by

$$C_i \dot{x}_i = \sum P_{in} - \sum P_{out} \tag{2.1}$$

where C_i is the capacitance of vertex v_i, and P_{in} and P_{out} are the flows entering and exiting vertex v_i, respectively. In graph-based modeling, the flows are defined as a function of the head state, tail state, and inputs u_j

$$P_j = f(x_j^{head}, x_j^{tail}, u_j) \tag{2.2}$$

In addition to capturing the dynamics within the system, the graph-based modeling

approach accounts for the external factors affecting the system through the use of sources and sinks. Power and mass flow into the system are represented as sources, while power and mass flow out of the system are represented as sinks. Sources are modeled as edges $\mathcal{E}_s = e_{s,j}, j \in [1 : N_s]$ with flows $P^{in} = P^{in,j}$. Meanwhile, sinks are modeled as vertices $\mathcal{V}_t = v_{t,j}, j \in [1 : N_t]$ with sink states $x^t = x^{t,j}$ where $\mathcal{V}_t \subseteq \mathcal{V}$ but the sink states are not part of the vector x.

To map the connections between the edges and vertices in a graph, the incidence matrix $M = [m_{ij}] \in \mathbb{R}^{N_v \times N_e}$ is defined by

$$m_{ij} = \begin{cases} 1 & v_i \text{ is the tail of } e_j \\ -1 & v_i \text{ is the head of } e_j \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The incidence matrix can be partitioned into an upper and lower matrix as

$$M = \begin{bmatrix} \overline{M} \\ \underline{M} \end{bmatrix} \text{ with } \overline{M} \in \mathbb{R}^{(N_v - N_t) \times N_e} \tag{2.4}$$

where the edges are indexed such that $\overline{M}$ maps the flows P to states and $\underline{M}$ maps the flows P to the sink states. For example, M for the graph in Figure 2.1 is given by

$$M = \begin{bmatrix} \overline{M} \\ \underline{M} \end{bmatrix} = \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{array}\right] \tag{2.5}$$

Similarly, mapping of source flows to states is given by $D = [d_{ij}] \in \mathbb{R}^{(N_v - N_t) \times N_s}$, where

$$d_{ij} = \begin{cases} 1 & v_i \text{ is the head of } e^s_j \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

The D matrix for the graph in Figure 2.1 is given by

$$D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{2.7}$$

From (2.1), (2.3), and (2.6), the dynamics of the graph system can be written as

$$C\dot{\mathbf{x}} = -\overline{M}P + DP^{in} \tag{2.8}$$

where $C = \text{diag}([C_i])$ is a diagonal matrix of the capacitance of each vertex, $\mathbf{x} = [x_i] \in \mathbb{R}^{N_d}$ are the states associated with the dynamic vertices, $P = [P_i] \in \mathbb{R}^{N_e - N_s}$ is a vector of each edge that is not from a source vertex, and $P^{in} = [P_i^{in}] \in \mathbb{R}^{N_s}$ is a vector of the flows from source vertices.

The flows can also be represented in nonlinear form as

$$P = F(\mathbf{x}, \mathbf{x^t}, \mathbf{u}). \tag{2.9}$$

This results in a nonlinear dynamic system represented by

$$C\dot{\mathbf{x}} = -\overline{M}F(\mathbf{x}, \mathbf{x^t}, \mathbf{u}) + DP^{in}. \tag{2.10}$$

While the nonlinear form captures more complex dynamics, graph-based models are often linearized to reduce the computation time and to be used in various feedback control structures.

2.2 Multi-State Graph-Based Modeling

While single-state graph-based models have proven to be a useful tool in modeling many energy management systems, they fail to adequately model systems in which multiple states are strongly coupled, such as in two-phase flow dynamics commonly found in HVAC systems. This is largely due to the formulation in (2.2), as the flows within a graph model may only be functions of adjacent vertex states. This means that each edge can only be a function of a maximum of two dynamic variables. However, this is not always true in physical systems. Additionally, the capacitance of a vertex can only be a function of inputs or the state variable assigned to that vertex. There are many times that the capacitance might be a function of more than one dynamic variable. These deficiencies in the single-state graph-based model

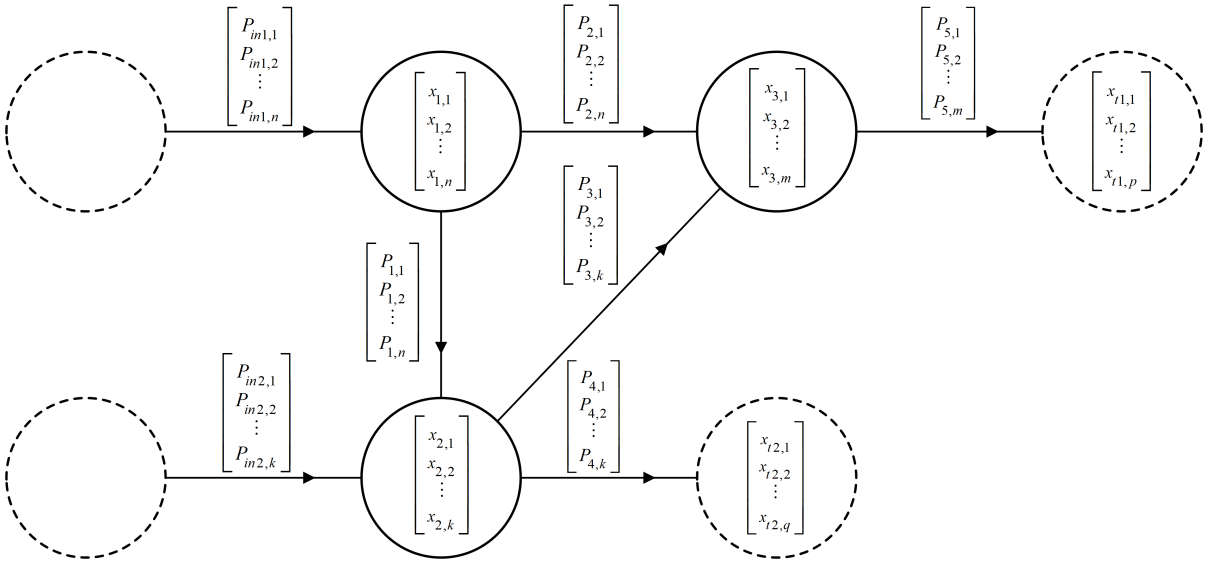


Figure 2.2: Multi-state graph model. The notation $x_{i,j}$ defines state j of vertex i. Similarly, $P_{i,j}$ is flow j of edge i. In this case it is assumed that $n \geq k \geq m$.

demonstrate the need for a multi-state graph-based modeling framework, which is an extension of the single-state graph-based modeling framework [14]. This extensions allows us to include more complex dynamic systems that might not be adequately captured in a single-state graph model. An example of a multi-state vertex graph-based model is depicted in Figure 2.2.

In the multi-state graph model formulation, each vertex is associated with at most n states of the overall system. The multi-state graph model dynamics are given as

$$C\dot{\mathbf{x}} = -(\overline{M} * S_{\overline{M}})P(\mathbf{x}, \mathbf{u}) + (D * S_D)P^{in} \tag{2.11}$$

Two modifications are made to the single-state graph formulation, (2.8), to accommodate multi-state vertices. First, the capacitance matrix is a block diagonal matrix, as opposed to a diagonal matrix. The size of each block is given by the number of states in each vertex. For example, the capacitance matrix for Figure 2.2 is given by

$$C = \begin{bmatrix} C_1 \in \mathbb{R}^{n\times n} & 0 & 0 \\ 0 & C_2 \in \mathbb{R}^{k\times k} & 0 \\ 0 & 0 & C_3 \in \mathbb{R}^{m\times m} \end{bmatrix}. \tag{2.12}$$

Also, two adaptive matrices, denoted as $S_{\overline{M}}$ and S_D, are used to map multi-flow edges to the states. These matrices adapt the incidence matrix and the D matrix to be compatible with the multi-state vertices and flows. The Khatri-Rao product, $*$, is a variation of the Kronecker product, $\bigotimes$. It is used to help ensure the matrix dimensions are correct. An

example of how the Khatri-Rao product works is given by

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \\ \hline A_{31} & A_{32} \end{array}\right], B = \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \\ \hline B_{31} & B_{32} \end{array}\right], A * B = \left[\begin{array}{c|c} A_{11} \bigotimes B_{11} & A_{12} \bigotimes B_{12} \\ \hline A_{21} \bigotimes B_{21} & A_{22} \bigotimes B_{22} \\ \hline A_{31} \bigotimes B_{31} & A_{32} \bigotimes B_{32} \end{array}\right] \tag{2.13}$$

The Khatri-Rao product is applied to block matrices, therefore $\overline{M}$ and D need to be block matrices. To create $\overline{M}$ and D for multi-state graphs, begin by formulating the single-state vertex $\overline{M}$ and D. Then treat each element in $\overline{M}$ and D as its own block matrix. $\overline{M}$ and D for Figure 2.2 are

$$\overline{M} = \left[\begin{array}{c|c|c|c|c} 1 & 1 & 0 & 0 & 0 \\ \hline -1 & 0 & 1 & 1 & 1 \\ \hline 0 & -1 & -1 & 0 & 1 \end{array}\right], D = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \\ \hline 0 & 0 \end{array}\right]. \tag{2.14}$$

Note that these matrices are the same as those presented in (2.5) and (2.7). The only difference is that they are block matrices.

The incidence adaptive matrix, $S_{\overline{M}}$, is composed of block matrices such that $S_{\overline{M}} \in [\mathbb{R}]^{(N_v - N_t) \times N_e}$. Each block $S_{i,j}$ in $S_{\overline{M}}$ is of size $S_{i,j} \in \mathbb{R}^{N_{x,v_i} \times N_{P,e_j}}$ where N_{x,v_i} is the number of states in vertex v_i and N_{P,e_j} is the number of power flows in edge e_j. For the example graph model in Figure 2.2, $S_{\overline{M}}$ is divided into three rows of block matrices, one for each internal vertex, and five columns, one for each edge,

$$S_{\overline{M}} = \left[\begin{array}{c|c|c|c|c} S_{1,1} \in \mathbb{R}^{n \times n} & S_{1,2} \in \mathbb{R}^{n \times n} & S_{1,3} \in \mathbb{R}^{n \times k} & S_{1,4} \in \mathbb{R}^{n \times k} & S_{1,5} \in \mathbb{R}^{n \times m} \\ \hline S_{2,1} \in \mathbb{R}^{k \times n} & S_{2,2} \in \mathbb{R}^{k \times n} & S_{2,3} \in \mathbb{R}^{k \times k} & S_{2,4} \in \mathbb{R}^{k \times k} & S_{2,5} \in \mathbb{R}^{k \times m} \\ \hline S_{3,1} \in \mathbb{R}^{m \times n} & S_{3,2} \in \mathbb{R}^{m \times n} & S_{3,3} \in \mathbb{R}^{m \times k} & S_{3,4} \in \mathbb{R}^{m \times k} & S_{3,5} \in \mathbb{R}^{m \times m} \end{array}\right]. \tag{2.15}$$

To determine the value of each element in the incidence adaptive matrix, $S_{\overline{M}}$, (2.16) is used.

$$[S_{m,n}]_{i,j} = \begin{cases} 1 & \text{if } C(\dot{x})_{i,m} = f(P_{j,n}) \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

(2.16) states that if state m of vertex v_i is a function of flow n in edge e_j, then a 1 is placed in the m^{th} row of the n^{th} column of the (i^{th}, j^{th}) block of S. All other elements in the adaptive matrix are 0. This ensures that for each block in $S_{\overline{M}}$ only one element in each row is equal to 1; the remaining elements in the row are 0.

Similarly, the power in adaptive matrix, S_D, is composed of block matrices such that $S_D \in [\mathbb{R}]^{(N_v - N_t) \times N_s}$. Each block, $S_{i,j}$, in S_D, is of size $S_{i,j} \in \mathbb{R}^{N_{x,v_i} \times N_{P,e_{s,j}}}$ where N_{x,v_i} is the number of states in vertex v_i and $N_{P,e_{s,j}}$ is the number of power flows in source edge $e_{s,j}$. For the example in Figure 2.2, the S_D matrix is divided into three rows of blocks, one for each vertex, and two columns, one for each external power flow

$$S_D = \left[\begin{array}{c|c} S_{1,1} \in \mathbb{R}^{n \times n} & S_{1,2} \in \mathbb{R}^{n \times k} \\ \hline S_{2,1} \in \mathbb{R}^{k \times n} & S_{2,2} \in \mathbb{R}^{k \times k} \\ \hline S_{3,1} \in \mathbb{R}^{m \times n} & S_{3,2} \in \mathbb{R}^{m \times k} \end{array}\right]. \tag{2.17}$$

To determine the value of each element in the adaptive matrix, S_D, (2.18) is used.

$$[S_{m,n}]_{i,j} = \begin{cases} 1 & \text{if } C(\dot{x})_{i,m} = f(P_{s,j,n}) \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

Note that if $n, k, m, p, q = 1$ (i.e. the system is full of single state vertices and edges), then both adaptive matrices would be all ones, such that $\overline{M} * S_{\overline{M}} = \overline{M}$ and $D * S_D = D$. This is consistent with the single-state vertex graph-based model formulation presented in Section 2.1.

2.3 Heat Exchanger Model Formulations

Dynamic VCS heat exchanger modeling has been thoroughly studied, as the two-phase fluids present in these heat exchangers contain strongly coupled and multi-timescale hydraulic and thermal dynamics. These dynamics create stiff systems which are computationally expensive to simulate. There are three main physics-based methods used to model heat exchangers: lumped parameter, moving boundary (MB), and finite volume (FV).

Lumped parameter modeling, as defined in [15], applies lumped parameter assumptions to either the entire heat exchanger as one control volume or to specific fluid phases in the heat exchanger (superheat vapor, two-phase, subcooled liquid) [16]-[17]. While lumped parameter models tend to be more simple than their MB or FV counterparts, they tend to lack the ability to capture high-order transient behavior. With recent improvements in computational power and better accuracy in the FV and MB approaches, MB and FV models have been the preferred VCS heat exchanger modeling method over lumped parameter methods.

The MB approach, first presented in [18], discretizes the heat exchanger in up to 3 control

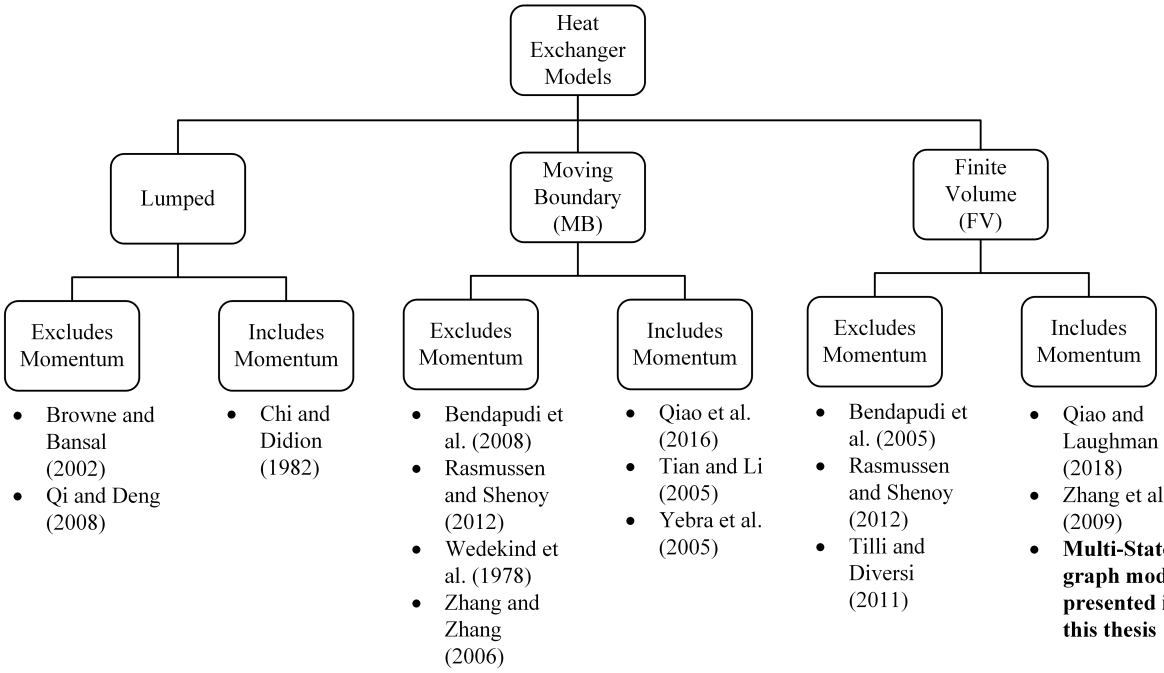


Figure 2.3: Tree Diagram of different types of heat exchanger modeling approaches with examples of each type from the literature

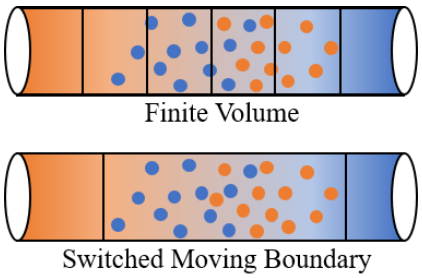


Figure 2.4: FV model with 6 CV and SMB heat exchanger diagrams

volumes, one per each fluid phase (superheated vapor, two-phase, subcooled liquid), as shown in Figure 2.4. MB models differ from lumped parameter models in that each fluid phase length can vary with time. The MB modeling approach was expanded with the development of the switched moving boundary (SMB) approach [19]. SMB models allow fluid phases to appear and disappear without causing numerical issues. Typically MB and SMB models neglect pressure drops [20]-[21]; however, there are noteworthy exceptions, such as those presented by [22]-[23].

The FV approach, first presented in [24], discretizes the heat exchanger into equal sized control volumes (CVs). The discretization level is dependent on the application, as an increase in the number of control volumes leads to an improvement in model accuracy at the expense of increased computational costs. As with the lumped parameter and MB approaches, FV

models can be categorized based on whether or not they include momentum. If it is desired to capture the pressure drop, then the conservation of momentum should be included in the model. Either transient momentum equations or steady state momentum equations can be used to capture pressure drop. However, the inclusion of transient momentum introduces more dynamic states into an already stiff system model. Because of this added complexity, researchers have chosen to either neglect momentum effects all together [11],[15],[25] or include only the steady state effects [11],[26]-[27]. Comparison of models that include transient conservation of momentum, steady state conservation of momentum, and exclude momentum can be found in [11],[27].

In this thesis the multi-state graph-based model will be used to simulate the system. However, the SMB approach will be the model used in the observer and MPC. This is because the graph-based model includes significantly more dynamic states than the SMB models which is not as conducive for real-time feedback control. Thus, the SMB model is less complicated to linearize and easier to implement in the observer and MPC.

2.3.1 Heat Exchanger Dynamics

This section presents the first principles and modeling assumptions used in deriving the FV, SMB, and graph-based models of the heat exchangers. All three formulations rely on the same first principles, with minor differences in the assumptions during the derivation steps and the integration limits. A discussion of the modeling essentials are provided below, but the reader is referred to [28] for an in-depth derivation of the FV approach and [29] for the SMB approach. There are several simplifying assumptions that are applied when deriving two-phase heat exchanger models

1. The heat exchanger is a long, thin, circular, uniform, and horizontal tube.
2. There is one dimensional flow along the tube.
3. Refrigerant flow is only in the positive direction.
4. Axial heat conduction in the refrigerant is negligible.
5. Two phase flow is homogeneous.
6. The dynamics on the air side are negligible.

Applying the above assumptions to the conservation of refrigerant mass, conservation of

energy in the refrigerant, and conservation of energy in the tube wall, results in

$$A_{cs,r}\frac{\partial \rho_r}{\partial t} + \frac{\partial \dot{m}_r}{\partial z} = 0 \tag{2.19}$$

$$A_{cs,r}\frac{\partial (\rho h - P)_r}{\partial t} + \frac{\partial (\dot{m} h)_r}{\partial z} = Q_r \tag{2.20}$$

$$(A_{cs}\rho c_p)_w \frac{\partial T_w}{\partial t} = Q_a - Q_r \tag{2.21}$$

where the subscripts r, a, and w denote the refrigerant, air, and wall, respectively. $A_{cs,r}$ denotes the cross-sectional area of the tube, ρ denotes density, $\dot{m}$ denotes mass flow rate, h denotes enthalpy, P denotes pressure, T denotes temperature, c_p denotes specific heat of the wall material, z denotes the length of the tube, and Q_a and Q_r are the heat transfer between the heat exchanger wall and the respective fluid.

2.3.2 Heat Exchanger Switched Moving Boundary Formulation

The SMB modeling formulation uses one zone per fluid phase (i.e. superheated vapor, two-phase fluid, and subcooled liquid), resulting in at most 3 zones. These zones can appear and disappear as needed, resulting in distinct operating modes for both the condenser and evaporator. The condenser and evaporator modes are depicted in Figure 2.5. These mode definitions will carry throughout the remainder of this thesis.

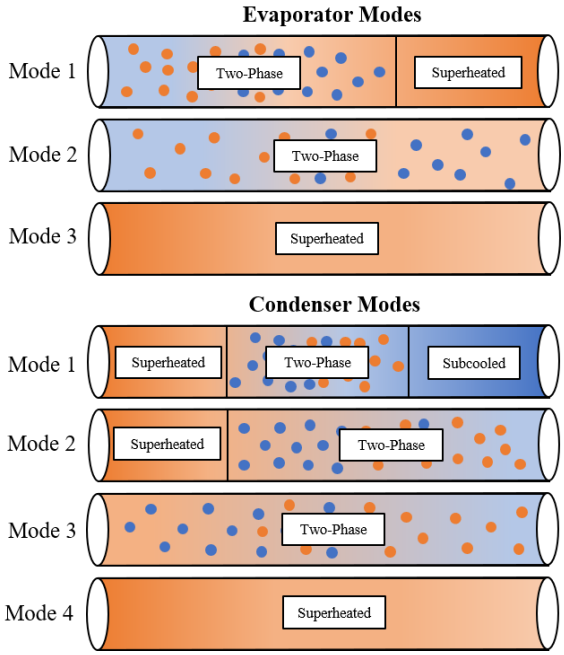


Figure 2.5: Switched Moving Boundary Modes.

A summary of the derivation steps for SMB approach is provided below. The full derivation can be found in [29].

1. To begin, take (2.19)-(2.21) and integrate them along the length of the fluid phase.
2. Assume lumped parameters for the refrigerant, wall, and air properties.
3. Use the product rule as necessary to obtain derivatives in the correct states.
4. Normalize the lengths by dividing each equation by the total length of the heat exchanger.

An example derivation for the condenser in mode 1 is provided below. The reader is referred to [29] for the full derivation of each mode. A diagram of a condenser in mode 1 is provided in Figure 2.6.

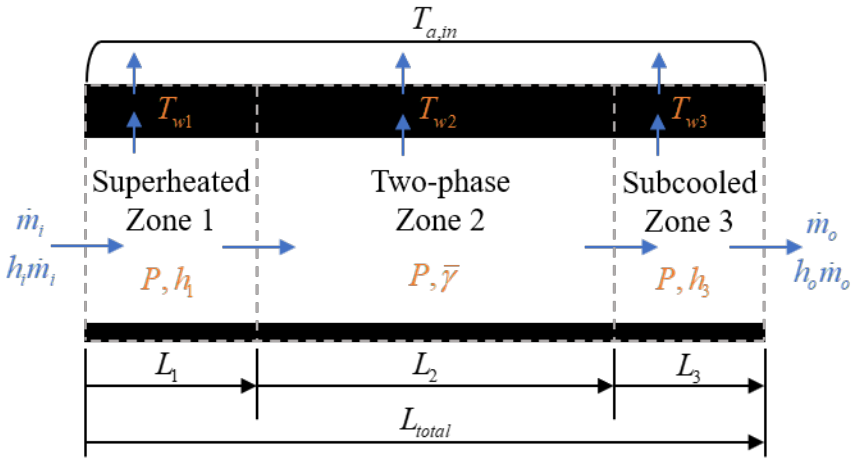


Figure 2.6: SMB model of a condenser in Mode 1. Orange variables are dynamic states and blue represents power or mass flow.

The first step in deriving the SMB heat exchanger model is integrating (2.19) - (2.21) along the correct fluid lengths. For the superheated zone, the integration bounds are 0 to L_1. The two-phase zone is integrated from L_1 to $L_1 + L_2$. The subcooled zone is integrated from $L_1 + L_2$ to L_{total}. Then the product rule is applied to obtain equations in the correct derivatives for the model. Lastly, the equation is divided by the total refrigerant length to obtain normalized lengths ζ_1, ζ_2,and ζ_3. Upon integrating and applying the product rule, the superheated zone equations are:

$$\zeta_1 \frac{d\rho_1}{dt} + (\rho_l - \rho_v)\frac{d\zeta_1}{dt} = \frac{\dot{m}_{r,in} - \dot{m}_{SH:TP}}{V_{total}} \tag{2.22}$$

$$\zeta_1 \left(\frac{d\rho_1 h_1}{dt} - \frac{dP}{dt} \right) + (\rho_1 h_1 - \rho_v h_v)\frac{d\zeta_1}{dt} = \frac{\dot{m}_{r,in} h_{r,in} - \dot{m}_{SH:TP} h_v + A_s \zeta_1 \alpha_r (T_{w1} - T_{r1})}{V_{total}} \tag{2.23}$$

$$\zeta_1 \frac{dT_{w1}}{dt} + \left[T_{w1} - T_w|_{L_1} \right] \frac{d\zeta_1}{dt} = \frac{A_s \zeta_1 \left[\alpha_r (T_{r1} - T_{w1}) + \alpha_a (T_a - T_{w1}) \right]}{m_w c_p} \tag{2.24}$$

The two-phase zone equations are:

$$\zeta_2 \frac{d\rho_2}{dt} + (\rho_2 - \rho_f)\frac{d\zeta_2}{dt} + (\rho_v - \rho_l)\frac{d\zeta_1}{dt} = \frac{\dot{m}_{SH:TP} - \dot{m}_{TP:SC}}{V_{Total}} \tag{2.25}$$

$$\zeta_2 \left[\frac{d(\rho_2 h_2)}{dt} - \frac{dP}{dt} \right] + [\rho_2 h_2 - \rho_f h_f]\frac{d\zeta_2}{dt} + (\rho_v h_v - \rho_f h_f)\frac{d\zeta_1}{dt} = \tag{2.26}$$

$$\frac{\dot{m}_{SH:TP} h_v - \dot{m}_{TP:SC} h_f + A_s \zeta_2 \alpha_r (T_{w2} - T_{r2})}{V_{total}} \tag{2.27}$$

$$\zeta_2 \frac{dT_{w2}}{dt} + \left[T_{w2} - T_w|_{L_1+L_2} \right] \frac{d\zeta_2}{dt} + \left[T_w|_{L_1} - T_w|_{L_1+L_2} \right] \frac{d\zeta_1}{dt} = \tag{2.28}$$

$$\frac{A_s \zeta_2 \left[\alpha_r (T_{r2} - T_{w2}) + \alpha_a (T_a - T_{w2}) \right]}{m_w c_p} \tag{2.29}$$

The subcooled zone equations are:

$$\zeta_3 \frac{d\rho_3}{dt} + [\rho_f - \rho_3] \left[\frac{d\zeta_1}{dt} + \frac{d\zeta_2}{dt} \right] = \frac{\dot{m}_{TP:SC} - \dot{m}_{r,out}}{V_{total}} \tag{2.30}$$

$$\zeta_3 \left[\frac{d\rho_3 h_3}{dt} - \frac{dP}{dt} \right] + (\rho_f h_f - \rho_3 h_3) \left[\frac{d\zeta_1}{dt} + \frac{d\zeta_2}{dt} \right] = \frac{\dot{m}_{TP:SC} h_f - \dot{m}_{r,out} h_{r,out} + A_s \zeta_3 \alpha_r (T_{w3} - T_{r3})}{V_{Total}} \tag{2.31}$$

$$\zeta_3 \frac{dT_{w3}}{dt} + \left[T_w|_{L_1+L_2} - T_{w3} \right] \left[\frac{d\zeta_1}{dt} + \frac{d\zeta_2}{dt} \right] = \frac{A_s \zeta_3 [\alpha_r (T_{r3} - T_{w3}) + \alpha_a (T_a - T_{w3})]}{m_w c_p} \tag{2.32}$$

This results in 9 dynamics states in the condenser in Mode 1 model.

$$x = \begin{bmatrix} \zeta_1 & \zeta_2 & P & h_1 & h_3 & \bar{\gamma} & T_{w1} & T_{w2} & T_{w3} \end{bmatrix} \tag{2.33}$$

However, it is clear that there will not always be 9 dynamic states in the condenser model, as fluid regions can appear and disappear when the condenser mode changes. This suggests

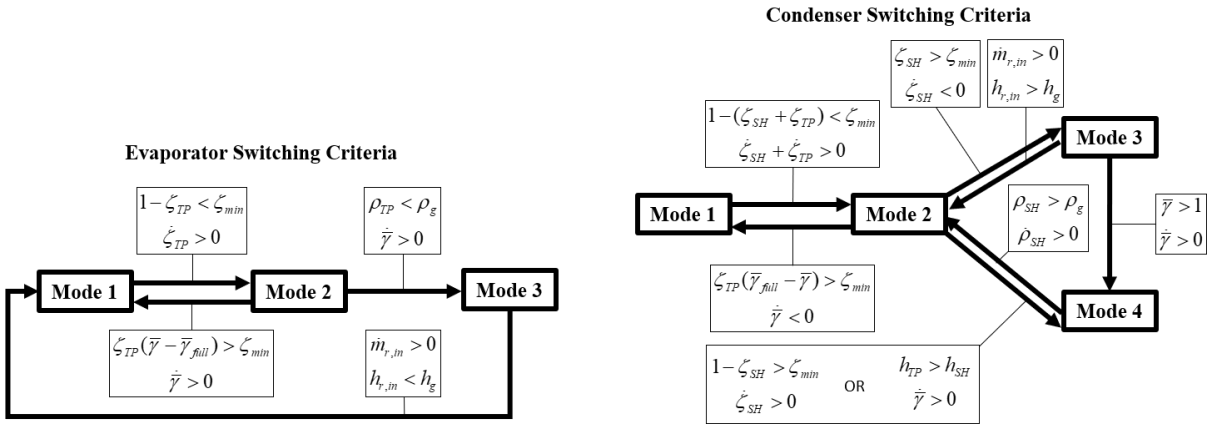


Figure 2.7: SMB switching criteria for condenser and evaporator

the need for pseudo-state equations. Pseudo-state equations are implemented to track states that are no longer present in the current moving boundary mode. For instance, the condenser in mode 2 does not contain the states associated with subcooled fluid, so those states would be tracked as a pseudo-state. More details on pseudo-states can be found at [29].

Additionally, switching criteria must be implemented in order to smoothly transition the model from one mode to another. More details on the switching criteria can be found at [29]. A summary of the switching criteria is provided in Figure 2.7.

To find the various parameters in (2.22)-(2.32) several different correlations are used. As done in [33], a lookup table was generated using REFPROP to determine the fluid density, temperature, and specific heat. The details of REFPROP are outlined in [34]. To determine the heat transfer coefficients (HTC), different correlations are used depending on the fluid phase. The specific correlations used in this work are provided in Table 2.1. To avoid numerical problems caused by the discontinuities when changing HTC correlations, the HTC profile was smoothed near the phase change region as was done in [33].

Component	Fluid Phase	Correlation
Condenser	Superheated Vapor	Gnielinski [30]
	Two-Phase	Dobson and Chato [31]
	Subcooled Liquid	Gnielinski [30]
Evaporator	Superheated Vapor	Gnielinski [30]
	Two-Phase	Wattelet [32]
	Subcooled Liquid	Gnielinski [30]

Table 2.1: Heat Transfer Coefficient Correlations

2.3.3 Heat Exchanger Finite Volume Formulation

When creating a heat exchanger model using the finite volume approach, the heat exchanger is divided into n equally sized control volumes. A refrigerant pressure, refrigerant enthalpy and wall temperature state are associated with each CV. Mass and power flows between each CV as shown in Figure 2.8.

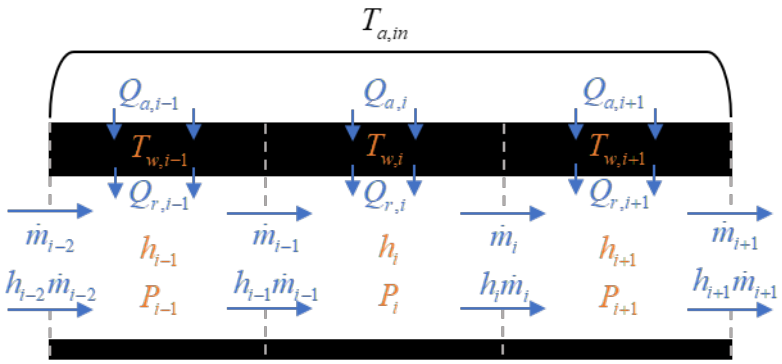


Figure 2.8: Finite Volume Model Schematic. Orange variables are dynamic states and blue represents power or mass flow.

A summary of the derivation is provided below. Note that while the SMB model formulation had slightly different approaches for deriving the condenser and evaporator models, the FV approach is the same for both heat exchangers. The full derivation of the finite volume heat exchanger can be found in [28].

1. (2.19)-(2.21) are integrated for each CV length.
2. The refrigerant, wall, and air properties are treated as lumped properties for each CV.
3. The refrigerant density and derivative of the density is treated as a function of enthalpy and pressure states.

This results in

$$\left[\frac{\partial \rho_{r,i}}{\partial P_{r,i}}\bigg|_{h_{r,i}} A_{cs,r} L_i\right] \dot{P}_{r,i} + \left[\frac{\partial \rho_{r,i}}{\partial h_{r,i}}\bigg|_{P_{r,i}} A_{cs,r} L_i\right] \dot{h}_r + \dot{m}_{out} - \dot{m}_{in} = 0 \tag{2.34}$$

$$\left[\frac{\partial \rho_{r,i}}{\partial P_{r,i}}\bigg|_{h_{r,i}} h_{r,i} - 1\right] A_{cs,r} L_i \dot{P}_{r,i} + \left[\frac{\partial \rho_{r,i}}{\partial h_{r,i}}\bigg|_{P_{r,i}} h_{r,i} + \rho_{r,i}\right] A_{cs,r} L_i \dot{h}_r + \dot{m}_{out} h_{out} - \dot{m}_{in} h_{in} = Q_{r,i} \tag{2.35}$$

$$(mc_p)_w \frac{dT_{w,i}}{dt} = Q_{a,i} - Q_{r,i} \tag{2.36}$$

for each CV, where $Q_{a,i}$ and $Q_{r,i}$ are given by

$$Q_{r,i} = \alpha_{r,i} A_{s,i} (T_{w,i} - T_{r,i}) \tag{2.37}$$

$$NTU = \frac{\alpha_{a,i} A_{s,a}}{\dot{m}_a c_{p,a}} \tag{2.38}$$

$$Q_{a,i} = \frac{\dot{m}_a c_{p,a}}{n} \left(T_{a,in} - \left[T_{w,i} + (T_{a,in} - T_{w,i}) e^{-NTU} \right] \right) \tag{2.39}$$

where α is the heat transfer coefficient and n is the total number of control volumes. The effectiveness number of transfer units (e-NTU) method was used to determine the heat transfer between the air and tube wall.

After identifying the continuity and conservation of energy equations for each CV, the inclusion or exclusion of pressure drop effects must be considered. While pressure drop is often neglected, there are certain heat exchanger applications where pressure drop effects become more significant, such as in micro-channeled heat exchangers. The pressure drop in the heat exchanger is evaluated by analyzing the steady-state momentum equation (2.40)

$$P_{in,i} - P_{out,i} + \frac{\dot{m}_{r,i}^2}{A_{cs,i}^2} \left(\frac{1}{\rho_{in,i}} - \frac{1}{\rho_{out,i}} \right) + \frac{\dot{m}_{r,i}^2 f}{2 \rho_{r,i} A_{cs,i}^2 D} = 0 \tag{2.40}$$

where f is the friction factor, L is the length along the control volume, and D is the tube diameter. The minor loss coefficient is determined by the heat exchanger geometry and known coefficients published in the literature [35]. The pressure drop can be a result of friction, acceleration drop and gravitational effects in the heat exchanger. [11] showed that the pressure drop is dominated by the friction losses, and therefore, it is assumed that the acceleration drop and gravitational effects are negligible. Additionally, minor losses are considered as some heat exchangers may have geometries in which minor losses are critical in the heat exchanger pressure drops. For instance, if there are a significant number of bends in the heat exchanger tube, then the minor losses would have a greater impact on the pressure drop. Rearranging (2.40) with only the friction loss term and adding minor loss into the equation, the refrigerant mass flow rate can be calculated by

$$\dot{m}_{r,i} = A_{cs,r} \sqrt{\frac{2(P_{in,i} - P_{out,i})}{\frac{Lf}{\rho_{r,i} D} + \frac{K_L}{\rho_{r,i}}}} \tag{2.41}$$

where K_L is the minor loss coefficient.

A FV heat exchanger with 3 CVs is provided below as an example of how this formulation can be implemented. Combining equations (2.34)-(2.39), the conservation of refrigerant mass

can be written as:

$$\begin{bmatrix} C_{i,1} & C_{i,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{i,1} & C_{i,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{i,1} & C_{i,2} \end{bmatrix} \begin{bmatrix} \dot{P}_{r,1} \\ \dot{h}_{r,1} \\ \dot{P}_{r,2} \\ \dot{h}_{r,2} \\ \dot{P}_{r,3} \\ \dot{h}_{r,3} \end{bmatrix} = \begin{bmatrix} \dot{m}_{r,in} - \dot{m}_{r,1} \\ \dot{m}_{r,1} - \dot{m}_{r,2} \\ \dot{m}_{r,2} - \dot{m}_{r,3} \end{bmatrix} \tag{2.42}$$

where

$$C_{i,1} = \left.\frac{\partial \rho_{r,i}}{\partial P_{r,i}}\right|_{h_{r,i}} V_{r,i} \tag{2.43}$$

$$C_{i,2} = \left.\frac{\partial \rho_{r,i}}{\partial h_{r,i}}\right|_{P_{r,i}} V_{r,i}. \tag{2.44}$$

The conservation of refrigerant energy equations can be expressed as:

$$\begin{bmatrix} C_{j,1} & C_{j,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{j,1} & C_{j,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{j,1} & C_{j,2} \end{bmatrix} \begin{bmatrix} \dot{P}_{r,1} \\ \dot{h}_{r,1} \\ \dot{P}_{r,2} \\ \dot{h}_{r,2} \\ \dot{P}_{r,3} \\ \dot{h}_{r,3} \end{bmatrix} = \begin{bmatrix} \dot{m}_{r,in} h_{r,in} - \dot{m}_{r,1} h_{r,1} + \alpha_{r,1} A_{s,1} (T_{w,1} - T_{r,1}) \\ \dot{m}_{r,1} h_{r,1} - \dot{m}_{r,2} h_{r,2} + \alpha_{r,2} A_{s,2} (T_{w,2} - T_{r,2}) \\ \dot{m}_{r,2} h_{r,2} - \dot{m}_{r,3} h_{r,3} + \alpha_{r,1} A_{s,1} (T_{w,3} - T_{r,3}) \end{bmatrix} \tag{2.45}$$

where

$$C_{j,1} = \left(\left.\frac{\partial \rho_{r,j}}{\partial P_{r,j}}\right|_{h_{r,j}} h_{r,j} - 1 \right) V_{r,j} \tag{2.46}$$

$$C_{j,2} = \left(\left.\frac{\partial \rho_{r,j}}{\partial h_{r,j}}\right|_{P_{r,j}} h_{r,j} + \rho_{r,j} \right) V_{r,j}. \tag{2.47}$$

The intermediate mass flow rates for (2.42)-(2.45) are calculated using (2.41). Lastly, the conservation of heat exchanger wall energy equations can be expressed as:

$$\begin{bmatrix} m_{w,1} c_{p,w} & 0 & 0 \\ 0 & m_{w,2} c_{p,w} & 0 \\ 0 & 0 & m_{w,3} c_{p,w} \end{bmatrix} \begin{bmatrix} \dot{T}_{w,1} \\ \dot{T}_{w,2} \\ \dot{T}_{w,3} \end{bmatrix} = \begin{bmatrix} \alpha_{r,1} A_{s,1} (T_{r,1} - T_{w,1}) + Q_{a,1} \\ \alpha_{r,2} A_{s,2} (T_{r,2} - T_{w,2}) + Q_{a,2} \\ \alpha_{r,3} A_{s,3} (T_{r,3} - T_{w,3}) + Q_{a,3} \end{bmatrix} \tag{2.48}$$

If pressure drop effects are neglected, then only one pressure is considered for the entire heat exchanger, as opposed to each CV having its own pressure. Each intermediate mass flow rate is placed in the state vector and (2.41) is ignored, as $P_{in,i} = P_{out,i}$. (2.34)-(2.36) remain the same, except every $P_{r,i}$ is equal and denoted by one pressure, P_r. A three CV heat exchanger example is given, as in [33]. For a three CV heat exchanger model approach, the continuity equations can be formulated as:

$$\begin{bmatrix} V_{r,1}\left(\frac{\partial \rho_{r,1}}{\partial P_r}\Big|_{h_{r,1}}\right) & & 1 & 0 \\ V_{r,2}\left(\frac{\partial \rho_{r,2}}{\partial P_2}\Big|_{h_{r,2}}\right) & I_{3\times 3} \times V_{r,i}\left(\frac{\partial \rho_{r,1}}{\partial h_{r,i}}\Big|_{P_r}\right) & -1 & 1 \\ V_{r,3}\left(\frac{\partial \rho_{r,3}}{\partial P_3}\Big|_{h_{r,3}}\right) & & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{P}_r \\ \dot{h}_{r,1} \\ \dot{h}_{r,2} \\ \dot{h}_{r,3} \\ \dot{m}_{r,1} \\ \dot{m}_{r,2} \end{bmatrix} = \begin{bmatrix} \dot{m}_{r,in} \\ 0 \\ -\dot{m}_{r,out} \end{bmatrix} \tag{2.49}$$

The conservation of refrigerant energy equations for a 3 CV heat exchanger without pressure drop effects can be formulated as:

$$\begin{bmatrix} V_{r,1}\left(\frac{\partial \rho_{r,1}}{\partial P_r}\Big|_{h_{r,1}} h_{r,1} - 1\right) & & h_{r,1} & 0 \\ V_{r,2}\left(\frac{\partial \rho_{r,2}}{\partial P_r}\Big|_{h_{r,2}} h_{r,2} - 1\right) & V_{r,i}Z & -h_{r,1} & h_{r,2} \\ V_{r,3}\left(\frac{\partial \rho_{r,3}}{\partial P_r}\Big|_{h_{r,3}} h_{r,3} - 1\right) & & 0 & -h_{r,2} \end{bmatrix} \begin{bmatrix} \dot{P}_r \\ \dot{h}_{r,1} \\ \dot{h}_{r,2} \\ \dot{h}_{r,3} \\ \dot{m}_{r,1} \\ \dot{m}_{r,2} \end{bmatrix} = \begin{bmatrix} \dot{m}_{r,in} h_{r,in} + A_{s,1}\alpha_{r,1}(T_{w,1} - T_{r,1}) \\ A_{s,2}\alpha_{r,2}(T_{w,2} - T_{r,2}) \\ -\dot{m}_{r,out} h_{r,out} + A_{s,3}\alpha_{r,3}(T_{w,3} - T_{r,3}) \end{bmatrix} \tag{2.50}$$

where

$$Z = diag\left(h_{r,1}\left.\frac{\partial \rho_{r,1}}{\partial h_{r,1}}\right|_{P_r} + \rho_{r,1}, h_{r,2}\left.\frac{\partial \rho_{r,2}}{\partial h_{r,2}}\right|_{P_r} + \rho_{r,2}, h_{r,3}\left.\frac{\partial \rho_{r,3}}{\partial h_{r,3}}\right|_{P_r} + \rho_{r,3}\right).$$

The conservation of wall energy equations are expressed using (2.48).

In (2.48)-(2.50) only the temperatures, pressure and enthalpies need integration, the intermediate mass flow rates do not. There are a few ways to address this. In [33], the mass flow rates are integrated through the use of a filter. The filter was implemented to improve robustness and speed of the model. Additionally, the intermediate mass flow rates can be eliminated through manipulations of the conservation equations, as done in [15].

2.3.4 Heat Exchanger Graph Model Formulation

The graph-based model of the heat exchanger utilizes the FV approach with pressure drop included. As the hydraulic and thermal dynamics are tightly coupled, a two-state vertex graph model is needed to accurately capture the coupling. This can be seen as the mass flow rates are functions of pressures and enthalpies. Figure 2.9 depicts the two-state vertex model for a heat exchanger with n control volumes.

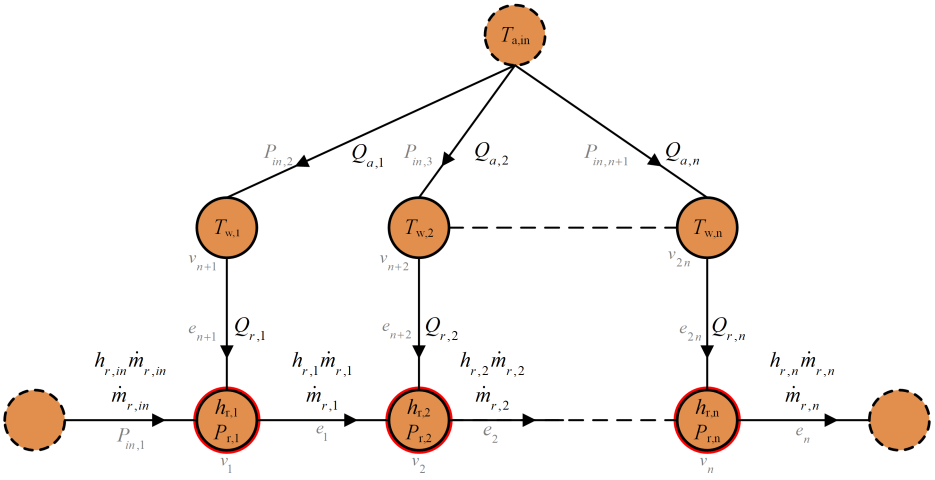


Figure 2.9: Graph-based model of a heat exchanger. Dashed edges represent an arbitrary number of control volumes.

States and power flows are denoted in black, while vertex and edge indices are denoted in gray. The vertices outlined in red are the two-state vertices that represent the refrigerant pressure and enthalpy in each CV in the heat exchanger model. The three vertices labeled with $T_{w,i}$ represent the heat exchanger wall energy storage associated for each CV. The dashed vertices represent interactions with external systems, such as the air side of the heat exchanger and the flow inlet and exit. The edges between the refrigerant vertices have two rows: one for the power flow and another for the mass flow. The edges connected to the wall temperature vertices only have one row, as only heat transfer occurs between the wall and surrounding fluid.

While the vertex and edge indices are arbitrary, indices need to be established in order to define adaptive matrices. The adaptive matrices for Figure 2.9 can be calculated using (2.16) and (2.18). If each block in the capacitance matrix has the first row corresponding to the conservation of energy and the second row corresponding to continuity, then the adaptive matrices for Figure 2.9 are given by:

$$S_{\bar{M}} = \left[\begin{array}{cc|cc|cc|c|c} 1 & 0 & \cdots & \cdots & 1 & 0 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 1 & 0 & \ldots & \ldots & 1 & 0 & 1 & \ldots & 1 \\ 0 & 1 & \ldots & \ldots & 0 & 1 & 0 & \ldots & 0 \\ \hline 1 & 0 & \ldots & \ldots & 1 & 0 & 1 & \ldots & 1 \\ \hline \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 1 & 0 & \ldots & \ldots & 1 & 0 & 1 & \ldots & 1 \end{array}\right], S_D = \left[\begin{array}{cc|c|c|c} 1 & 0 & 1 & \ldots & 1 \\ 0 & 1 & 0 & \ldots & 0 \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 1 & 0 & 1 & \ldots & 1 \\ 0 & 1 & 0 & \ldots & 0 \\ \hline 1 & 0 & 1 & \ldots & 1 \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline 1 & 0 & 1 & \ldots & 1 \end{array}\right] \tag{2.51}$$

An example of a multi-state vertex graph-based model with 3 CVs is provided in Appendix A. As mentioned in Section 2.2, one of the challenges with single-state vertex graph-based models is the convention that power flows must be a function of at most one state per adjacent vertex. However, whenever dynamics are tightly coupled, as in the case of two-phase heat exchangers, the dynamics may not adhere to the single-state vertex graph modeling framework. It is clear through the two-phase heat exchanger example that the multi-state vertex graph model formulation advances the single-state convention to capture more complex system dynamics. The refrigerant mass flow rates are a function of the upstream pressure, downstream pressure and fluid enthalpy which could not be executed with the single-state vertex graph-based models.

2.4 Other Component Models

2.4.1 Electronic Expansion Valve Modeling

The nonlinear model for an electronic expansion valve (EEV) presented below comes directly from [28]. Since the dynamics in the EEV are significantly faster than the thermal dynamics in the heat exchangers, the EEV is treated as an algebraic model. The mass flow rate is given by (2.52), where C_d is the discharge coefficient given by a mapping between the valve input and pressure differential $\Delta P = P_{in} - P_{out}$.

$$\dot{m}_v = C_d\sqrt{\rho(P_{in} - P_{out})} \tag{2.52}$$

$$C_d = f(u_v, \Delta P) \tag{2.53}$$

$$h_{v,in} = h_{v,out} \tag{2.54}$$

Since the states in the EEV are treated as algebraic, the graph-based model is simply a single edge that connects the vertices from the upstream and downstream components (Figure 2.10).

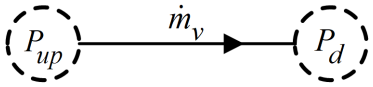


Figure 2.10: EEV Graph Model. P_{up} is the pressure vertex from the upstream component and P_d is the pressure from the downstream component.

In the case of the VCS, the upstream pressure is the outlet pressure from the condenser and the downstream pressure is the inlet pressure from the evaporator.

2.4.2 Compressor Modeling

Two different compressor models are presented in this section. The first model is taken directly from [28] and can be represented statically or dynamically, by adding a first order filter to the static model. The second model is a graph-based model given in [36]. This model assumes that the compressor enthalpy is dynamic and accounts for heat loss from the refrigerant to the atmosphere. Both models will be implemented in this thesis. The graph-based model will be implemented with the heat exchanger and EEV graph-based models to complete the VCS. The other compressor model will be used as the linear model of the system for the model predictive control (MPC) and observer.

2.4.2.1 Compressor Model without Heat Loss Effects

Since the compressor is first treated as a static model in [28], it is represented by two algebraic equations. The mass flow rate is calculated by (2.55). The compression is assumed to be adiabatic with an isentropic efficiency, η_k, so the outlet enthalpy can be calculated by rearranging (2.56), resulting in (2.57), where $h_{out,s} = h(P_{out}, s_k)$ and $s_k = s(P_{in}, h_{in})$.

$$\dot{m}_k = \omega_k V_k \rho_k \eta_{vol} \tag{2.55}$$

$$\frac{h_{out,s} - h_{in}}{h_{out} - h_{in}} = \eta_k \tag{2.56}$$

$$h_{out} = h_{in} + \frac{1}{\eta_k}(h_{out,s} - h_{in}) \tag{2.57}$$

$$\eta_{vol} = f_1(P_r, \omega_k) \tag{2.58}$$

$$\eta_k = f_2(P_r, \omega_k) \tag{2.59}$$

This static model can also be represented dynamically by adding a first order filter to the outlet enthalpy. The dynamic outlet enthalpy is calculated by

$$\dot{h}_{out} = \frac{1}{\tau}(h_{out,static} - h_{out}) \tag{2.60}$$

where τ is the time constant of the first order filter.

2.4.2.2 Dynamic Graph-Based Compressor Model

The second dynamic compressor modeling method is based on first principles and is implemented through the graph-based modeling approach [36].Figure 2.11 shows the graph-based compressor model when the motor is turned on.

This model has three internal vertices corresponding to the three different conservation of energy equations. First, there is power going into the system through an electric outlet. It is assumed that the power goes to a motor with an efficiency, η_m. Most of the motor work is used to compress the refrigerant flowing through the compressor. However, there are some inefficiencies in the form of wasted heat to the compressor shell. The motor vertex, P_E in figure 2.11, is assumed to be algebraic and can be represented by

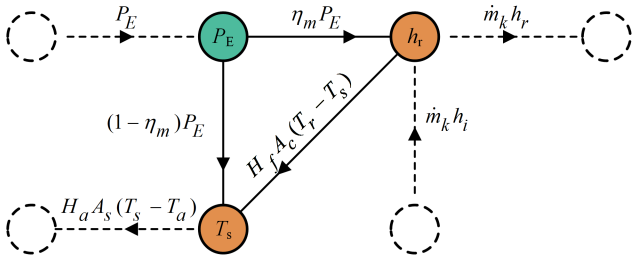


Figure 2.11: Graph-based compressor model when the motor is turned on

$$0 = P_E - \eta_m P_E - (1 - \eta_m) P_E. \tag{2.61}$$

where P_E is the electric power coming from the wall outlet.

Additionally, the energy in the refrigerant is conserved. The power flows affecting the refrigerant are the power associated with fluid flow into the compressor, power associated with fluid flow out of the compressor and the compressor power. The dynamics of the refrigerant can be represented by

$$\frac{dh}{dt} = \frac{1}{\rho V_k}(\eta_m P_E + \dot{m}_k h_i - \dot{m}_k h_r - H_f A_c (T_r - T_s)). \tag{2.62}$$

where ρ is the density of the refrigerant at the inlet, V_k is the compressor volume, $\dot{m}_k$ is the refrigerant mass flow rate, h_i is the enthalpy at the compressor inlet, h_r is the enthalpy at the compressor outlet, H_f is the convective heat transfer coefficient, A_c is the associated surface area, T_r is the outlet temperature of the refrigerant, and T_s is the shell temperature. Note that the mass flow rate of the refrigerant is calculated by (2.55), as is done in the static case.

Lastly, the energy in the compressor shell is conserved. The compressor is a treated as a lumped thermal storage with heat transfer between the refrigerant, ambient conditions, and motor inefficiencies. The compressor shell vertex, represented by T_s in Figure 2.11, can be expressed by the following equation

$$\frac{dT_s}{dt} = \frac{1}{M_k c_k}((1 - \eta_m) P_E + H_f A_c (T_r - T_s) + H_a A_s (T_s - T_{amb})). \tag{2.63}$$

where M_k is the mass of the compressor shell, c_k is the specific heat of the compressor shell, H_a is the convective heat transfer coefficient, A_s is the associated surface area, and T_{amb} is the ambient temperature. There is also a graph model case for when the motor is off and the compressor is no longer actively running. This case is not considered in this thesis, but the interested reader can refer to [36] for more details.

2.5 VCS Model Verification

2.5.1 VCS Graph Model Formulation

The graph model for a VCS is provided in Figure 2.12, where the orange vertices represent thermal vertices and the green vertices represent electrical vertices. Both heat exchangers were discretized into 10 CVs for the model verification, as there was diminishing improvement in the model behavior for more than 10 CVs. The component parameters for this system are given in Table 2.2.

2.5.2 Simulation Results

The graph-based heat exchanger model presented in this thesis is verified against the experimentally validated FV model presented in [37]. To begin with this verification, two different VCS models were constructed. One VCS model is the graph-based model presented in Figure 2.12. The second VCS model replaces the two heat exchanger models with the finite control volume model presented in [37]. Thus, the only difference between the two models is the heat exchanger models.

To test the transient response of the two models, the valve opening and compressor speed were given step commands, as shown in Figure 2.13. A comparison of the condenser and evaporator results between the multi-state graph-based model and the model presented in [37] is displayed in Figure 2.14.

Both the evaporator and condenser results are verified with only slight offsets. The

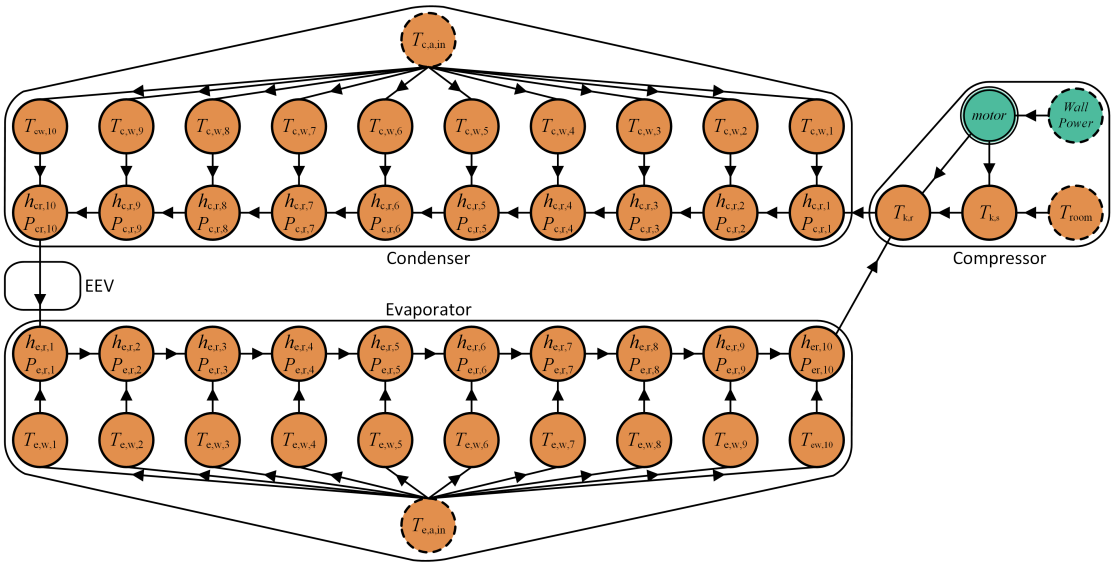


Figure 2.12: Graph model of a VCS with 10 control volumes per heat exchanger.

Condenser	
Hydraulic diameter [m]	8.1×10^{-3}
Total length [m]	10.7
Wall mass [kg]	4.66
Wall specific heat $[kJkg^{-1}K^{-1}]$	0.467
Inlet Air Mass Flow Rate $[kgs^{-1}]$	0.220
Inlet Air Temperature $[^oC]$	24.0
Evaporator	
Hydraulic diameter [m]	8.01×10^{-3}
Total length [m]	11.5
Wall mass [kg]	2.74
Wall specific heat $[kJkg^{-1}K^{-1}]$	0.488
Inlet Air Mass Flow Rate $[kgs^{-1}]$	0.120
Inlet Air Temperature $[^oC]$	22.9
Compressor	
Mass [kg]	18.0
Shell specific heat $[kJkg^{-1}K^{-1}]$	0.460
Motor efficiency [kg]	0.900
Volume $[m^3]$	8.05×10^{-5}
Surface area $[m^2]$	3.50×10^{-3}

Table 2.2: VCS Simulation Parameters

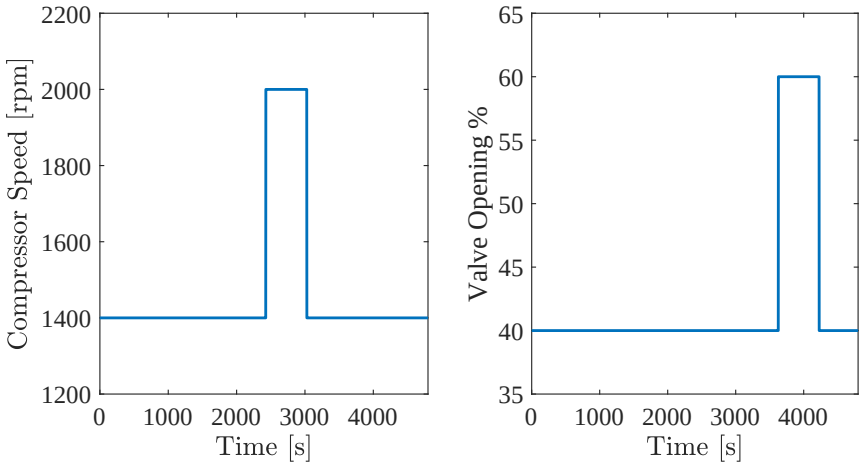
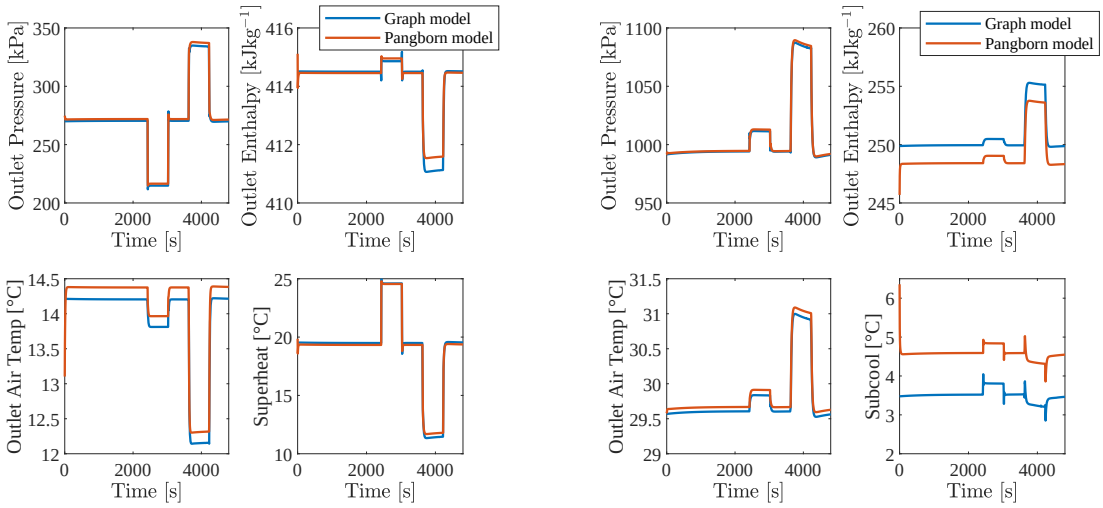


Figure 2.13: Actuator inputs for graph-based model validation

deviation in the outlet pressure and enthalpy in the condenser and evaporator is within 1% difference. Some offset is expected because, as [27] points out, there is steady-state deviation between models that include pressure drop and those that exclude pressure drop. The offset may also be a result of the discretization method. The graph model uses a modified upwind method, where the outlet enthalpy is equal to the enthalpy in the last CV. This is to adhere



(a) Evaporator results

(b) Condenser results

Figure 2.14: Evaporator and condenser simulation results

to the power flow rule for graph models that states the power flow must be a function of the states in adjacent vertices. On the other hand, the model provided by [37] uses a modified central difference method, where the outlet enthalpy is found by

$$h_{r,out} \approx h_{r,N} + \frac{h_{r,N} - h_{r,N-1}}{2} \tag{2.64}$$

2.5.3 Computational Complexity Analysis

One of the many benefits of single-state vertex graph-based models is that the modeling approach is computationally efficient. This section aims to show that this still holds true for the multi-state vertex case, through the comparison of the multi-state graph-based model with other models commonly found in the literature.

2.5.3.1 Computation Time Comparison

The speed of a model can be measured by calculating the real time factor (RTF). RTF is the ratio of the CPU-time over the simulated time, with a smaller RTF being preferable. The graph-based model was run 10 times and the average CPU-time was 16.18 seconds for a 4800 second simulation, resulting in an RTF = 0.0034. This RTF is faster than most comparable models found in the literature. Figure 2.15 presents comparable existing models with published simulation times. Note that the x-axis is 1/RTF, so the faster models are located on the right-hand side of the plot. The tradeoff between the level of physics and

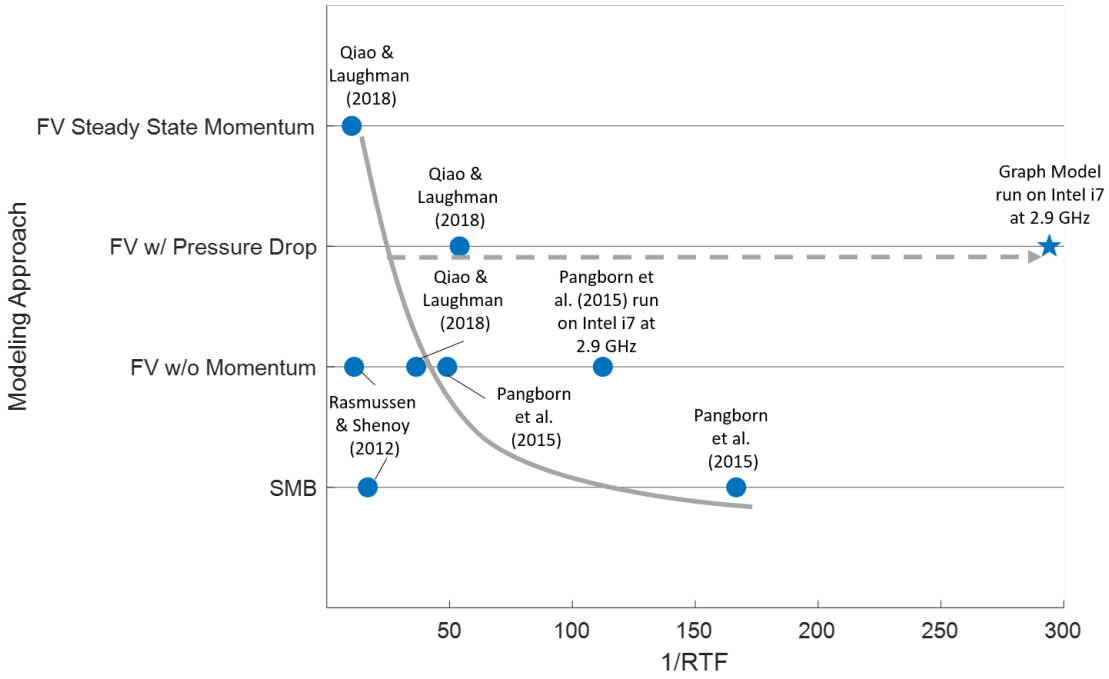


Figure 2.15: VCS speed versus modeling approach tradeoff curve

the computational speed is evident. Typically, models follow the trend where more detailed physics models have longer CPU-times. However, the graph-based model is not on this curve, as it is faster than other models while also including the pressure drop due to frictional losses.

The models presented in Figure 2.15 were chosen as comparison points because they modeled comparable loops and conditions. The model provided by [37] is a basic VCS with the four main components: compressor, condenser, EEV, and evaporator. The model presented by [11] is a heat pump and contains the four main VCS components plus an accumulator. The model provided by [15] is of a VCS with the four main components plus an accumulator and receiver. Other notable heat exchanger models have been developed, such as the one presented by [38]. Their FV model had 10 CVs and took 2.89 seconds to run a 625 second simulation. Meanwhile, their MB model took 0.73 seconds. Additionally, [39] created an object-oriented model of an evaporator that took 0.64 seconds to run a 15000 second simulation. However, in both [38] and [39], the heat exchangers were run individually as opposed to in a VCS loop.

The computational efficiency of the graph-based model can be attributed to several different factors. First, the nature of the graph-based models has a large impact on the computational speed. In the graph-based modeling framework, all of the component models are placed in a single dynamic equation which allows the solver to solve for all of the states

with vectorized equations in one calculation. This should be faster than component based toolboxes in which each component has a different function, and the solver must solve for the derivatives of each component individually. Also, the simulation conditions have a large impact on the computational speed. Some simulation profiles may drive the system through harsher transients than others, causing the simulation to take longer to run. Additionally, variations in the simulator, computer processor, and solver type affect the computational speed. While the graph-based models can be implemented in any programming language (MATLAB, Python, Modelica, etc.), this model was run using the variable step ode23tb solver in MATLAB. Variable solvers, such as ode23tb, tend to be much faster than fixed step solvers, because the computer can take smaller steps for the more complicated transients and larger steps for the more steady state sections of the simulation. Meanwhile, fixed step solvers must use the largest allowable step size that works for the entire simulation profile. This often results in the fixed step solver using much smaller time steps on average than the variable step solver. Also, the graph-based model presented in this thesis was simulated on a 2.9GHz Intel i7 processor which is likely a different type of processor than was used in other publications. Because the processor and simulation conditions have a significant impact on simulation speed, the model presented in [37] was simulated using ode23tb on the same computer and under similar conditions as the grpah-based model presented in this thesis. These results are presented in Figure 2.15. While the results in Figure 2.15 look promising, the model presented in [37] and the graph model were run on two different platforms, making it challenging to accurately compare the models.

2.5.3.2 Floating Point Operations Analysis

Because it proves challenging to fairly compare models run on different processors using only computational run time, the number of floating point operations (FLOPs) are counted for each model. FLOP analysis is particularly beneficial in capturing the computational complexity in matrix and vector operations as it counts the number of operations performed on numbers. Table 2.3, adapted from [40], provides a summary of the FLOP count for different matrix/vector operations. The number of FLOPs were counted for each time step in the graph-based, FV, and SMB models, with a summary of the results provided in Table 2.4.

From the results presented in Table 2.4, it is clear that the SMB approach has the least amount of FLOPs per time step. This is due to the fact that SMB models typically have significantly fewer dynamic states than its FV counterpart. The FV model approach with constant pressure has the second best FLOP count. This is due to the fact that the FV model without pressure drop only has one pressure state for each heat exchanger, while the graph-based model has a pressure state for each control volume in the heat exchanger.

Operation	FLOP
Matrix Addition (size $m \times n$)	mn
Matrix-Matrix product (size $m \times n$,$n \times p$)	$2mnp$
Matrix Inversion (size $n \times n$)	$4n^3 + 2n^2$
Vector-Vector Addition (size n)	n

Table 2.3: FLOP in Matrix/Vector Operations

Modeling Approach	FLOP count per time step (n CVs)	$n = 10$	$n = 50$
Graph Model	$846n^3 + 1104n^2 + 506n + 78$	9.80×10^5	1.11×10^8
FV Constant Pressure	$64n^3 + 128n^2 + 88n + 22$	7.77×10^4	8.32×10^6
SMB	4135	NA	NA

Table 2.4: FLOP analysis for different VCS models per time step

At first glance, the graph-based model seems to perform the worst in terms of FLOP count; however, there are multiple linear algebra techniques that could be applied to lower the number of FLOPs. For instance, one of the main factors contributing to a large number of FLOPs in the graph model is inversion of the large capacitance matrix. However, this matrix, by design, is a block diagonal matrix, so inverting the matrix could be done by inverting each individual block, as opposed to the entire matrix. This would significantly reduce the number of FLOPs. The FLOP count results with linear algebra simplifications are provided in Table 2.5.

Modeling Approach	FLOP count per time step (n CVs)	$n = 10$	$n = 50$
Graph Model	$168n^2 + 252n + 40$	1.94×10^4	4.33×10^5
FV Constant Pressure	$8n^3 + 44n^2 + 58n + 22$	1.30×10^4	1.11×10^6
SMB	4135	NA	NA

Table 2.5: FLOP analysis with reduction techniques

While the SMB model still has the least amount of FLOPs, the graph model performs better for larger number of CVs. This is due to the fact that the capacitance matrix in the graph model is composed of small diagonal blocks, while a typical FV model without pressure drop has a larger diagonal block. This allows the graph model to have a FLOP count that scales with n^2, but the FV model FLOP count scales with n^3 where n is the number of CVs. Thus, as the number of CVs increases, the graph model is more computationally efficient than the FV model.

2.6 Cabin Modeling

For completeness an overview of the car cabin thermal model is presented. Refer to [41] for full details of the car cabin model. The following assumptions need to be made in order to simplify the cabin model analysis:

1. The air inside the cabin is treated as a lumped parameter. (i.e. the air is the same temperature throughout the cabin)
2. Mass does not accumulate within the cabin (i.e. mass flow rate in equals mass flow rate out)
3. The radiative heat transfer with the interior surfaces is negligible
4. Solar radiation is split into vertical and horizontal components

There are five main heat loads acting on the cabin: solar radiation (Q_{solar}), heat generation from people within the vehicle (Q_{gen}), heat conduction through the vehicle walls and windows ($Q_{surfaces}$), heat leaking from the cabin to ambient air (Q_{leak}), and cooling from the VCS (Q_{cool}).

The dynamic equation governing the cabin temperature is given by:

$$C\dot{T}_{cab} = Q_{solar} + Q_{gen} + Q_{surfaces} + Q_{leak} + Q_{cool} \tag{2.65}$$

where C is the capacitance of the car air.

2.7 Complete System Model

Once all the system components have been modeled, they are combined to formulate the complete system model. The VCS model was formulated in section 2.5, so all that remains is

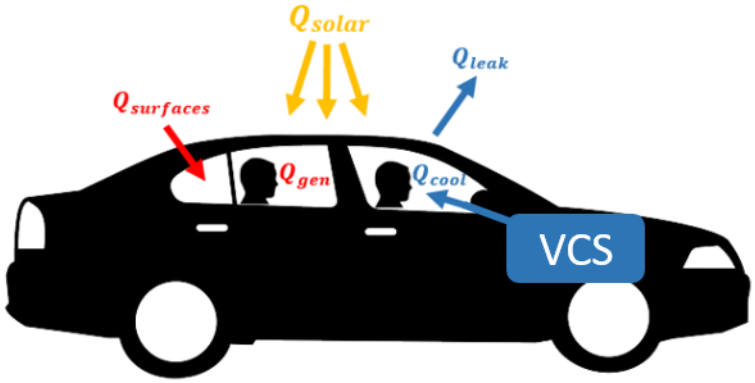


Figure 2.16: Heat loads on the cabin. Adapted from [41]

combining the VCS with the car cabin model. The VCS and car cabin model are connected through the evaporator. The evaporator outlet air is the fluid cooling the cabin. Meanwhile, the car cabin air combined with ambient air is used as the inlet air to the evaporator.

The following simulation parameters are used for the remainder of this thesis. These quantities are used to mimic a generic VCS sized for an automobile. Any of these quantities can be tuned to represent a different car with a different VCS.

Condenser	
Hydraulic diameter [m]	9.00E-4
Total length [m]	3.905
Wall mass [kg]	2.81
Wall specific heat $[kJkg^{-1}K^{-1}]$	0.89
Evaporator	
Hydraulic diameter [m]	2.40E-3
Total length [m]	0.428
Wall mass [kg]	1.863
Wall specific heat $[kJkg^{-1}K^{-1}]$	0.89
Compressor	
Surface area $[m^2]$	1.00E-6
Cabin	
Cabin volume $[m^3]$	3.11
Vehicle base area $[m^2]$	6.00
Vehicle roof area $[m^2]$	1.98
Vehicle wall area $[m^2]$	0.53
Vehicle windshield area $[m^2]$	0.82
Vehicle window area $[m^2]$	0.44

Table 2.6: Car System Parameters

2.8 Power Consumption

An important and commonly used metric in modeling and control of TMSs on EVs is the total power consumption of the EV. The two main actuators that require the most electrical power are the evaporator fan and the compressor. While the EEV requires some electrical power, it is considered minimal compared to the other actuators and can be neglected. Additionally, the two heat exchangers require no power input.

2.8.1 Evaporator Fan Power Consumption

The evaporator fan model was modified from data from an experimental setup at the University of Illinois, as done in [42]. Included with this data was a map of the electrical power as a function of fan speed. The maximum air mass flow rate is 0.12 kg/s and it consumed 180W. It was found that the electrical power is a quadratic function of the fan speed.

2.8.2 Compressor Power Consumption

The electrical power consumed by the compressor can be calculated by simple conservation of energy analysis. If it is assumed that the fluid goes through adiabatic compression in the compressor, then the electrical power can be calculated by

$$P_{comp} = \frac{\dot{m}_{comp}\Delta h_{comp}}{\eta_{comp}} \tag{2.66}$$

In reality, the compression of the refrigerant is not adiabatic as there is some heat transfer between the refrigerant and the ambient temperature. However, the heat transfer lost to the surroundings can be neglected as it is normally much smaller than the compression power. Thus, (2.66) is a low yet reasonable estimate of the total electrical power consumed by the compressor.

Chapter 3

Baseline Controllers

This chapter presents the baseline controller that will be compared with the model predictive controller analyzed in Chapter 4. Several different control strategies have been implemented for HVAC and VCSs. The most simple is the bang-bang strategy which turns the system on until the temperature reaches the desired reference temperature. Then the system turns off until the temperature deviates from the set point. While the bang-bang control strategy is the most simple, it is typically very inefficient [43]. The other common and inexpensive control method is the proportional-integral-derivative controller (PID). PID is one of the most commonly implemented control methods in thermal management of EVs because of its simplicity and ease of implementation. Because of this, PID control will be the baseline controller used in this thesis.

The rest of this chapter is organized as follows. Section 3.1 describes the controller problem. Section 3.2 presents the PI control loop to maintain the evaporator superheat within a given range and provides the results. Section 3.3 describes the PI control loop to maintain the cabin temperature within a given range and the results. Section 3.4 combines the controllers given in sections 3.2 and 3.3 and presents the results of the dual single-input single-output (SISO) PI controlled system.

3.1 Controller Problem Formulation

The thermal management system of the cabin in an EV has two main goals: maintain the cabin temperature at a specific set point and keep the evaporator superheat within a desired operating range. It is important to maintain the cabin temperature near a desired set point for passenger comfort. The evaporator superheat operating range is necessary to ensure safe and efficient operation of the VCS. If there is no superheat at the evaporator exit then liquid enters the compressor which could cause damage. Therefore, there is a lower limit on the

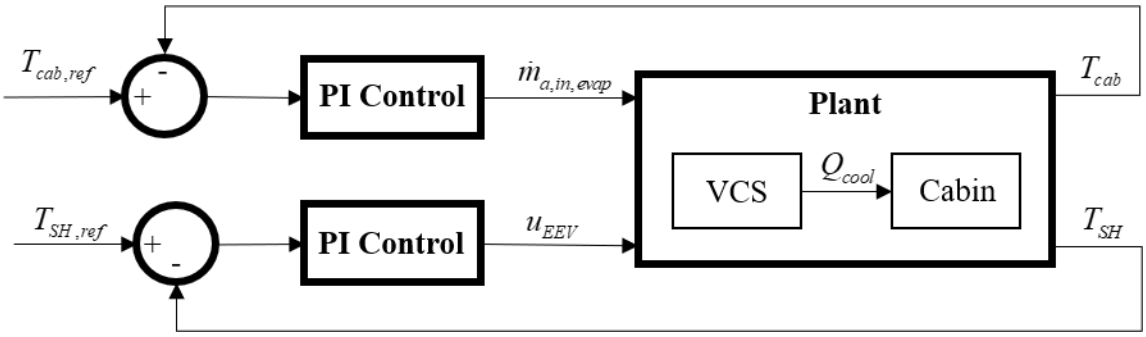


Figure 3.1: Dual PI controlled system

evaporator superheat to ensure safe operation. Additionally, the evaporator superheat is commonly used as a measure of efficiency as the efficiency of the system depends on the superheat at the outlet [44]. Heat transfer between the refrigerant and air is most efficient when the refrigerant is a two-phase fluid as all of the energy is used to convert the two-phase fluid to a vapor. Because of this, there is an upper limit placed on the evaporator superheat to keep the system operating efficiently.

In EVs there are up to three different actuators: the evaporator fan speed, the compressor speed, and the EEV opening. To manage both the cabin temperature and the evaporator superheat objectives, two SISO PI loops will be implemented. One loop will actuate the EEV to control the evaporator superheat. The other loop will actuate the evaporator fan speed to maintain the cabin temperature. This architecture is displayed in Figure 3.1. These two control loops were chosen in particular as they should not compete with one another. Occasionally the compressor is chosen to control the cabin temperature. However, this often results in the compressor and EEV competing with each other to reach their desired objectives as both of these devices control mass flow rate. In this work the compressor speed will be treated as constant and nominally chosen in the middle of the compressor's operating range.

The two main disturbances that affect the TMS are the air flow rate through the condenser and the ambient temperature. The air mass flow rate through the condenser is treated as a disturbance because it is largely dependent on the vehicle velocity. To generate the disturbance profile for this work, the SC03 driving schedule was used to generate the air flow rate [45]. The SC03 driving schedule is known as the air conditioning supplemental FTP driving schedule and is used by the United States Environmental Protection Agency (EPA) to test vehicle emissions and fuel economy. The driving schedule provides the vehicle speed and the condenser air flow rate was estimated as a function of the vehicle speed based on data from Ford Motor Company. The driving schedule is a 10 minute profile; however, it is repeated three times in this work resulting in a 30 minute profile. It was repeated to allow the model dynamics to settle out due to initial conditions in the first profile and then the

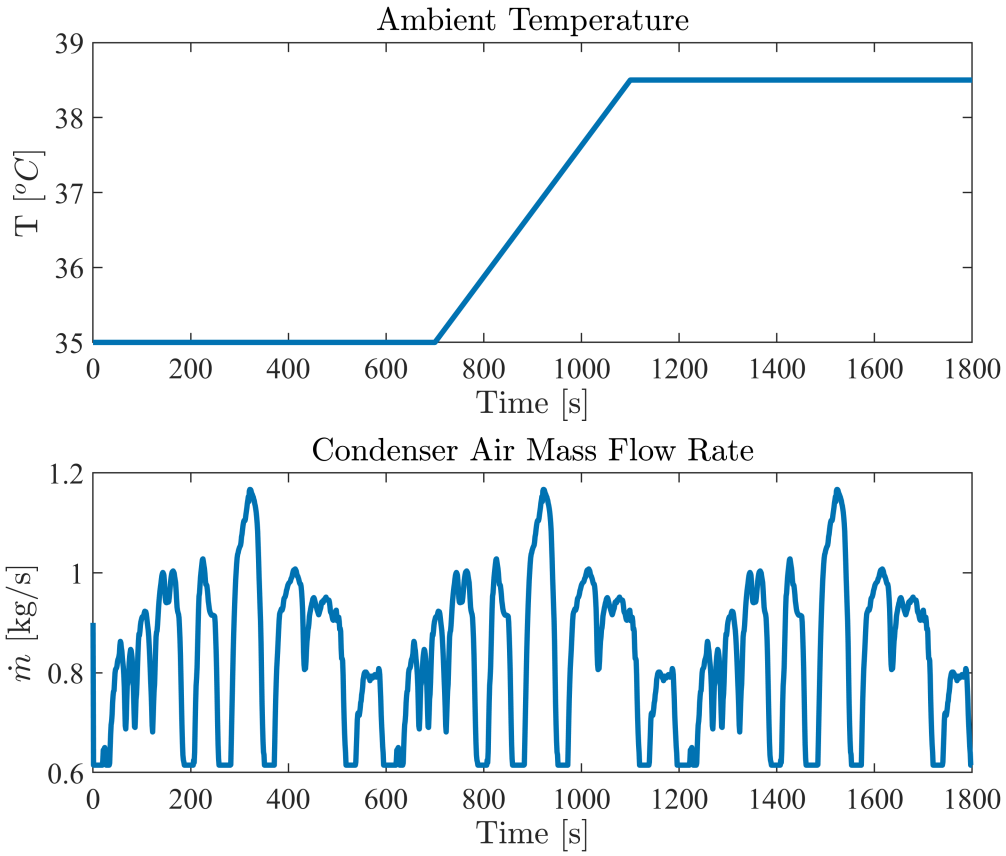


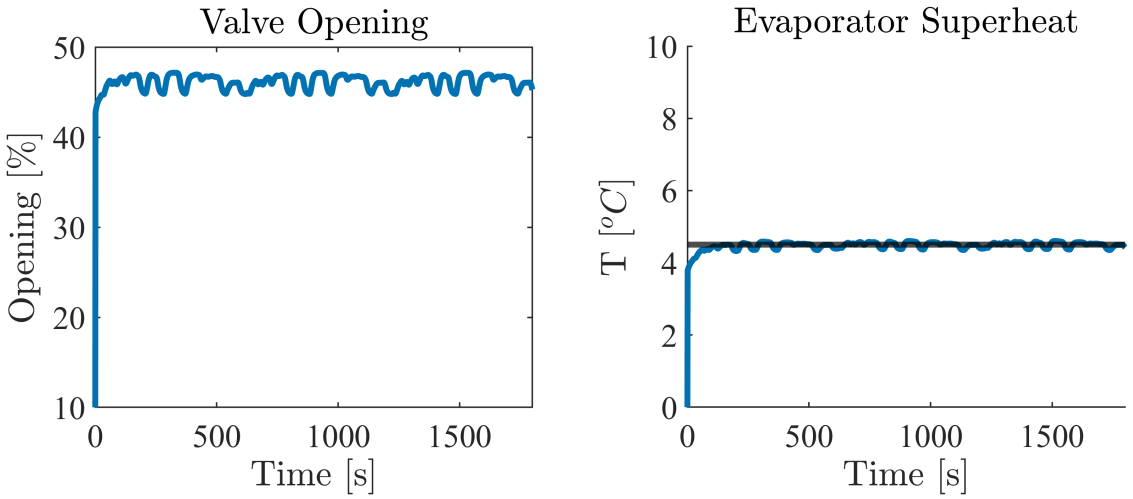
Figure 3.2: Disturbances on the EV

controller could be tested more accurately in the following two profiles. The disturbance profile used in this chapter and the following is presented in Figure 3.2. The ambient air temperature is treated as a ramp disturbance as the surrounding air can steadily change temperature, such as when there are large changes in elevation.

3.2 EEV-Superheat PI Formulation

At the beginning of the PI tuning process, each SISO PI loop was evaluated individually. To construct the superheat control loop, the compressor speed and evaporator fan speed were maintained at constant nominal values of 1800 rpm and 50%, respectively. A constant compressor speed of 1800 rpm was chosen because that is near the average speed of the car engine when driving in city conditions. A constant evaporator fan speed of 50% was chosen because that is the middle of the operating range. Then the disturbances were applied to the system and the gains were tuned heuristically. The chosen PI gains were P=10 and I=0.1. The results are displayed in Figure 3.3.

Outside of the initial 0.1 seconds, the PI controller was able to maintain the evaporator



(a) Actuator Command

(b) System Output

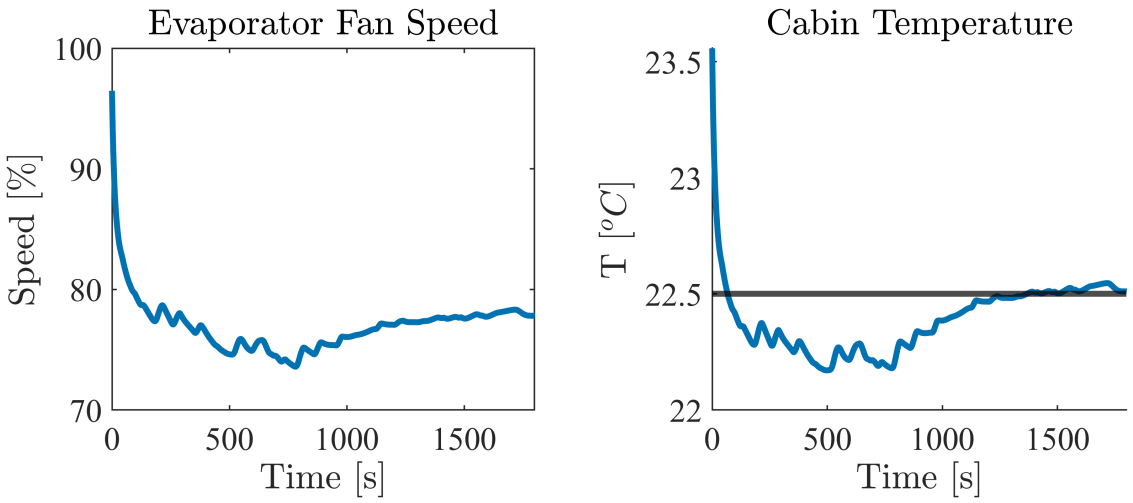
Figure 3.3: Evaporator and condenser simulation results

superheat within 0.17^{o}C of the reference temperature of 4.5 ^{o}C, with a maximum superheat of 4.60^{o}C and a minimum superheat of 4.33^{o}C. A reference superheat of 4.5 ^{o}C was chosen because this is the midpoint between the lower and upper bounds for the model predictive control (MPC) in chapter 4. The large fluctuation in valve opening percentage and error in the evaporator superheat in the initial 0.1 seconds is likely due to poor initial conditions of the model. Overall the root mean square error of the evaporator superheat is 0.07. This PI should be sufficient at controlling the superheat as the maximum deviations from the reference point are within 0.5^{o}C.

3.3 Evaporator Fan-Cabin Temperature PI Formulation

The cabin temperature PI loop was also analyzed independently. In order to do so, the compressor speed and EEV opening percent were maintained at constant nominal values. Then the disturbances were applied to the system and the gains were tuned heuristically. The chosen PI gains were P=15 and I=0.015. The results are displayed in Figure 3.4.

Outside of the first 100 seconds when the controller is initially working to bring the cabin temperature near the reference, the PI controller was able to maintain the cabin temperature within 0.33^{o}C of the reference temperature of 22.5 ^{o}C, with a maximum temperature of 22.54^{o}C and a minimum of 22.17^{o}C. Overall the root mean square error of the cabin temperature is 0.19. These results suggest that the PI adequately controls the cabin temperature.



(a) Actuator Command

(b) System Output

Figure 3.4: Evaporator and condenser simulation results

3.4 Complete PI Formulation

After each individual SISO loop was tuned, they were combined to control both the superheat and the cabin temperature simultaneously. The actuator commands and system outputs are shown in Figures 3.5-3.6.

Outside of the initialization time period, the PI controller was able to maintain the evaporator superheat within 0.22^{o}C of the reference temperature of 4.5 ^{o}C, with a maximum superheat of 4.62^{o}C and a minimum superheat of 4.28^{o}C. Overall the root mean square error of the evaporator superheat is 0.08. Additionally, disregarding the initial 100 seconds when the controller is initially working to bring the cabin temperature near the reference, the PI controller was able to maintain the cabin temperature within 0.19^{o}C of the reference temperature of 22.5 ^{o}C, with a maximum temperature of 22.64^{o}C and a minimum of 22.31^{o}C. Overall the root mean square error of the cabin temperature is 0.14. Interestingly, this cabin temperature root mean square error is slightly better than that of the SISO loop. This suggests that the EEV opening percent and the evaporator fan speed assisted each other in reaching their respective reference temperatures. Overall, this multiple SISO loop PI controlled system serves as a reasonable baseline to compare with the MPC in Chapter 4.

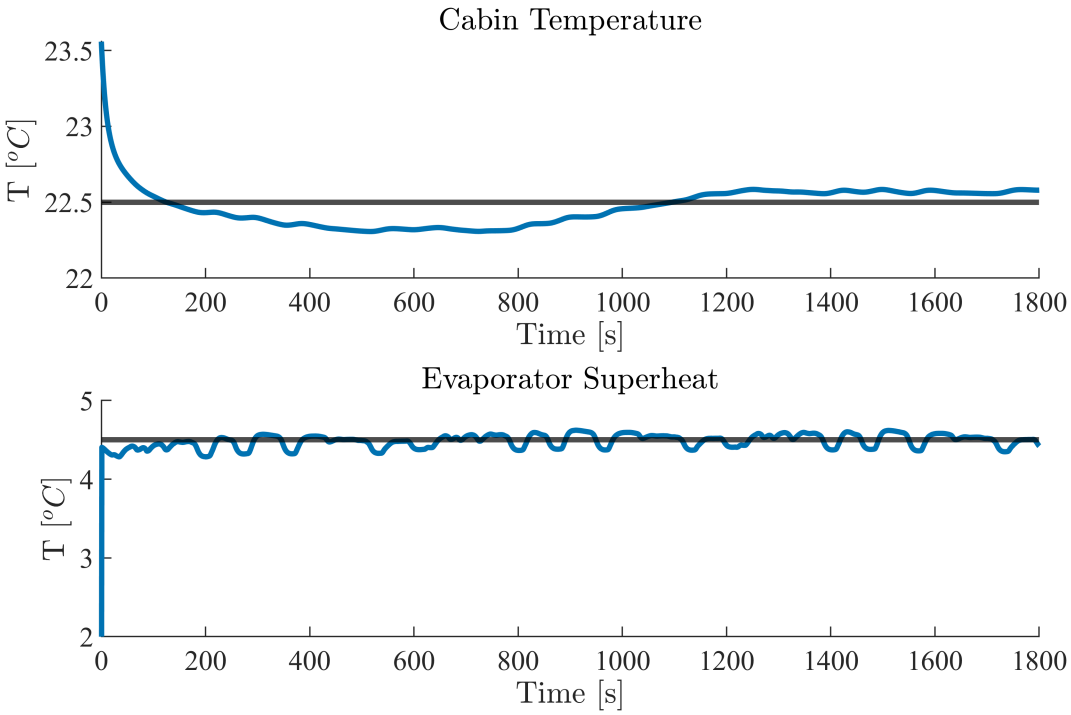


Figure 3.5: System Outputs

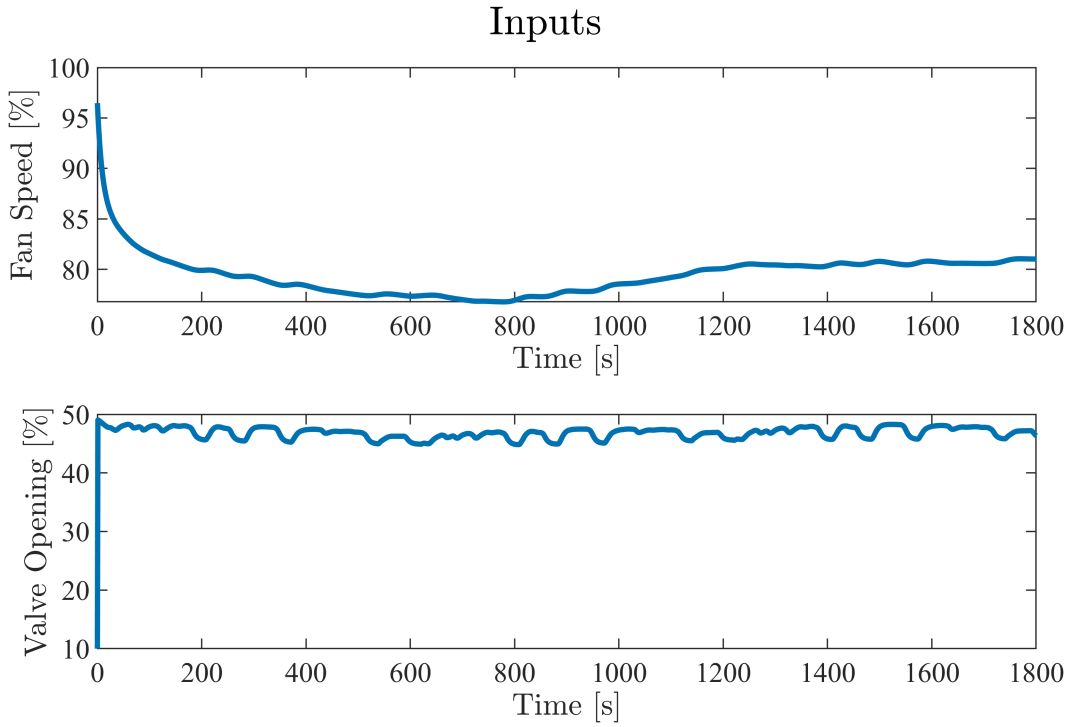


Figure 3.6: System Actuator Commands

Chapter 4

Model Predictive Controller

This chapter will provide details on the model predictive control (MPC) implemented in this work and compare its performance with the baseline controller. Section 4.1 introduces MPC with a literature review and general overview of the formulation. Section 4.2 outlines the linearization process used to create linear models that were implemented in the MPC. Section 4.3 provides details on the state estimation technique used to obtain states from the output measurements. Section 4.4 describes the details of the specific MPC applied in this work. Section 4.5 presents the results. Lastly, section 4.6 compares the MPC results with the baseline controller.

4.1 Model Predictive Control Background

4.1.1 Literature Review

MPC first gained popularity through its application in the petro-chemical industry in the late 1970s and early 1980s [46]. Since then, MPC has been implemented in a wide variety of applications and industries, such as agriculture [47], automotive [48], aerospace [49], power electronics [50], and HVAC [43]. In HVAC systems MPC is implemented to maintain a desired reference temperature while miniziming power consumption. When implemented on a conventional internal combustion engine vehicle or on a hybrid electric vehicle, this can results in savings in fuel consumption. There exist several instances of MPC for vehicle thermal management in the literature. For instance, [51] used MPC to heat a hybrid electric vehicle cabin by optimizing the workload between an electrical heater and the heater core. This technique saw a 3% increase in fuel savings compared to the baseline controller. Additionally, [52] implemented a nonlinear MPC (NMPC) to regulate the battery temperature in real-time. While one of the disadvantages of NMPC is that it can be challenging to implement in

real-time, [52] was able to implement the NMPC through the use of multiple cores and multiple processors. Linear MPC is chosen as the control method for this work, because it is computationally cheaper to implement than NMPC. This chapter will focus on developing and applying a linear MPC to optimally cool a cabin while respecting system constraints and minimizing power consumption.

4.1.2 General MPC Formulation

A MPC solves a finite-horizon optimal control problem. As shown in Figure 4.1, the controller is given future knowledge of reference and disturbance trajectories (denoted r and d in the figure, respectively), a dynamic model of the system, system constraints, and the current system states. In linear MPC a discrete linear model of the system is used to predict future states. The linear model is given in state space form by:

$$x_{k+1} = Ax_k + Bu_k \tag{4.1}$$

$$y_k = Cx_k + Du_k \tag{4.2}$$

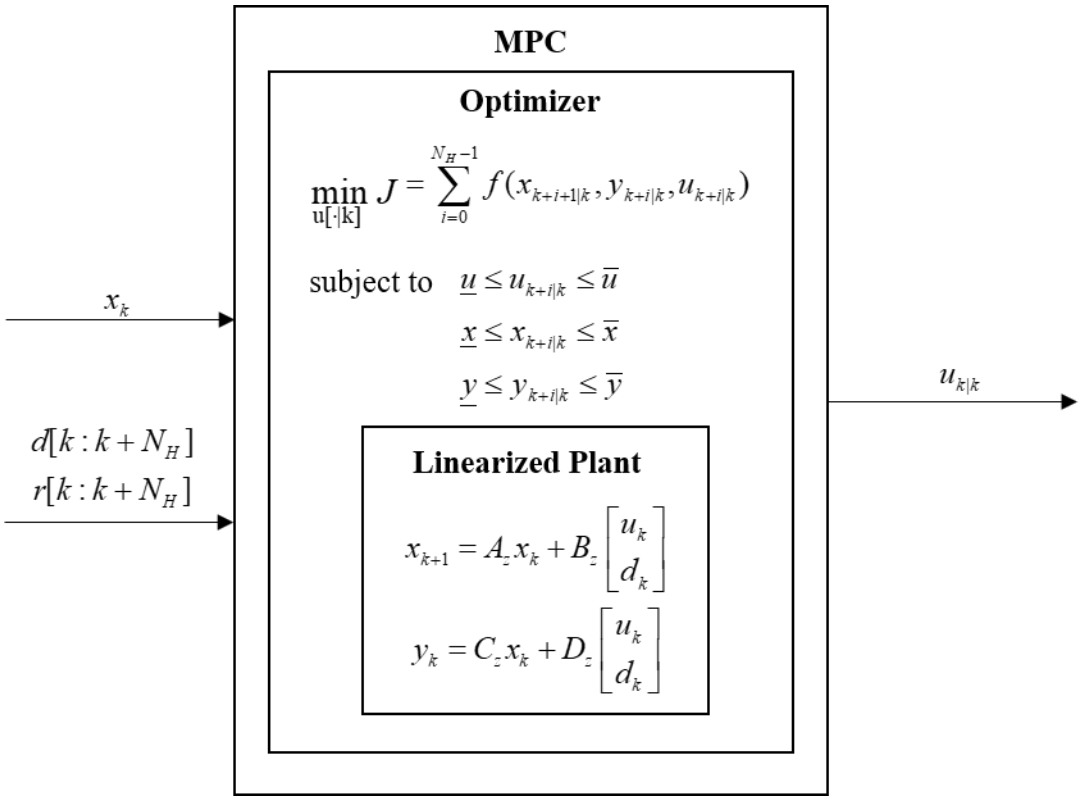


Figure 4.1: MPC inputs and outputs

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $u \in \mathbb{R}^p$. The control horizon, N_H, is the length of time that the controller is predicting in the future. While a larger control horizon can improve the MPC prediction, it comes at the cost of increased computational time. Thus, a suitable balance must be determined between the computational time and prediction accuracy in order to implement MPC in real-time. The length of the horizon is often dependent on the application as certain dynamic systems vary widely in modeling complexity and time constants.

At each time step the MPC iterates through the following steps:

1. Obtain the state at the current time step from either measurement data or predictions from an observer.
2. Optimize future states and inputs, given the current time step.
3. Send the current actuator command to the system.
4. Repeat Steps 1-3 on the following time step.

4.2 Model Linearization

4.2.1 VCS Component Model Linearization

The linearization of the VCS components in this work directly follows from [33]. Each component in the VCS was linearized about a steady state point near the middle of the operating range. Then each linear component was combined using the Redheffer Star product. This results in a linear VCS model with 15 states, 14 outputs and 6 inputs/disturbances (4.3). In order to obtain the discrete linear model for the MPC, the zero-order hold was applied to the continuous linear model.

$$x_{VCS} = \begin{bmatrix} \zeta_{1,e} \\ P_e \\ h_{2,e} \\ T_{w1,e} \\ T_{w2,e} \\ \bar{\gamma}_e \\ \zeta_{1,c} \\ \zeta_{2,c} \\ P_c \\ h_{1,c} \\ h_{3,c} \\ \bar{\gamma}_c \\ T_{w1,c} \\ T_{w2,c} \\ T_{w3,c} \end{bmatrix}, y_{VCS} = \begin{bmatrix} P_e \\ h_{e,r,o} \\ T_{e,r,o} \\ T_{SC} \\ T_{e,a,o} \\ \dot{Q}_e \\ \dot{m}_{EEV} \\ P_c \\ h_{c,r,o} \\ T_{c,r,o} \\ T_{SC} \\ T_{c,a,o} \\ \dot{Q}_c \\ \dot{m}_k \end{bmatrix}, u_{VCS} = \begin{bmatrix} u_{eev} \\ T_{e,a,i} \\ u_{fan} \\ \omega_k \\ T_{a,c,i} \\ \dot{m}_{a,c,i} \end{bmatrix}. \tag{4.3}$$

4.2.2 Cabin Model Linearization

The linear cabin model follows directly from [42], but is provided here for modeling completeness. While the linear VCS models were calculated analytically about a steady state operating point, the linearized cabin model was determined through state space system identification techniques. The standard black-box modeling technique was used, in which a random binary signal is applied to each input and the output response of the system is measured. From this input/output data, the Matlab System Identification Toolbox is used to determine the linear state-space model, in the form of (4.4), that minimizes the prediction error to obtain the maximum likelihood system.

$$\begin{aligned} x[k+1] &= A\Delta x[k] + B\Delta u[k] \\ y[k] &= C\Delta x[k] + D\Delta u[k] \end{aligned} \tag{4.4}$$

While the nonlinear cabin model has 22 dynamic states, the linear model was reduced to one state, the cabin air temperature, which was shown to be sufficient at capturing the dominant dynamics of the nonlinear system. The inputs considered for the linear cabin model are the evaporator fan speed, the air temperature at the evaporator outlet, and the ambient temperature. These three inputs are chosen, because they are considered measurable. All other inputs to the nonlinear cabin model, such as the solar irradiance, are treated as

unmeasurable disturbances.

The linear cabin model was found to be:

$$\begin{aligned} x_{cab} &= T_{cab} \\ y_{cab} &= T_{cab} \\ u_{cab} &= \begin{bmatrix} u_{fan} & T_{e,a,o} & T_{amb} \end{bmatrix}^T \end{aligned} \tag{4.5}$$

$$\begin{aligned} A_{cab} &= 0.9756, \\ B_{cab} &= \begin{bmatrix} -0.0024 & 0.0176 & 0.0031 \end{bmatrix}, \\ C_{cab} &= 1, \\ D_{cab} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{4.6}$$

4.2.3 Combined VCS-Cabin Linear Model

After creating the linear VCS and cabin models, they were combined. The evaporator air outlet temperature from the VCS was fed back as an input to the cabin model, as shown in Figure 4.2. This interconnection was done via the Redheffer Star Product to calculate new A_{sys}, B_{sys}, C_{sys}, and D_{sys} matrices. The complete VCS-Cabin system vectors are given by:

$$x_{sys} = \begin{bmatrix} T_{cab} \\ \zeta_{1,e} \\ P_e \\ h_{2,e} \\ T_{w1,e} \\ T_{w2,e} \\ \bar{\gamma}_e \\ \zeta_{1,c} \\ \zeta_{2,c} \\ P_c \\ h_{1,c} \\ h_{3,c} \\ \bar{\gamma}_c \\ T_{w1,c} \\ T_{w2,c} \\ T_{w3,c} \end{bmatrix}, y_{sys} = \begin{bmatrix} T_{cab} \\ P_e \\ h_{e,r,o} \\ T_{e,r,o} \\ T_{SC} \\ T_{e,a,o} \\ \dot{Q}_e \\ \dot{m}_{EEV} \\ P_c \\ h_{c,r,o} \\ T_{c,r,o} \\ T_{SC} \\ T_{c,a,o} \\ \dot{Q}_c \\ \dot{m}_k \end{bmatrix}, u_{sys} = \begin{bmatrix} u_{eev} \\ u_{fan} \\ \omega_k \\ T_{e,a,i} \\ T_{amb} \\ T_{a,c,i} \\ \dot{m}_{a,c,i} \end{bmatrix} \tag{4.7}$$

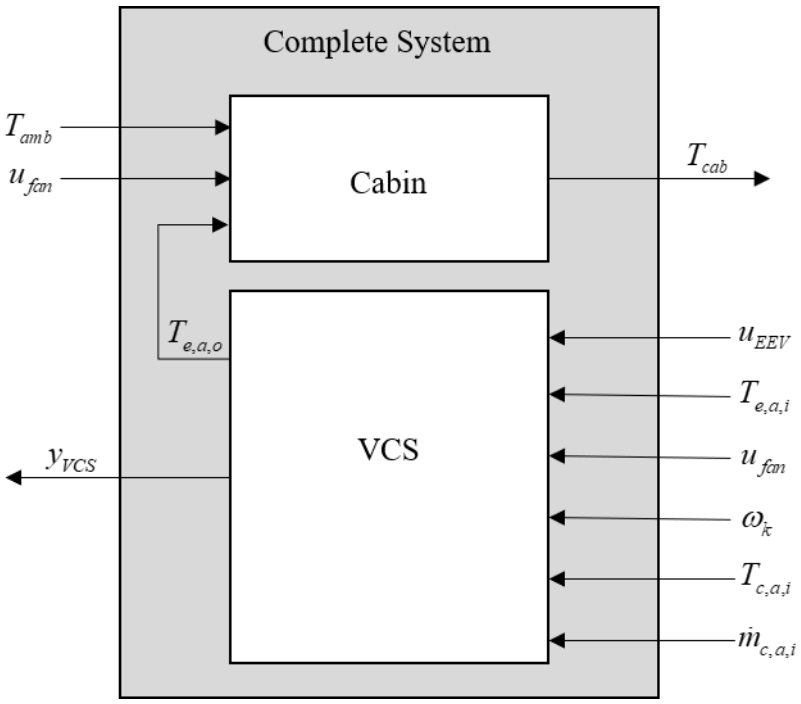


Figure 4.2: VCS and Cabin Combination

Note that the only difference between x_{sys} and x_{VCS} is that T_{cab} is inserted as the first element. Similarly, T_{cab} is inserted in y_{VCS} to create y_{sys}. The order of the elements in u was also rearranged such that the top three elements are treated the actuators in the system. This is done to keep the system in a more workable form for the controller. Additionally, T_{amb} was added as another input to the system.

The linear model for the combined system was compared with the nonlinear model to ensure the model would be sufficient for the controller and observer. In order to compare the two models, the system inputs were stepped up and down to obtain a dynamic response from the models. Figure 4.3 shows the sequence of inputs used to compare linear and nonlinear models. The resulting outputs of both models are compared in Figure 4.4.

Overall the linear model tracked well with the nonlinear model. The linear model stepped up and down with similar magnitudes to the nonlinear model. The most notable difference between the two models is that there is a slight steady state offset between the linear and nonlinear models. However, this offset is small enough that the linear model should be sufficiently accurate to be used in the MPC and Kalman filter

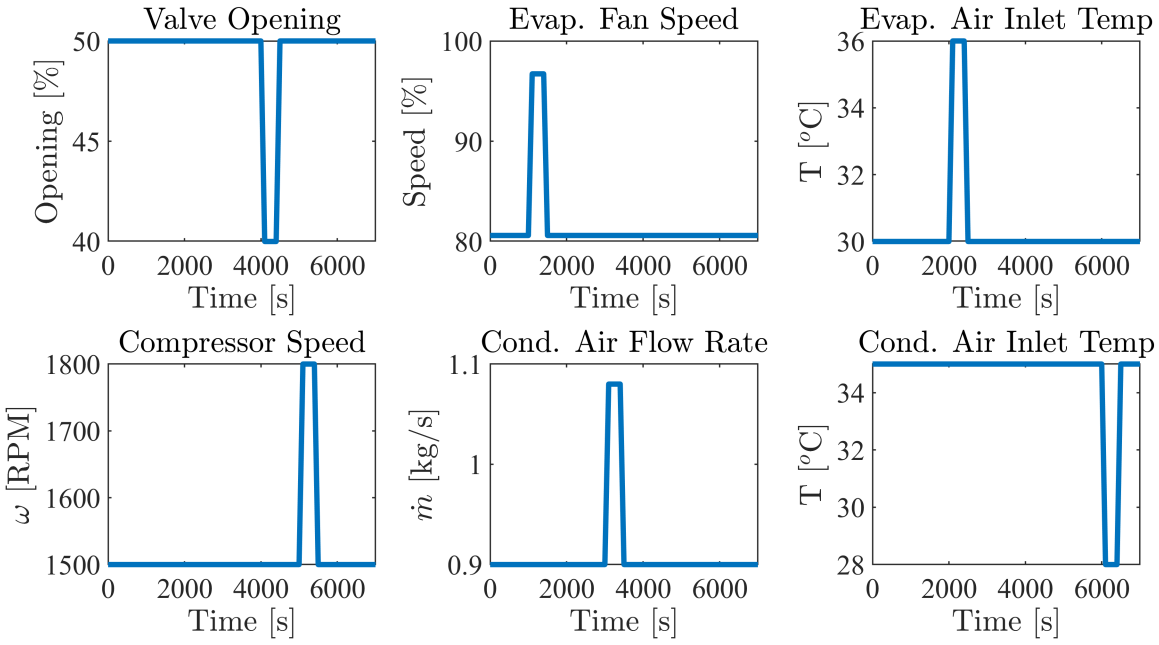


Figure 4.3: System Input

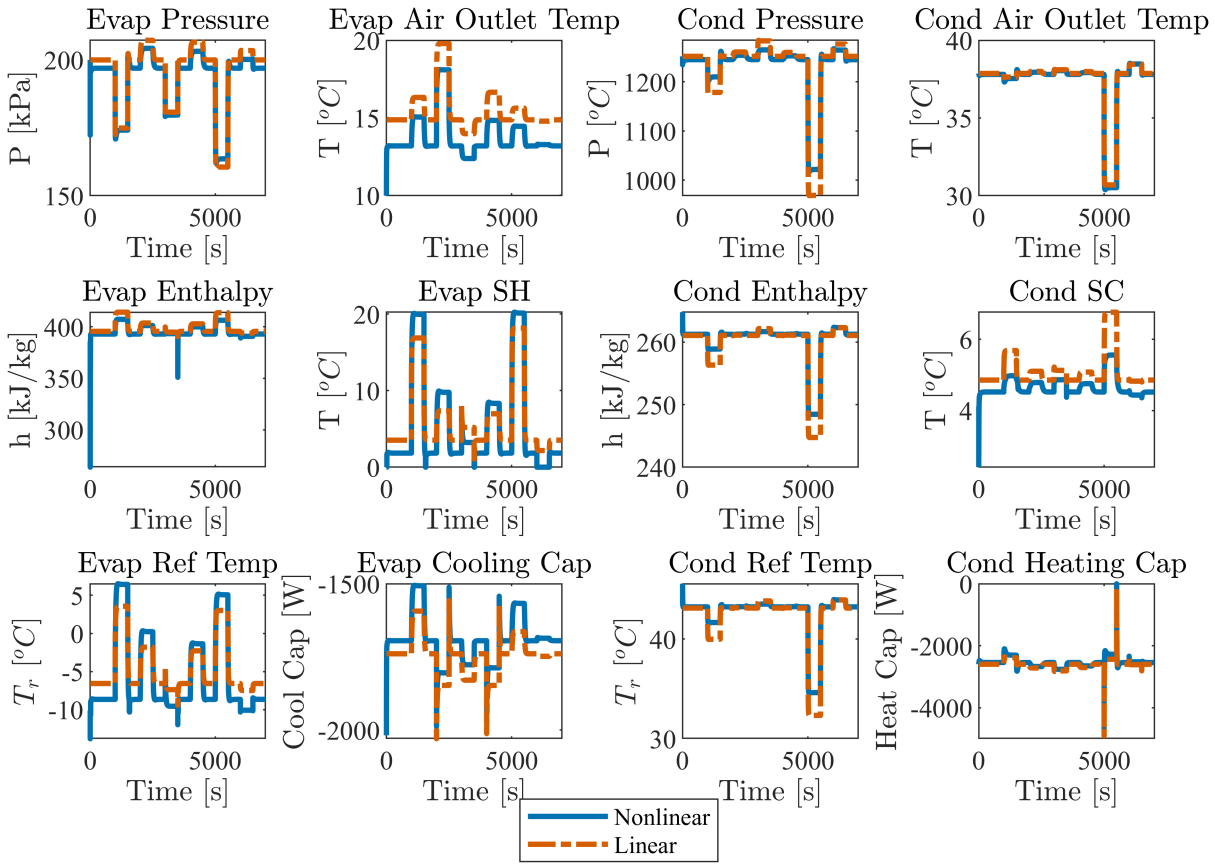


Figure 4.4: System Outputs

4.3 State Estimation

Since several of the states in the VCS-Cabin system are not measurable (e.g. evaporator two phase length) and MPC relies on the current state values, an observer is designed to estimate the states. The measured output vector is

$$y_{meas} = \begin{bmatrix} T_{cab} & P_e & T_{e,r,o} & T_{e,a,o} & P_c & T_{c,r,o} & T_{c,a,o} \end{bmatrix}^T. \tag{4.8}$$

All other outputs are considered not measurable. Additionally, the measurable input vector is

$$u_{meas} = \begin{bmatrix} u_{EEV} & u_{fan} & \omega_k & T_{e,a,i} & T_{c,a,i} & \dot{m}_{a,c} \end{bmatrix}^T. \tag{4.9}$$

A Kalman filter is used to estimate the states in this thesis [53]. The discrete Kalman filter has a prediction phase and an update phase to calculate the state prediction, as shown in Figure 4.5. In the Figure 4.5, $\hat{x}$ represents the state estimate, u is the system input, P is the covariance matrix, z is the measured value of the output, K is the Kalman gain, k is the time step, $\tilde{y}$ is the residual, Q is the covariance of the process noise, and R is the covariance of the observation noise. Often times Q and R are not known, so they can be treated as tuning parameters for the Kalman gain.

During the prediction step, the linearized model, previous prediction, and measured inputs are used to calculate the state and covariance matrix estimations. Then in the update step the output measurements are used to update the state and covariance estimates resulting in the final state and covariance predictions.

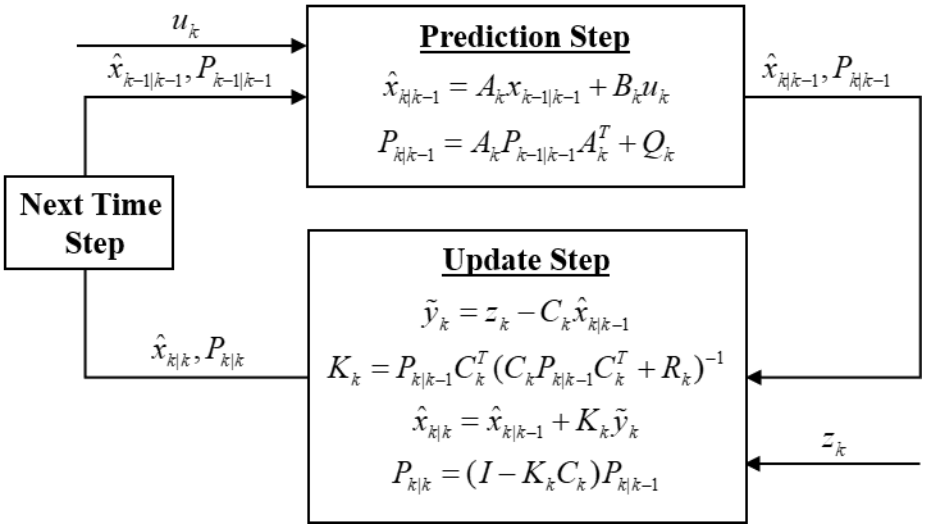


Figure 4.5: Kalman Filter Flow Chart

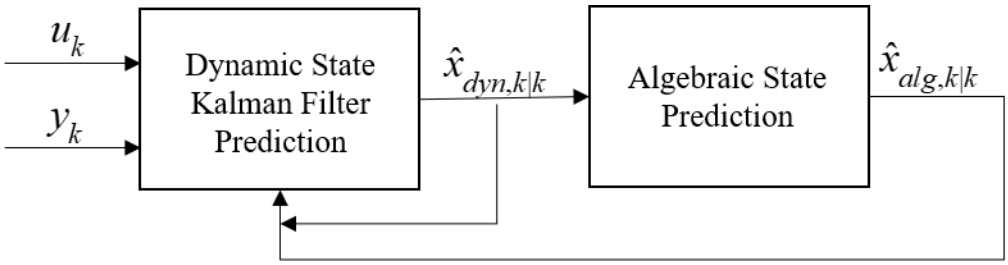


Figure 4.6: Algebraic and Dynamic State Predictions

One noteworthy point about the Kalman filter implementation in this work is that the faster dynamic states are treated as algebraic when solving for the estimates. The fast states in this work are considered to be ζ_{1e}, ζ_{1c}, ζ_{2c}, and γ_c, representing the moving boundary fluid lengths and mean void fractions in the evaporator and condenser. These were treated as the fast states because [54] determined that the faster states in a VCS are those associated with the refrigerant energy, such as pressure or moving boundary fluid lengths. However, most of the states depend on the evaporator and condenser pressure, so the mean void fraction was treated as the fast state in place of the pressure.

Treating some of the states as algebraic helped improve the accuracy and efficiency of the Kalman filter. To solve for the algebraic states, first the Kalman filter predicted the dynamic states. Then the linear model approximations and dynamic states are used to predict the algebraic states by solving

$$\dot{x} = A_a x_{k,a} + B_a u_k = 0. \tag{4.10}$$

where the subscript a denotes algebraic. This process is shown in Figure 4.6.

The Kalman filter was tuned and applied to the VCS-Cabin system operating in the middle of the operating range. The results are presented in Figure 4.7. The steps in the results were created by stepping the disturbances. Overall, the Kalman filter produced accurate predictions for most of the states. Only the evaporator wall 2 temperature, condenser two phase length, and condenser mean void fraction deviated noticeably from the actual plant response. However, since these state predictions normally stepped in the correct direction and were close in magnitude, the Kalman filter was deemed sufficient to be implemented with the MPC.

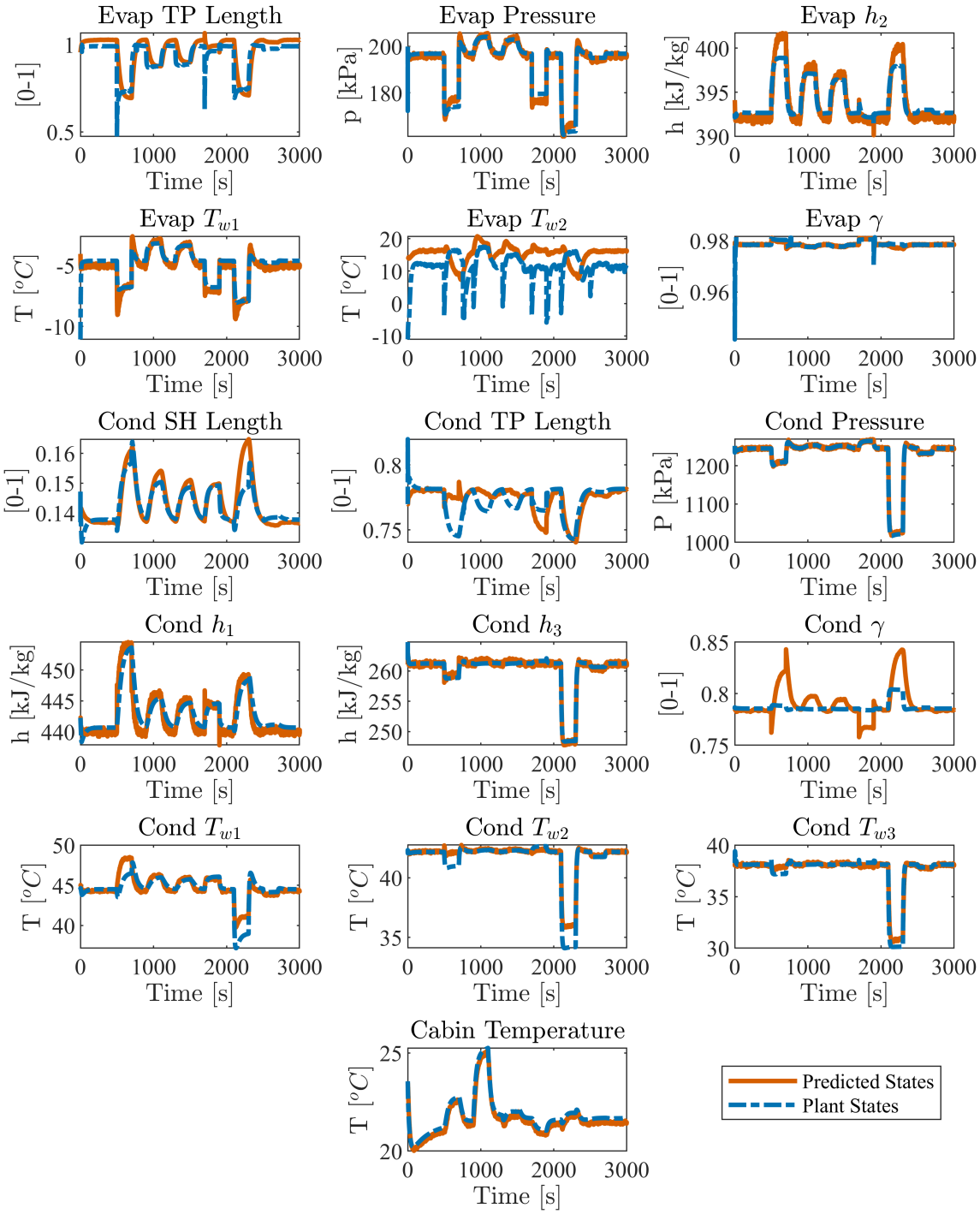


Figure 4.7: Kalman filter predictions

4.4 MPC Formulation

As with the PI control, two of the main goals of the MPC are to maintain the cabin temperature within a reference range and to keep the evaporator superheat within a desired

operating range. Achieving these goals ensures that the cabin temperature demands are met while operating the vehicle in an efficient manner. The MPC is also given the objective to minimize the actuator inputs. This encourages the system to put less energy towards the VCS, improving the overall efficiency of the EV. Meanwhile, the MPC must capture certain constraints within the system, such as respecting actuator bounds and following the system dynamics. The MPC optimization problem was formulated using YALMIP [55] and the formulation is given by

$$J_k = \Lambda_u ||u_k||^2 + \Lambda_s ||s_k||^2 + \Lambda_b ||b_k||^2 + \Lambda_{du} ||u_k - u_{k-1}||^2 \tag{4.11a}$$

$$\min_{u,s,b} J = \sum_{k=1}^{N_H} J_k \tag{4.11b}$$

subject to
$\forall k \in [1 : N_H]$

$$x_{k+1} = A_z x_k + B_z \begin{bmatrix} u_k \\ d_k \end{bmatrix}, \tag{4.11c}$$

$$y_k = C_z x_k + D_z \begin{bmatrix} u_k \\ d_k \end{bmatrix}, \tag{4.11d}$$

$$\underline{x} - s_{k+1} \leq x_{k+1} \leq \overline{x} + s_{k+1}, \tag{4.11e}$$

$$s_{k+1} \geq 0, \tag{4.11f}$$

$$\underline{y} - b_{k+1} \leq y_{k+1} \leq \overline{y} + b_{k+1}, \tag{4.11g}$$

$$b_{k+1} \geq 0, \tag{4.11h}$$

$$\underline{u} \leq u_k \leq \overline{u}. \tag{4.11i}$$

Each Λ is a weight variable for the different objectives. These weights are tuned to achieve the desired system behavior. The first term in (4.11a) is used to minimize the actuator commands for the compressor and evaporator fan speed. This objective was implemented to reduce the power consumption of the VCS. The EEV opening command is neglected in this term because it is assumed that it consumes negligible energy compared to the other actuator inputs. The second and third terms are use to minimize the slack variables, s and b. Slack variables allow the system to violate certain constraints, but with a cost. Slack variables allow the MPC to solve more quickly and easily. For this implementation slack variables are applied to the evaporator superheat and cabin temperature. The last term in (4.11) is used to minimize the rate of change of the actuator inputs. This is used for all three actuators, as it is undesirable for any of them to oscillate unnecessarily. Oscillations can cause extra wear

and tear on the system and the dynamics can be more challenging to predict.

As for the constraints, (4.11c)-(4.11d) ensure that the MPC is making predictions based on the system dynamics. (4.11e) aims to maintain the cabin temperature within the desired range. However, the slack variable s allows the system to violate that constraint. Similarly, (4.11g) aims to maintain the evaporator superheat within the desired operating range. Again, the slack variable b allows the system to violate that constraint. (4.11f)-(4.11h) maintain that the slack variables must be non-negative. Lastly, (4.11i) ensures that the MPC does not violate actuator constraints inherent in the system. For instance, the EEV opening percent has a lower bound of 0 and an upper bound of 100.

4.5 Simulation Results

The observer, plant and MPC are combined in a feedback control loop as shown in Figure 4.8. The disturbance profile from Chapter 3 was applied in this chapter to ensure fair comparison between the baseline controller and the MPC. The results of the controlled system are shown in Figures 4.9 - 4.10.

The controller met the two main constraints of keeping the cabin temperature and evaporator superheat within their respective bounds. These results show that while some of the state predictions from the Kalman filter deviate from the true system (Figure 4.7), it is still sufficient in providing reasonable estimates for the MPC to utilize.

The controller was given a prediction horizon of 3 minutes and a discrete step size of 1 second. The MPC calculation time was measured to ensure that the MPC could run in real time. On average the MPC took 0.41 seconds to calculate the optimal inputs and took a maximum of 0.57 seconds to run. These times are well below the 1 second step size, suggesting that the MPC prediction could be extended, if needed. However, the results suggest that the 3 minute horizon was sufficient at keeping the outputs within the specified constraints.

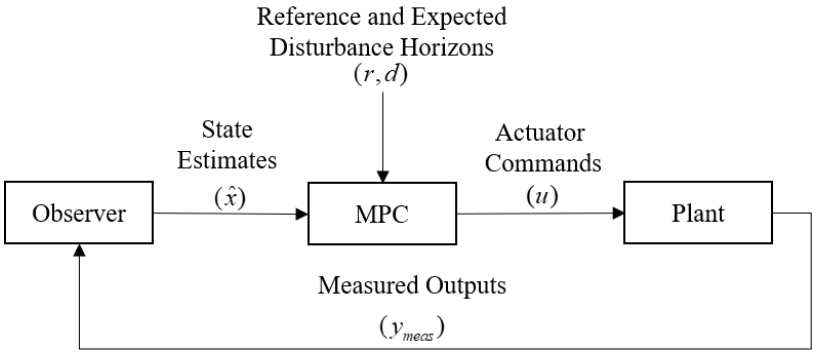


Figure 4.8: Feedback control loop

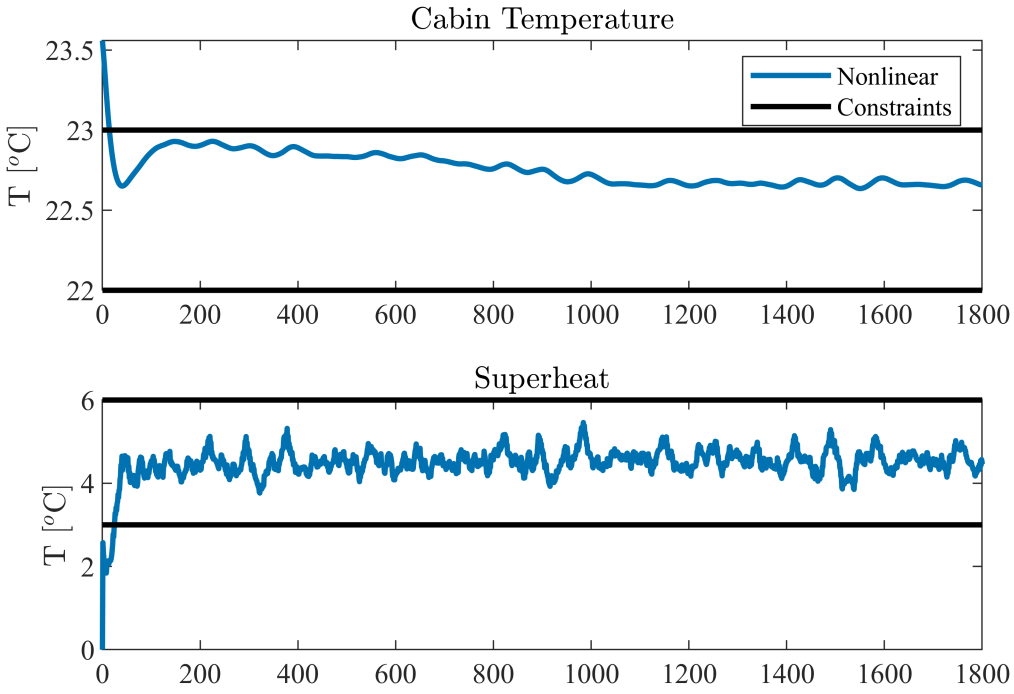


Figure 4.9: MPC output results

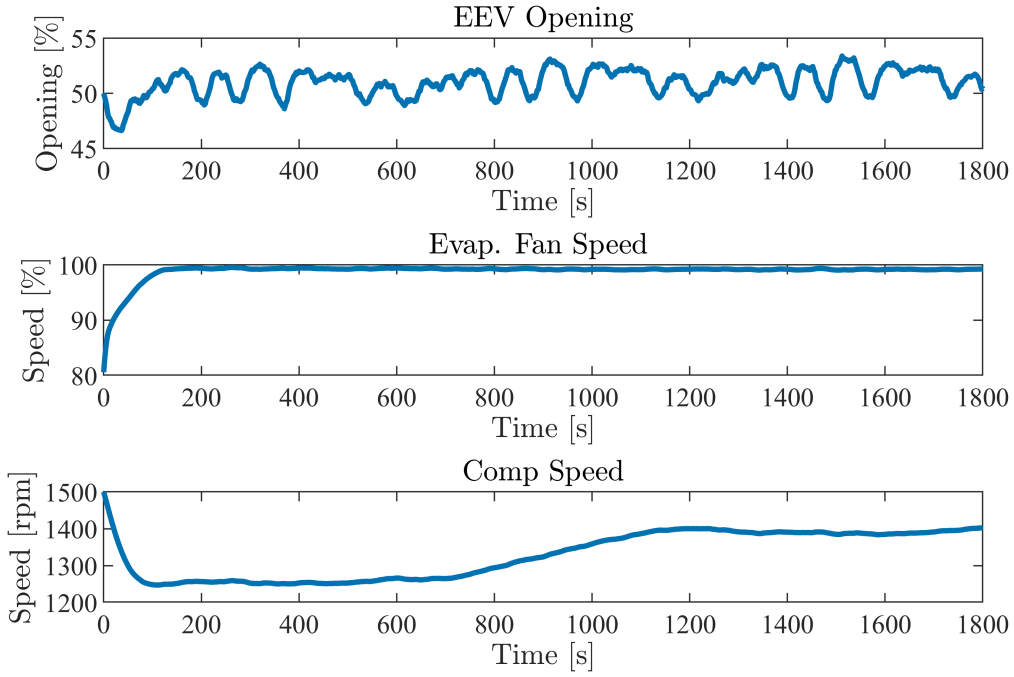


Figure 4.10: MPC actuator commands

4.6 Comparison with Baseline Controller

This section will compare the baseline control results with the MPC. The outputs and inputs from the different control systems are presented in Figures 4.11 - 4.12.

As can be seen, both controllers are able to keep the cabin temperature and superheat from violating constraints. The PI and MPC provided similar evaporator superheat responses. The only minor difference was the sharp increase in the PI EEV opening and subsequent increase in superheat. This allowed the superheat to be within the bounds more quickly than the MPC. The MPC likely did not have this response because of the constraint on the input rate of change. However, after this initial difference in EEV opening response, the MPC and PI used similar EEV opening commands, with the PI operating slightly lower than the MPC.

On the other hand, the MPC and PI resulted in noticeably different cabin temperature outputs. This is due, in part, to the fact that the MPC could change all three actuators to maintain the cabin temperature, while the PI control could only change the evaporator fan speed. The MPC allowed the compressor speed to decrease at the expense of the evaporator fan speed increasing. Meanwhile the PI initially had a higher fan speed than the MPC, but it lowered as the compressor speed was operating at a higher speed than the MPC controlled compressor. Since the evaporator fan speed is the only actuator used to control the cabin temperature in the PI case, the PI system has the potential to struggle more if there is a noticeable change in ambient temperature. When the ambient temperature increased from 700 - 1100 seconds, the MPC was able to take this into account and increase the compressor speed. This allowed the cabin temperature to actually decrease, with an increase in compressor

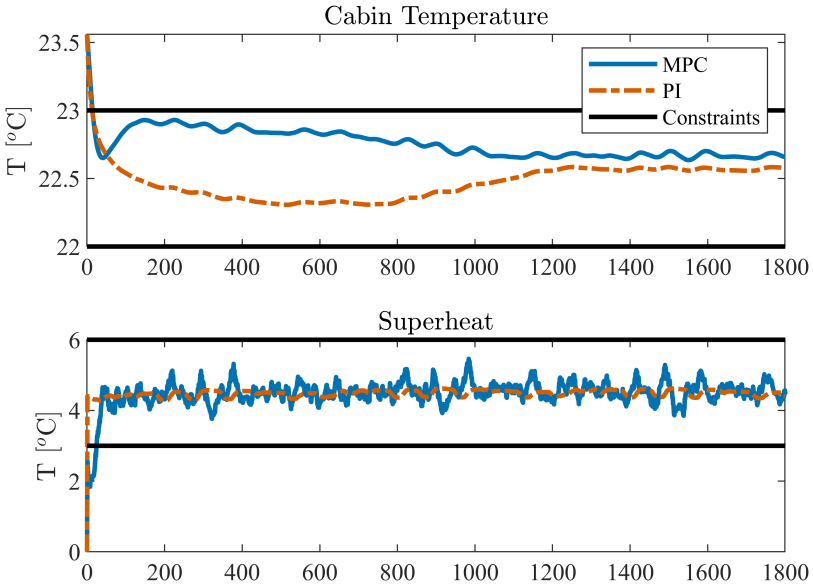


Figure 4.11: MPC vs PI system outputs

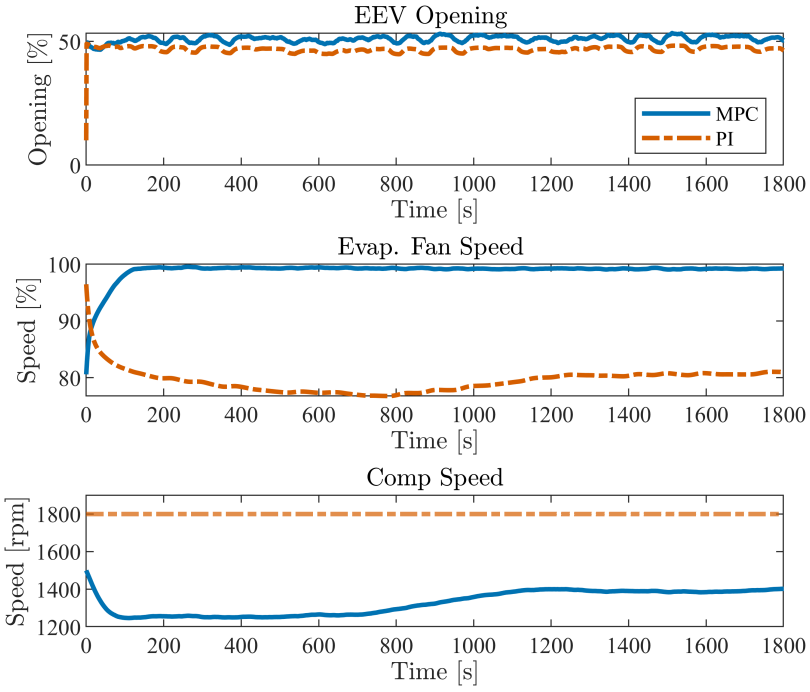


Figure 4.12: MPC vs PI actuator commands

speed. However, the PI results increased as the system was operating near it's limit. While this problem is dependent on the conditions, this highlights the potential for the PI to have problems if there are drastic fluctuations in the disturbances.

In addition to noting that the output constraints are met, a power consumption analysis was performed on the PI and MPC controlled systems as another mode of comparison. The power consumption was calculated every 1 second using the methods outlined in Section 2.8. The power consumption of each case is displayed in Figure 4.13. Initially the MPC has a lower power consumption as the compressor is operating at lower speeds. Then once the ambient temperature increases, the MPC compressor speed increases and causes the MPC to consume more power than the PI case.

It should be noted that these power consumption results are dependent on several different factors. As the compressor power calculation (2.66) is largely dependent on the compressor efficiency, the overall power consumption results vary widely depending on the assumed efficiency. The results provided in Figure 4.13 are calculated with an assumed 80% compressor efficiency. The results for other compressor efficiencies are presented in Figure 4.14. In these results the PI can occasionally outperform the MPC in terms of power consumption, as the ratio of energy going to the compressor and evaporator fan changes with the change in efficiency. Additionally, these results change depending on the assumed constant compressor speed in the PI case. Lastly, these results can change due to the tuning of the MPC and PI

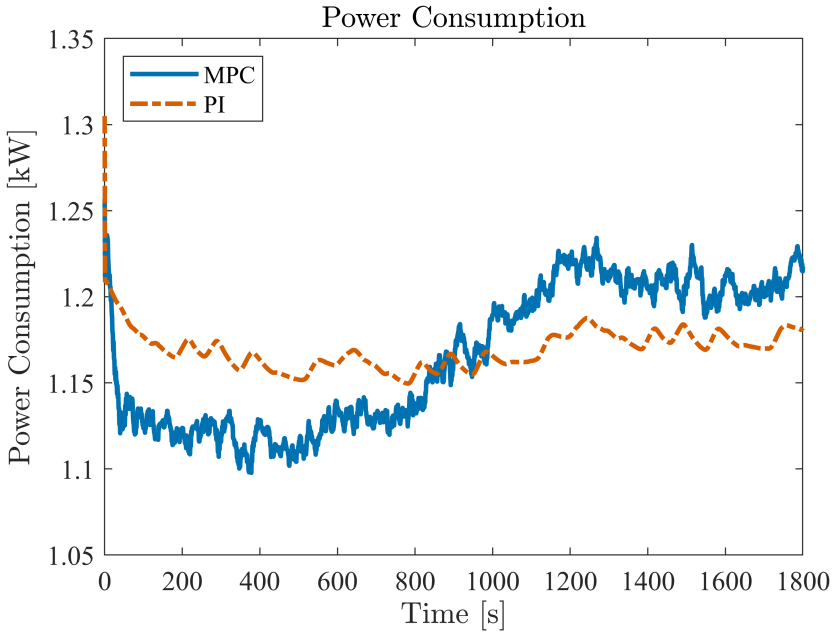


Figure 4.13: MPC vs PI power consumption

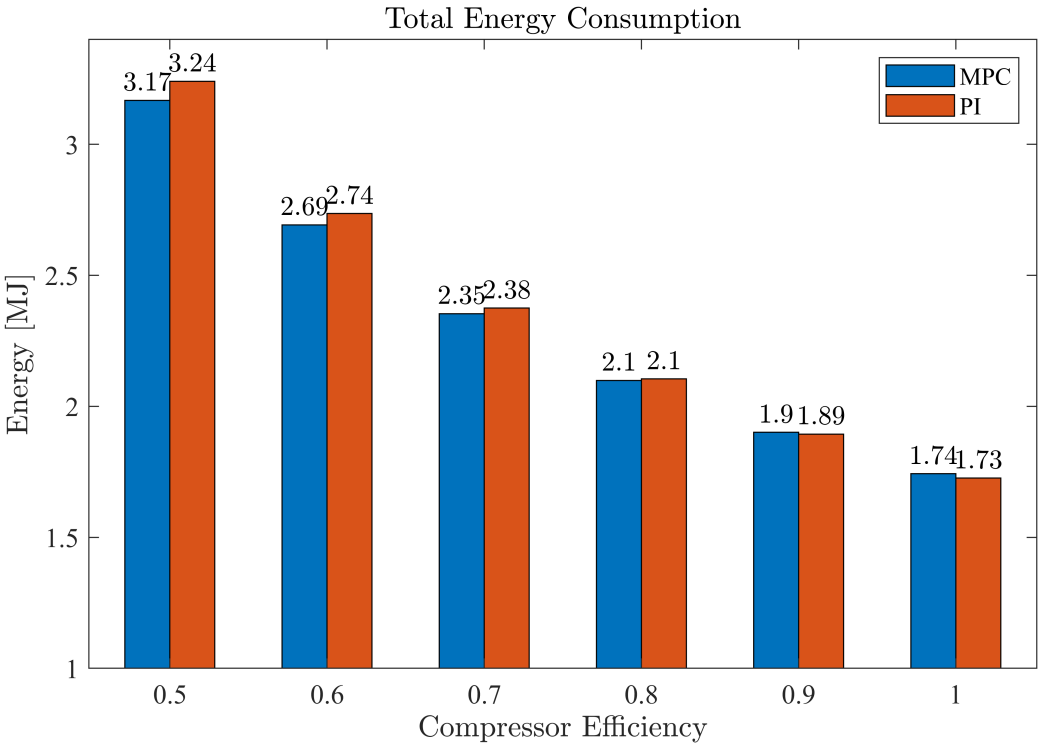


Figure 4.14: MPC vs PI power trends with compressor efficiency

gains. Overall, the power consumption from each system is comparable and depends on the system conditions and controller tuning.

Overall, both the PI and MPC respected the cabin temperature and evaporator superheat

constraints. The main benefit of the MPC is that it is able to adjust all three actuators, giving it a greater capacity to handle a wider range of operating conditions than the PI.

Chapter 5

Conclusion

5.1 Research Contributions

The main contribution in this work is the creation of a multi-state graph-based modeling framework which is an extension of the previously developed single-state graph-based modeling framework. This extended framework allows us to model dynamic systems and components with strong dynamic coupling in a modular and computationally efficient manner. A specific case of a two-state modeling formulation was applied to two-phase heat exchanger modeling for a VCS. This heat exchanger model was based on first principles and includes complex physical phenomena such as pressure drop throughout the heat exchanger due to friction losses. The graph-based model was verified against other models in the literature and the results suggest that it operates in a more favorable trade-off between simulation speed and model accuracy. Additionally, the modularity of the heat exchanger graph-based model allows it to more easily interact with other energy systems, such as an energy storage element [56] or building zone elements [57]. While the multi-state graph-based modeling framework was applied specifically to two-phase heat exchangers in the TMS of an EV, it could be implemented to other applications where similar coupling has been noted.

In addition to extending the graph-modeling framework, MPC and PI controls were explored as two different control options for a TMS onboard an EV. Both the PI and MPC showed they are capable of meeting the different constraints and requirements necessary for the TMS on an EV to operate safely.

5.2 Future Work

Next steps for this work include validating the models and control with experimental results. While the multi-state graph-based heat exchanger model was verified against other models in the literature, it would be beneficial to have experimental results validating the model's accuracy. In addition to validating the heat exchanger model, experimental validation of the complete VCS-cabin system would further show that the multi-state graph-based models can provide accurate results when combined with single-state graph-based models and other non-graph-based state-space models. Lastly, experimental validation of the MPC and PI controls would support the claims made in this work.

In addition to experimental validation, future work could include adding more components to the model. This work focused on the TMS on the cabin; however, in an EV the battery temperature must be maintained within a certain range to achieve optimal and safe performance. It would be interesting to include the battery as an extra objective for the controller. Once the battery is included, there are several different topologies that could be tested as different cooling loops can be configured. For instance, different configurations such as adding a separate cooling loop for the battery, a liquid-cooled battery loop, an air-cooled battery loop, or using one large cooling loop for both the cabin and the battery could be explored.

Lastly, additional future work could include optimization of the controlled system. Often times the system components are optimized first and then controlled; however, the results in Section 4.6 show us how different controllers can have different affects on the system dynamics. It would be interesting to formulate a control codesign problem in which the control gains and the components sizing are optimized simultaneously.

References

[1] U. E. P. Agency. "Sources of greenhouse gas enussuibs." (2022), [Online]. Available: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.

[2] Z. Li, A. Khajepour, and J. Song, "A comprehensive review of the key technologies for pure electric vehicles," *Energy*, vol. 182, pp. 824–839, 2019, ISSN: 03605442. DOI: 10.1016/j.energy.2019.06.077. [Online]. Available: https://doi.org/10.1016/j.energy.2019.06.077.

[3] M. A. Roscher, W. Leidholdt, and J. Trepte, "High efficiency energy management in BEV applications," *International Journal of Electrical Power and Energy Systems*, vol. 37, no. 1, pp. 126–130, 2012, ISSN: 01420615. DOI: 10.1016/j.ijepes.2011.10.022. [Online]. Available: http://dx.doi.org/10.1016/j.ijepes.2011.10.022.

[4] T. D. Harrison, "Air conditioning system utilizing vehicle waste energy," *SAE International*, 2009, ISSN: 26883627. DOI: 10.4271/2009-01-0543.

[5] J. Rugh and R. Farrington, "Vehicle ancillary load reduction project close-out report," Tech. Rep., 2008.

[6] K. Umezu and H. Noyama, "Air-conditioning system for electric vehicles (i-MiEV)," in *SAE Automotive Refrigerant & System Efficiency Symposium*, 2010.

[7] M. F. Sukri, M. N. Musa, M. Y. Senawi, and H. Nasution, "Achieving a better energy-efficient automotive air-conditioning system: a review of potential technologies and strategies for vapor compression refrigeration cycle," *Energy Efficiency*, vol. 8, no. 6, pp. 1201–1229, 2015, ISSN: 15706478. DOI: 10.1007/s12053-015-9389-4.

[8] S. K. Park, H. S. Kim, H. N. Ahn, and H. S. Park, "Study on the reduction of fuel consumption in the A/C system, used variable displacement swash-plate compressor and the performance improvement by field test," *SAE International*, 2006, ISSN: 26883627. DOI: 10.4271/2006-01-0164.

[9] M. Mitsui, "Improvement of Refrigerant Flow Control method in Automotive Air Conditioners," *SAE Technical Papers*, 1987.

[10] "A new method for controlling refrigerant flow in automobile air conditioning," *Applied Thermal Engineering*, vol. 24, no. 7, pp. 1073–1085, 2004, ISSN: 13594311. DOI: 10.1016/j.applthermaleng.2003.11.005.

[11] H. Qiao and C. R. Laughman, "Comparison of approximate momentum equations in dynamic models of vapor compression systems," *International Heat Transfer Conference*, pp. 2043–2050, 2018. DOI: 10.1615/ihtc16.cms.023630.

[12] J. P. Koeln, M. A. Williams, H. C. Pangborn, and A. G. Alleyne, "Experimental Validation of Graph-Based Modeling for Thermal Fluid Power Flow Systems," *ASME 2016 Dynamic Systems and Controls Conference*, pp. 1–10, 2016.

[13] M. A. Williams, J. P. Koeln, and A. G. Alleyne, "Hierarchical control of multi-domain power flow in mobile systems - Part I: Framework Development and Demonstration," *ASME 2015 Dynamic Systems and Control Conference, DSCC 2015*, pp. 1–10, 2015. DOI: 10.1115/DSCC2015-9904.

[14] K. M. Russell, C. T. Aksland, and A. G. Alleyne, "Graph-Based Dynamic Modeling of Two-Phase Heat Exchangers in Vapor Compression Systems," *International Journal of Refrigeration*, vol. 137, pp. 244–256, 2022. DOI: 10.1016/j.ijrefrig.2022.02.018.

[15] B. P. Rasmussen and B. Shenoy, "Dynamic modeling for vapor compression systems-Part II: Simulation tutorial," *HVAC and R Research*, vol. 18, pp. 956–973, 2012. DOI: 10.1080/10789669.2011.582917.

[16] M. W. Browne and P. K. Bansal, "Transient simulation of vapour-compression packaged liquid chillers," *International Journal of Refrigeration*, vol. 25, pp. 597–610, 2002. DOI: 10.1016/S0140-7007(01)00060-3.

[17] Q. Qi and S. Deng, "Multivariable control-oriented modeling of a direct expansion (DX) air conditioning (A/C) system," *International Journal of Refrigeration*, vol. 31, no. 5, pp. 841–849, 2008. DOI: 10.1016/j.ijrefrig.2007.10.009.

[18] G. L. Wedekind, B. L. Bhatt, and B. T. Beck, "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. DOI: 10.1016/0301-9322(78)90029-0.

[19] T. L. McKinley and A. G. Alleyne, "An advanced nonlinear switched heat exchanger model for vapor compression cycles using the moving-boundary method," *International Journal of Refrigeration*, vol. 31, pp. 1253–1264, 2008. DOI: 10.1016/j.ijrefrig.2008.01.012.

[20] S. Bendapudi, J. E. Braun, and E. A. Groll, "A comparison of moving-boundary and finite-volume formulations for transients in centrifugal chillers," *International Journal of Refrigeration*, vol. 31, pp. 1437–1452, 2008. DOI: 10.1016/j.ijrefrig.2008.03.006.

[21] W. J. Zhang and C. L. Zhang, "A generalized moving-boundary model for transient simulation of dry-expansion evaporators under larger disturbances," *International Journal of Refrigeration*, vol. 29, pp. 1119–1127, 2006. DOI: 10.1016/j.ijrefrig.2006.03.002.

[22] H. Qiao, C. R. Laughman, V. Aute, and R. Radermacher, "An advanced switching moving boundary heat exchanger model with pressure drop," *International Journal of Refrigeration*, vol. 65, pp. 154–171, 2016. DOI: 10.1016/j.ijrefrig.2016.01.026.

[23] L. J. Yebra, M. Berenguel, and S. Dormido, "Extended moving boundary model for two-phase flows," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 16, pp. 368–373, 2005. DOI: 10.3182/20050703-6-cz-1902.00062.

[24] J. W. MacArthur, "Transient heat pump behaviour: a theoretical investigation," *International Journal of Refrigeration*, vol. 7, pp. 123–132, 1984.

[25] S. Bendapudi, J. E. Braun, and E. A. Groll, "Dynamic model of a centrifugal chiller system - Model development, numerical study, and validation," *ASHRAE Transactions*, vol. 111 PART 1, pp. 132–148, 2005.

[26] H. Tummescheit, J. Eborn, and F. J. Wagner, "Development of a Modelica Base Library for Modeling of Thermo-Hydraulic Systems," in *Modelica Workshop Proceedings*, 2000, pp. 41–51.

[27] W. J. Zhang, C. L. Zhang, and G. L. Ding, "On three forms of momentum equation in transient modeling of residential refrigeration systems," *International Journal of Refrigeration*, vol. 32, pp. 938–944, 2009. DOI: 10.1016/j.ijrefrig.2008.11.002.

[28] B. P. Rasmussen, "Dynamic Modeling and Advanced Control of Air Conditioning and Refrigeration Systems," Ph.D. Dissertation, Univ. Illinois Urbana-Champaign, 2005.

[29] J. M. Fasl, "Modeling and Control of Hybrid Vapor Compression Cycles," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2013.

[30] V. Gnielinski, "New equations for heat and mass transfer in turbulent pipe and channel flow," *Int. Chem. Eng.*, vol. 16, no. 2, pp. 359–368, 1976.

[31] M. K. Dobson and J. C. Chato, "Condensation in smooth horizontal tubes," *Journal of Heat Transfer*, vol. 120, pp. 193–213, 1998, ISSN: 15288943. DOI: 10.1115/1.2830043.

[32] J. P. Wattelet, J. C. Chato, B. R. Christoffersen, and J. A. Gaibel, "Heat Transfer Flow Regimes of Refrigerants in a Horizontal-Tube Evaporator," Tech. Rep., 1994.

[33] H. Pangborn, "Dynamic, Modeling, Validation, and Control for Vapor Compression Systems," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2015.

[34] E. Lemmon, M. O. McLinden, and M. L. Huber, "NIST reference fluid thermodynamic and transport properties-REFPROP," *NIST Standard Reference Database*, vol. 23, 2002.

[35] B. R. Munson, T. H. (H. Okiishi, W. W. Huebsch, and A. P. Rothmayer, *Fundamentals of Fluid Mechanics*, 7th edition. Hoboken, NJ: John Wiley & Sons, Inc., 2013, ISBN: 9781118399712.

[36] C. T. Aksland, J. P. Koeln, and A. G. Alleyne, "A graph-based approach for dynamic compressor modeling in vapor compression systems," in *ASME 2017 Dynamic Systems and Control Conference*, 2017, pp. 1–8. DOI: 10.1115/DSCC2017-5238.

[37] H. Pangborn, A. G. Alleyne, and N. Wu, "A comparison between finite volume and switched moving boundary approaches for dynamic vapor compression system modeling," *International Journal of Refrigeration*, vol. 53, pp. 101–114, 2015, ISSN: 01407007. DOI: 10.1016/j.ijrefrig.2015.01.009. [Online]. Available: http://dx.doi.org/10.1016/j.ijrefrig.2015.01.009.

[38] A. Desideri, B. Dechesne, J. Wronski, *et al.*, "Comparison of moving boundary and finite-volume heat exchanger models in the Modelica language," *Energies*, vol. 9, no. 5, pp. 1–18, 2016, ISSN: 19961073. DOI: 10.3390/en9050339.

[39] J. Bonilla, L. J. Yebra, and S. Dormido, "Chattering in dynamic mathematical two-phase flow models," *Applied Mathematical Modelling*, vol. 36, no. 5, pp. 2067–2081, 2012. DOI: 10.1016/j.apm.2011.08.013. [Online]. Available: http://dx.doi.org/10.1016/j.apm.2011.08.013.

[40] A. Valade, P. Acco, P. Grabolosa, and J. Y. Fourniols, "A study about kalman filters applied to embedded sensors," *Sensors*, vol. 17, no. 12, pp. 1–18, 2017, ISSN: 14248220. DOI: 10.3390/s17122810.

[41] S. Garrow, C. Aksland, S. Sharma, and A. Alleyne, "Integrated Modeling for Battery Electric Vehicle Transcritical Thermal Mangement System," in *American Control Conference*, 2018.

[42] S. G. Garrow, "Dynamic Modeling and Control of Transcritical Vapor Compression System for Battery Electric Vehicle Thermal Management," Ph.D. dissertation, Univeristy of Illinois at Urbana-Champaign, 2018.

[43] R. Kwadzogah and M. Zhou, "Model predictive control for HVAC systems - a review," in *IEEE Internatioal Conference on Automation Science and Engineering*, 2013, pp. 442–447.

[44] R. Shah, A. G. Alleyne, C. W. Bullard, B. P. Rasmussen, and P. S. P. Hrnjak, "Dynamic modeling and control of single and multi-evaporator subcritical vapor compression systems," Ph.D. dissertation, 2003.

[45] U. E. P. Agency. "Dynamometer drive schedules." (2022), [Online]. Available: https://www.epa.gov/vehicle-and-fuel-emissions-testing/dynamometer-drive-schedules.

[46] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice- a Survey," *IFAC Proceedings Volumes*, vol. 21, no. 4, pp. 1–12, 1988. DOI: 10.1016/b978-0-08-035735-5.50006-1.

[47] Y. Ding, L. Wang, Y. Li, and D. Li, "Model predictive control and its application in agriculture: A review," *Computers and Electronics in Agriculture*, vol. 151, pp. 104–117, 2018, ISSN: 01681699. DOI: 10.1016/j.compag.2018.06.004. [Online]. Available: https://doi.org/10.1016/j.compag.2018.06.004.

[48] D. Hrovat, S. Di Cairano, H. Tseng, and I. Kolmanovsky, "The Development of Model Predictive Control in Automotive Industry: A Survey," in *IEEE International Conference on Control Applications*, 2012, pp. 295–302.

[49] C. Aksland, "Modular Modeling and Control of a Hybrid Unmanned Aerial," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2019.

[50] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model Predictive Control of Power Electronic Systems: Methods, Results, and Challenges," *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020. DOI: 10.1109/ojia.2020.3020184.

[51] H. Esen, T. Tashiro, D. Bernardini, and A. Bemporad, "Cabin heat thermal management in hybrid vehicles using model predictive control," in *22nd Mediterranean Conference on Control and Automation*, Palermo, Italy, 2014, pp. 49–54.

[52] J. Lopez-Sanz, C. Ocamp-Martinez, J. Alvarex-Florez, *et al.*, "Thermal Management in Plug-In Hybrid Electric Vehicles A Real-Time Nonlinear Model Predictive Control Implementation," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 7751–7760, 2017.

[53] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, no. 1, pp. 35–45, 1960, ISSN: 1528901X. DOI: 10.1115/1.3662552.

[54] B. P. Rasmussen and A. G. Alleyne, "Control-oriented modeling of transcritical vapor compression systems," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 126, pp. 54–64, 2004. DOI: 10.1115/1.1648312.

[55] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*, 2004, pp. 284–289, ISBN: 0780386361. DOI: 10.1109/cacsd.2004.1393890.

[56] C. E. Laird and A. G. Alleyne, "A hybrid electro-thermal energy storage system for high ramp rate power applications," in *DSCC2019*, 2019, pp. 1–9.

[57] V. S. Harish and A. Kumar, "A review on modeling and simulation of building energy systems," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1272–1292, 2016, ISSN: 18790690. DOI: 10.1016/j.rser.2015.12.040. [Online]. Available: http://dx.doi.org/10.1016/j.rser.2015.12.040.

Appendix A

Multi-State Graph-Based Heat Exchanger Model Example

This appendix guides the reader through the creation of a multi-state graph-based model for a two-phase heat exchanger discretized into 3 CVs. The graph model is displayed in Figure A.1. In Figure A.1 the vertex, edge, and input power flow indices are noted in gray. The states and power flows are noted in black. The chosen vertex and edge indices are nominal as they may be written in any order. However, changing the indices may result in different matrices from those derived in this section.

For this example it will be assumed that the state vector and power flow vectors are given

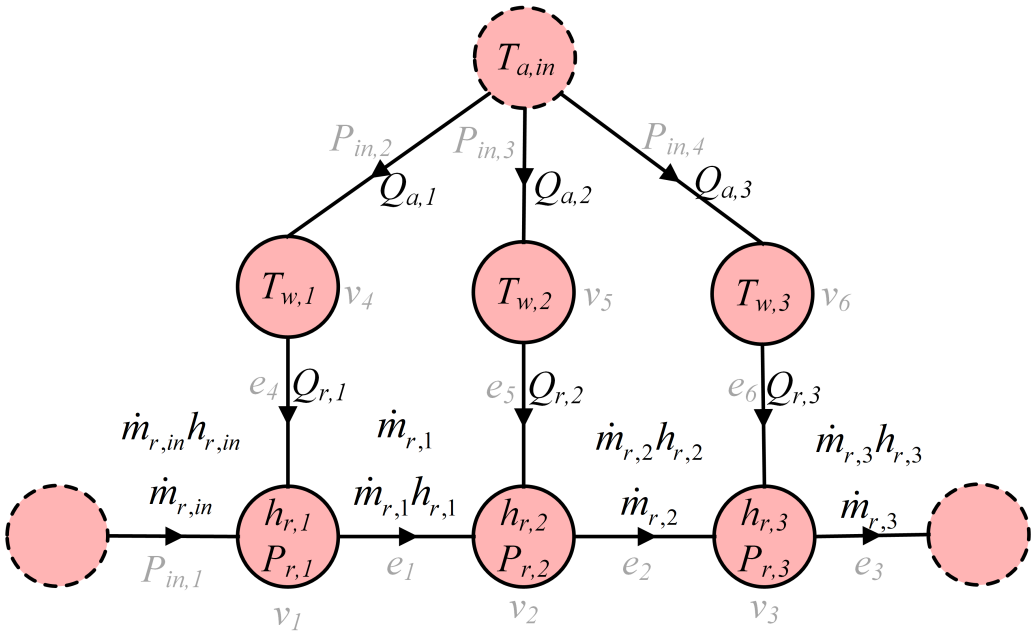


Figure A.1: 3 CV heat exchanger multi-state graph-based model example

as

$$
\begin{aligned}
x &= \begin{bmatrix} h_{r,1} & P_{r,1} & h_{r,2} & P_{r,2} & h_{r,3} & P_{r,3} & T_{w,1} & T_{w,2} & T_{w,3} \end{bmatrix}^T, \\
P &= \begin{bmatrix} \dot{m}_1 & \dot{m}_1 h_{r,1} & \dot{m}_2 h_{r,2} & \dot{m}_2 & \dot{m}_3 h_{r,3} & \dot{m}_3 & Q_{r,1} & Q_{r,2} & Q_{r,3} \end{bmatrix}^T, \\
P^{in} &= \begin{bmatrix} \dot{m}_{in} h_{in} & \dot{m}_{in} & Q_{a,1} & Q_{a,2} & Q_{a,3} \end{bmatrix}^T.
\end{aligned} \tag{A.1}
$$

Next the capacitance matrix needs to be defined before the adaptive matrices can be determined. The capacitance matrix is composed of 6 different block matrices, one for each vertex, as shown in (A.2).

$$
C = \begin{bmatrix}
C_{r,1} & 0 & 0 & 0 & 0 & 0 \\
0 & C_{r,2} & 0 & 0 & 0 & 0 \\
0 & 0 & C_{r,3} & 0 & 0 & 0 \\
0 & 0 & 0 & C_{w,1} & 0 & 0 \\
0 & 0 & 0 & 0 & C_{w,2} & 0 \\
0 & 0 & 0 & 0 & 0 & C_{w,3}
\end{bmatrix} \tag{A.2}
$$

The capacitance blocks associated with refrigerant vertices (i.e. v_1, v_2, and v_3) are 2×2 matrices. They contain two rows because there are two different dynamic equations calculated for each vertex, one for conservation of energy and a second for continuity. There are two columns because these vertices contain two states. Meanwhile each of the capacitance blocks associated with the wall vertices (i.e. v_4, v_5, and v_6) are a single element, corresponding to the conservation of energy equation. For this example, it will be assumed that the refrigerant capacitances are defined by (A.3),(A.4) and the wall capacitances are defined by (A.5).

$$
C_{r,i} = \begin{bmatrix}
\left(\frac{\partial \rho_{r,i}}{\partial h}\Big|_P + \rho_{r,i} \right) V_{r,i} & \left(\frac{\partial \rho_{r,i}}{\partial P}\Big|_h h_{r,i} - 1 \right) V_{r,i} \\
\frac{\partial \rho_{r,i}}{\partial h}\Big|_P V_{r,i} & \frac{\partial \rho_{r,i}}{\partial P}\Big|_h V_{r,i}
\end{bmatrix}^T \quad \text{for } i \in [1:2], \tag{A.3}
$$

$$
C_{r,i} = \begin{bmatrix}
\frac{\partial \rho_{r,i}}{\partial h}\Big|_P V_{r,i} & \frac{\partial \rho_{r,i}}{\partial P}\Big|_h V_{r,i} \\
\left(\frac{\partial \rho_{r,i}}{\partial h}\Big|_P + \rho_{r,i} \right) V_{r,i} & \left(\frac{\partial \rho_{r,i}}{\partial P}\Big|_h h_{r,i} - 1 \right) V_{r,i}
\end{bmatrix}^T \quad \text{for } i = 3, \tag{A.4}
$$

$$
C_{w,i} = m_w c_{p,w} \tag{A.5}
$$

Next the upper incidence matrix, $\bar{M}$, and D matrix are determined as done in the single-state

vertex case. For this example, they are

$$\overline{M} = \left[\begin{array}{c|c|c|c|c|c} 1 & 0 & 0 & -1 & 0 & 0 \\ \hline -1 & 1 & 0 & 0 & -1 & 0 \\ \hline 0 & -1 & 1 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right], D = \left[\begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{A.6}$$

The adaptive matrices are calculated to be

$$S_{\bar{M}} = \left[\begin{array}{cc|cc|cc|c|c|c} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right], S_D = \left[\begin{array}{cc|cc|c} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 \end{array}\right]. \tag{A.7}$$

$S_{\bar{M}}$ contains six rows of block matrices (one for each vertex) and six columns of block matrices (one for each edge). The top three rows of block matrices contain two rows each, as v_1, v_2, and v_3 have two different states. Meanwhile, the bottom three rows of block matrices contain only one row, as v_4, v_5, and v_6 represent only one state. Similarly, the first three columns of block matrices contain two columns, as e_1, e_2, and e_3 each contain two elements, whereas the last three columns of block matrices are only one column, as e_4, e_5, and e_6 only have one element. To determine the values of each element in a block matrix, the procedure outlined in Section 2.2 will be followed. For instance, in the first block of $S_{\bar{M}}$, the top row is $\begin{bmatrix} 0 & 1 \end{bmatrix}$ because the capacitance elements for the top row is associated with the conservation of energy equation and the second element in the e_1 is part of the energy domain (i.e. the capacitance of row 1 is a function of the second element in edge 1). Similarly, the second row in the first block matrix is $\begin{bmatrix} 1 & 0 \end{bmatrix}$, because the capacitance elements for the second row are part of the continuity equations and the first element in e_1 are in the mass conservation domain. The remainder of the elements defined in $S_{\bar{M}}$ can be determined through a similar process.

Following a similar process to creating $S_{\bar{M}}$, S_D contains six rows of block matrices (one

for each vertex) and four columns of block matrices (one for each input power flow). As with $S_{\bar{M}}$, the top three rows of block matrices in S_D contain two rows each, as v_1, v_2, and v_3 have two different states while the bottom three rows of block matrices contain only one row, as v_4, v_5, and v_6 have only one state. To determine the values of each element, the procedure outlined in Section 2.2 will be followed. Take for instance the block in the second row and first column of blocks in S_D. The first row is $\begin{bmatrix} 1 & 0 \end{bmatrix}$. This is because the capacitance in the first row of this block is part of the conservation of energy equations, while the first element in the first power flow are terms in the conservation of energy equation. On the other hand, the second row in this block matrix is $\begin{bmatrix} 0 & 1 \end{bmatrix}$ as the second row of capacitance is associated with the continuity equations and the second element in the power flow matrix is part of the continuity equations. The remainder of the blocks can be defined in a similar manner.