View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Illinois Digital Environment for Access to Learning and Scholarship Repository

© 2013 Justin Koeln

A DECENTRALIZED CONTROL DESIGN APPROACH TO A CLASS OF LARGE-SCALE SYSTEMS

BY

JUSTIN KOELN

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 at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Professor Andrew Alleyne

Abstract

Large-scale systems present a unique control challenge. The large number of states, actuators, and control objectives for these systems often restricts the ability to analyze and control the system as a whole. Typically, these large systems are decomposed into multiple smaller subsystems which can be analyzed and controlled separately using a decentralized control approach. However, if the interactions between subsystems significantly affect the dynamics of the system, a decentralized control approach may prove to be ineffective and even result in unstable behavior.

This thesis develops a control strategy for a class of systems with a particular hierarchical structure known as a Block Arrow Structure (BAS). Many real world systems naturally exhibit this two-level hierarchical structure, where a common subsystem at the higher, global, level interacts with multiple subsystems at the lower, local, level. There is no direct interaction among the lower level subsystems. A standard decentralized control approach would control each subsystem separately, ignoring the interactions between the higher and lower level subsystems. However, the interaction between the two levels may significantly affect the system dynamics, rendering the decentralized control approach ineffective. The proposed control strategy, referred to as the BAS control strategy, retains the scalability of the decentralized control approach but is also able to directly consider the interactions between the higher and lower level subsystems. This allows the BAS control approach to perform significantly better than a decentralized approach. Model predictive control (MPC) is used to evaluate the performance of the BAS control strategy relative to both centralized and decentralized approaches for two different BAS systems.

In addition to the BAS control approach, this thesis develops an extremum seeking control (ESC) strategy which is used to improve the overall efficiency of the BAS system. In

addition to performance objectives such as tracking a desired value for a state of the system, many systems have an efficiency objective. This objective seeks to control the system in the most efficient way possible, while still meeting the performance objectives. Minimizing the total energy use of all the actuators in the system is a common example of such an efficiency objective. In this work, ESC is used to augment the BAS control strategy at the global level to further improve the efficiency of the overall system. The model-free nature of ESC makes this control strategy especially effective in the presence of unknown disturbances and system nonlinearity, which may not be captured by the models used for the MPC controllers of the BAS control strategy.

A linear example system is used to demonstrate the concepts and ideas presented throughout this thesis. For this example system, the BAS control architecture with ESC is able to achieve a control performance very similar to that of the centralized control approach while retaining the scalability of the decentralized approach. The benefits of the BAS control approach are also demonstrated for a more realistic system: a variable refrigerant flow (VRF) air-conditioning and refrigeration system for a building. Through a gray-box modeling approach, it is shown that VRF systems naturally exhibit a BAS structure and, therefore, can benefit from a BAS control approach. VRF systems are becoming widely used to meet the air-conditioning and refrigeration needs of buildings because of their greater efficiency in removing heat versus the conventional forced air systems. For these systems, it is very important to meet both the performance objectives, such as maintaining a desired air temperature in a room, as well as the efficiency objective of minimizing the total energy consumed by the system. Through a series of simulation examples, the BAS control approach is found to be a very effective control strategy for meeting both of these objectives.

To my family, friends, and teachers.

Acknowledgements

First, I would like to thank my advisor, Dr. Andrew Alleyne, for his invaluable and continuing commitment to help me reach my greatest potential. His mentorship over the last two years has allowed me to make tremendous strides as a researcher. As a shining example of what it means to be a leader, Dr. Alleyne has taught me priceless lessons that I will take with me throughout my career. I am very honored to have his continued mentorship as I begin my work towards a Ph.D.

I must also thank my family. My parents have always set an outstanding example and truly are my biggest role models. They instilled in me the value of education and the curiosity and determination required to become a successful researcher. They have always been my biggest supporters and I would not be where I am without them.

These acknowledgements would not be complete without thanking my fellow Alleyne Research Group (ARG) members. I consider myself extremely lucky to have had the opportunity to work alongside the current and past member of the ARG. To Megan, Joey, Sarah, Dave, Kira, Neera, Tim, Vikas, Matt, Erick, Nanjun, Yangmin, Bin, and Kasper, I want to thank each one of you for making my time at UIUC so enjoyable and hope we can continue to keep in touch as we go in our separate directions. I especially want to thank Neera Jain for her large role in the work on VRF systems presented in Chapter 7.

Finally, I would like to acknowledge the support of the National Science Foundation (NSF) and the Graduate Research Fellowship Program (GRFP). Their support has given me the flexibility to conduct research which both interests me and is of value to society.

Table of Contents

CHAPTER Page

LIST OF TABLES ix

LIST OF FIGURES x

NOMENCLATURE xiv

ABBREVIATIONS xvi

CHAPTER 1 INTRODUCTION 1

1.1 Motivation 2

1.2 Organization of Thesis 3

CHAPTER 2 A CLASS OF BLOCK ARROW STRUCTURE SYSTEMS 4

2.1 Block Arrow Structure 4

2.2 Literature Review 6

2.3 BAS System Analysis 7

2.4 Control of BAS Systems 11

2.4.1 Control Objectives 11

2.4.2 Information Constraints 12

2.4.3 Actuator, State, and Output Constraints 13

2.5 Example System 14

2.5.1 Example System Description 14

2.5.2 Example System Control Objectives 18

CHAPTER 3 MODEL PREDICTIVE CONTROL (MPC) 20

3.1 Basic MPC Formulation 21
3.2 MPC Formulation for Example System 24

CHAPTER 4 EXTREMUM SEEKING CONTROL 30

4.1 Basic ESC Formulation 30
4.2 ESC Parameter Selection 33
4.3 ESC Augmentations 34
4.4 ESC Formulation for Example System 35

CHAPTER 5 CONTROLLER ARCHITECTURES 37

5.1 Centralized Control Architecture 38
5.1.1 Centralized Control Problem 40
5.1.2 Centralized Formulation for Example System 40

5.2 Decentralized Control Architecture 41
5.2.1 Decentralized Control Problem 43
5.2.2 Decentralized Formulation for Example System 43

5.3 BAS Control Architecture 44
5.3.1 BAS Control Problem 46
5.3.2 BAS Formulation for Example System 47

5.4 ESC Control Architecture 47
5.4.1 ESC Control Problem 49
5.4.2 ESC Formulation for Example System 50

CHAPTER 6 ILLUSTRATIVE EXAMPLE SYSTEM 51

6.1 System Parameters and Analysis 51
6.2 Control Architecture Comparison 54
6.2.1 Baseline Controller Performance 55
6.2.2 Augmented Controller Performance via ESC 64
6.2.3 Scalability 69

CHAPTER 7 VRF SYSTEM 73

7.1 VRF Background 73

7.2 System Modeling 77
7.2.1 Fluid Dynamics 78
7.2.2 Thermal Dynamics 80
7.3 Complete System Model 85
7.4 Model Validation 90
7.5 Control Design and Analysis 97
7.5.1 Controllability 97
7.5.2 Control Objectives 98
7.5.3 MPC Formulation 104
7.5.4 Decentralized Control 106
7.6 Simulation Results 108
7.6.1 Baseline Scenario 108
7.6.2 ESC Scenario 114
7.6.3 Realistic Scenario 118
CHAPTER 8 CONCLUSION 124
8.1 Summary of Research Contributions 124
8.2 Future Work 126
8.2.1 Theory 126
8.2.2 Application 126
REFERENCES 128
APPENDIX A EXAMPLE SYSTEM MATLAB CODE 132
A.1 Parameters and System Development 132
A.2 MPC Formulation 141
A.3 MPC Code 143
APPENDIX B VRF SYSTEM MATLAB CODE 145
B.1 Parameters and System Development 145
B.2 MPC Formulation 152
B.3 MPC Code 155

List of Tables

Table 6.1 Example System Parameters 52
Table 6.2 ESC Parameters 65
Table 7.1 Electrical Circuit Analogues for Fluid Dynamics of VRF Systems 79
Table 7.2 Electrical Circuit Analogues for Thermal Dynamics of VRF Systems 81
Table 7.3 VRF System Matrix Elements 87
Table 7.4 VRF System Matrix Elements Continued 88
Table 7.5 Identified Fluid and Thermal Parameters 91
Table 7.6 Actuator Constraints for VRF System 105
Table 7.7 ESC Parameters for VRF System 115

List of Figures

Figure 2.1 Subsystem interaction diagram for BAS systems. 5
Figure 2.2 Electrical circuit diagram for example BAS system. 15
Figure 3.1 Model predictive control. 21
Figure 4.1. ESC schematic. 31
Figure 4.2. Modified ESC schematic. 36
Figure 5.1 Qualitative comparison between centralized, C, decentralized, D, and BAS, B, control architectures – a) relative plant knowledge and communication, b) relative performance and scalability. 38
Figure 5.2 Centralized control architecture. 39
Figure 5.3 Decentralized control architecture. 42
Figure 5.4 BAS control architecture. 45
Figure 5.5 Augmented control architecture with ESC. 49
Figure 6.1 System disturbances $\boldsymbol{I}_{Li}$ and $\boldsymbol{V}_{h}$ for Scenario 1. 55
Figure 6.2 Open-loop response for $\boldsymbol{V}_{ai}$ due to the disturbances for Scenario 1. 56
Figure 6.3 Open-loop response for $\boldsymbol{I}_{e}$ due to the disturbances for Scenario 1. 56
Figure 6.4 Ability of each control architecture to meet local performance objectives by tracking the desired value for $\boldsymbol{V}_{ai}$ for Scenario 1. 57
Figure 6.5 Ability of each control architecture to meet the global performance objective by tracking the desired value for $\boldsymbol{I}_{e}$ for Scenario 1. 57
Figure 6.6 Ability of each control architecture to meet the global efficiency objective by minimizing the total actuator cost for Scenario 1. 58
Figure 6.7 Actuator inputs $\boldsymbol{u}_{i}$ from each control architecture for Scenario 1. 59
Figure 6.8 Actuator inputs $\boldsymbol{u}_{ai}$ from each control architecture for Scenario 1. 59

Figure 6.9 Actuator inputs $\boldsymbol{u}_t$ and $\boldsymbol{u}_g$ from each control architecture for Scenario 1. 60
Figure 6.10 Improved system efficiency achieved by the modified BAS controller for Scenario 1. ... 61
Figure 6.11 Change in $\boldsymbol{u}_g$ when using the modified BAS controller for Scenario 1. 61
Figure 6.12 System disturbances $\boldsymbol{I}_{Li}$ and $\boldsymbol{V}_h$ for Scenario 2. 62
Figure 6.13 Actuator inputs $\boldsymbol{u}_i$ from each control architecture for Scenario 2. 63
Figure 6.14 Actuator inputs $\boldsymbol{u}_{ai}$ from each control architecture for Scenario 2. 63
Figure 6.15 Ability of each control architecture to meet local performance objectives by tracking the desired value for $\boldsymbol{V}_{ai}$ for Scenario 2. 64
Figure 6.16 Actuator inputs $\boldsymbol{u}_t$ and $\boldsymbol{u}_g$ from each control architecture for Scenario 2. 64
Figure 6.17 System disturbances $\boldsymbol{I}_{Li}$ and $\boldsymbol{V}_h$ for ESC Scenario. 65
Figure 6.18 Ability of each control architecture to meet local performance objectives by tracking the desired value for $\boldsymbol{V}_{ai}$ for ESC Scenario. 66
Figure 6.19 Actuator inputs $\boldsymbol{u}_i$ from each control architecture for Scenario 1. 67
Figure 6.20 Actuator inputs $\boldsymbol{u}_{ai}$ from each control architecture for Scenario 1. 67
Figure 6.21 Actuator inputs $\boldsymbol{u}_t$ and $\boldsymbol{u}_g$ from each control architecture for Scenario 1. 68
Figure 6.22 Improved system efficiency achieved by the modified BAS controller for Scenario 1. ... 69
Figure 6.23 Computation time for each control approach for Scenario 1. 71
Figure 6.24 Computation time for each control approach for Scenario 1. 72
Figure 7.1 Single-evaporator VCS – a) 4 component system schematic, b) P-h diagram of vapor compression cycle. 74
Figure 7.2 VRF system – a) Industrial system used to heat and cool multiple rooms in a building [31], b) schematic of VRF system. 75
Figure 7.3 Electrical circuit diagram of the fluid dynamics of a VRF system. The symbol ○— indicates the presence of a control input. 79
Figure 7.4 Electrical circuit diagram of the thermal dynamics of a VRF system. The symbol ○— indicates the presence of a control input. 81
Figure 7.5 ATTMO model of 5-evaporator VRF system. 91
Figure 7.6 Compressor and condenser fan input signal for parameter identification. 93
Figure 7.7 EEV and evaporator fan input signal for parameter identification. 93

Figure 7.8 Validation of linear model for pressure, wall temperature, and room air temperature for evaporator 1. 95
Figure 7.9 Validation of linear model for condenser pressure, junction pressure, and junction temperature. 96
Figure 7.10 Linear approximation of the relationship between saturation pressure and saturation temperature for R-134a. 101
Figure 7.11 Power consumption for heat exchanger fans. 102
Figure 7.12 Power consumption for compressor. 103
Figure 7.13 Compressor speed and EEV inputs used to develop compressor power map. 103
Figure 7.14 Junction superheat response for changes in compressor speed and condenser fan speed from ATTMO model and linear models. 107
Figure 7.15 Changes in compressor speed and condenser fan speed for superheat analysis. 107
Figure 7.16 System disturbances $\boldsymbol{Q_{Li}}$ and $\boldsymbol{T_{ac}}$ for baseline scenario 108
Figure 7.17 Open-loop response for $\boldsymbol{T_{ai}}$ due to disturbances for baseline scenario. 109
Figure 7.18 Open-loop response for $\boldsymbol{T_{SH,q}}$ due to disturbances for baseline scenario. 109
Figure 7.19 Ability of each control architecture to meet local performance objectives by tracking the desired value for $\boldsymbol{T_{ai}}$ for baseline scenario. 110
Figure 7.20 Ability of each control architecture to track the desire junction superheat for baseline scenario. 110
Figure 7.21 Different refrigerant superheat for each evaporator for baseline scenario. 111
Figure 7.22 Ability of each control architecture to meet the global efficiency objective by minimizing the total power consumption by the actuators for baseline scenario 111
Figure 7.23 Actuator inputs $\boldsymbol{a_{vi}}$ from each control architecture for baseline scenario. 112
Figure 7.24 Actuator inputs $\boldsymbol{\omega_{fi}}$ from each control architecture for baseline scenario. 113
Figure 7.25 Actuator inputs $\boldsymbol{\omega_{k}}$ from each control architecture for baseline scenario. 113
Figure 7.26 Actuator inputs $\boldsymbol{\omega_{fc}}$ from each control architecture for baseline scenario. 114
Figure 7.27 Ability of each control architecture to meet local performance objectives by tracking the desired value for $\boldsymbol{T_{ai}}$ for ESC scenario. 115
Figure 7.28 Ability of each control architecture to track the desired junction superheat for baseline scenario. 116

Figure 7.29 Ability of each control architecture to meet the global efficiency objective by minimizing the total power consumption by the actuators for ESC scenario. 116
Figure 7.30 Actuator inputs $\boldsymbol{a_{vi}}$ from each control architecture for ESC scenario. 117
Figure 7.31 Actuator inputs $\boldsymbol{\omega_{fi}}$ from each control architecture for ESC scenario. 117
Figure 7.32 Actuator inputs $\boldsymbol{\omega_{k}}$ from each control architecture for ESC scenario. 118
Figure 7.33 Actuator inputs $\boldsymbol{\omega_{fc}}$ from each control architecture for ESC scenario. 118
Figure 7.34 System disturbances $\boldsymbol{Q_{Li}}$ and $\boldsymbol{T_{ac}}$ for realistic scenario. 119
Figure 7.35 Ability of BAS control architecture to meet local performance objectives by tracking the desired value for $\boldsymbol{T_{ai}}$ for realistic scenario 120
Figure 7.36 Ability of BAS control architecture to track the desire junction superheat for realistic scenario. 120
Figure 7.37 Different refrigerant superheat for each evaporator for realistic scenario. 121
Figure 7.38 Ability of BAS control architecture to meet the global efficiency objective by minimizing the total power consumption by the actuators for realistic scenario. 121
Figure 7.39 Actuator inputs $\boldsymbol{a_{vi}}$ from BAS control architecture for realistic scenario. 122
Figure 7.40 Actuator inputs $\boldsymbol{\omega_{fi}}$ from BAS control architecture for realistic scenario. 122
Figure 7.41 Actuator inputs $\boldsymbol{\omega_{k}}$ from BAS control architecture for realistic scenario. 123
Figure 7.42 Actuator inputs $\boldsymbol{\omega_{fc}}$ from BAS control architecture for realistic scenario. 123

Nomenclature

$\dot{m}$	Mass flow rate
$\dot{Q}$	Heat transfer rate
0	Index referring to the common subsystem
a	Perturbation amplitude
a_v	EEV aperture
C	Capacitance
d	Disturbance vector
E	Interconnection matrix
e	Error state vector
$G(s)$	System transfer function
i	Index, typically refers to i^{th} subsystem
I	Current
j	Index
J	Cost
J_p	Performance cost
J_u	Input cost, associated with efficiency objective
$J_{\Delta u}$	Change in input cost
k	Sample
k	ESC gain
$\alpha,\beta,\eta,K,\kappa,\lambda,\mu$	Linearization coefficients
N	Number of subsystems
$\mathcal{N}$	Set of subsystems
n_d	Number of disturbances
N_p	Prediction horizon
n_r	Number of references
n_s	Number of states

n_u	Number of inputs
N_u	Control horizon
n_y	Number of outputs
P	Pressure
Q	Weighting matrix
q	Subscript for VRF junction
R	Reachability matrix
R	Resistance
r	Reference
$\mathbf{S}$	Subsystem
T	Temperature
T_{SH}	Superheat
u	Input vector
U	Lifted control input vector
V	Voltage
x	State vector
y	Output vector
z	Integral of error state vector
Δt	Sample time
γ	Weighting term
κ	Actuator cost
φ	Perturbation phase shift
$\tilde{\rho}$	Generic rank
ω	Perturbation frequency
ω_f	Evaporator fan speed
ω_{fc}	Condenser fan speed
ω_h	High-pass filter frequency
ω_k	Compressor speed
ω_l	Low-pass filter frequency

Abbreviations

AFRL	Air Force Research Laboratory
ATTMO	AFRL Transient Thermal Modeling and Optimization
BAS	Block Arrow Structure
BBD	Bordered Block Diagonal
DSAS	Dither Signal Amplitude Schedule
EEV	Electronic Expansion Valve
EKF	Extended Kalman Filter
ESC	Extremum Seeking Control
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
ME-VCS	Multi-Evaporator Vapor Compression System
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
PI	Proportional-Integral
SISO	Single-Input Single-Output
VCC	Vapor Compression Cycle
VCS	Vapor Compression System
VRF	Variable Refrigerant Flow

Chapter 1
Introduction

This thesis investigates the control architecture design for a class of large-scale systems with a particular structure known as a Block Arrow Structure (BAS). In general, the control of large-scale systems is a particularly challenging problem that has been the focus of many research efforts over the past few decades. The large number of actuators, states, and control objectives for these systems often restricts the ability to analyze and control the system as a whole due to the high communication and computational costs associated with a centralized control approach. Often, large-scales systems can be decomposed into interacting subsystems, permitting a decentralized control approach where multiple controllers are developed and each controller only has access to information for the corresponding subsystem. By reducing a single large control problem into multiple smaller control problems, communication and computational costs can be drastically reduced, improving the practicality and scalability of the control approach. However, the decentralization of the control problem comes at a cost. Without explicit knowledge of the interactions between subsystems, the control performance of a decentralized approach can be significantly degraded when compared to a centralized approach.

In this thesis, several control architectures with various levels of decentralization are developed and analyzed for a class of BAS systems. BAS systems, also known as bordered block diagonal (BBD) systems, have a two-level hierarchical structure. The lower, local, level consists of a set of subsystems which are completely decoupled, meaning that there is no direct interaction between these subsystems. The higher, global, level contains a single subsystem which has a bidirectional interaction with each of the subsystems in the lower level. This structure is defined more rigorously in Chapter 2. The goal of this thesis is to utilize the unique

structure of these systems to develop a control strategy that provides both the scalability of decentralized control and the high performance of centralized control.

1.1 Motivation

Despite extensive research efforts focused on the exploitation of the BAS (often referred to as BBD) in relation to parallel computing [1] [2], surprisingly little work has studied the control of systems with this structure. BAS systems are found in a variety of applications including power systems, resource management, and hydraulic systems [3]. Additionally, any sparse matrix can be rearranged into a nested BBD form [4], allowing the use of BAS control techniques on a wide variety of large-scale systems.

Previous work on the development of control strategies for BAS system, which is summarized in Chapter 2, has focused exclusively on the use of Linear Matrix Inequalities (LMIs) or Linear Quadratic Regulators (LQR) to design static feedback controllers which exploit the structure of a BAS system. This thesis aims to extend this work through the development of a Model Predictive Control (MPC) framework which also exploits the system structure while providing the additional benefits of MPC [5]; in particular, the ability to enforce constraints and use predictions of how the system responds to various control decisions.

In additional to developing a MPC framework for BAS systems, the ideas and techniques developed throughout this thesis are demonstrated for a Variable Refrigerant Flow (VRF) system in Chapter 7. VRF systems, also known as multi-evaporator vapor-compression systems, are becoming widely used to meet the heating and cooling demands for buildings [6]. These large-scale systems are often used to heat or cool over 30 different zones or rooms with a single system. With multiple states and actuators per zone, a centralized control approach is often infeasible, suggesting the use of decentralized control. However, the large degree of interaction between the various subsystems in a VRF system poses a problem. In [7], it was found that despite this coupling, decentralized control can be effective in meeting the performance objective for the system, namely regulating a desired air temperature or cooling capacity for each room in the building. However, heating and cooling consume a significant portion of the energy use in buildings each year [8], and therefore, the efficiency of these systems is also of great importance. In Chapter 7, it is found that VRF systems have a natural BAS which can be exploited when

developing a control strategy used to meet both the performance and efficiency requirements for these systems. While exploiting the structure of VRF systems can lead to improved control performance, it is found that the nonlinearity and numerous disturbances seen in VRF systems limit the capabilities of a MPC control strategy which is based on the assumption of a linear system model. Therefore, in addition to MPC, a model-free adaptive control approach known as Extremum Seeking Control (ESC) is development and implemented to further improve the efficiency of the VRF system.

1.2 Organization of Thesis

The remainder of the thesis is organized as follows. The class of BAS systems is defined and analyzed in Chapter 2. Additionally, a linear example system is introduced which is used to demonstrate the ideas and techniques presented throughout the thesis. In Chapter 3, model predictive control is introduced along with the details of the specific MPC formulation used in this thesis. The basics of extremum seeking control are presented in Chapter 4 along with some details pertaining to the implementation of ESC on a physical system. Chapter 5 develops the various control architectures used to control the linear example system with the results shown in Chapter 6. A brief overview of variable-refrigerant flow systems is given in Chapter 7, followed by the details of a gray-box model identification approach used to obtain a linear model representation for the system. Chapter 7 also shows how each of the various control architectures can be used to control a VRF system and the associated control performance. Finally, some concluding remarks and future research directions are presented in Chapter 8.

Chapter 2

A Class of Block Arrow Structure Systems

This chapter presents a class of systems with Block Arrow Structure (BAS). Section 1.1 develops a linear system representation for BAS systems and Section 1.2 details previous work on the analysis and control of these systems. Various features of BAS systems relevant to control are analyzed in Section 2.3 and general control objectives for these systems are presented in Section 2.4. Finally, Section 2.5 describes an example BAS system which is used to demonstrate the various ideas and techniques presented throughout this thesis.

2.1 Block Arrow Structure

In this work we consider linear, time-invariant systems with a block arrow structure (BAS), also referred to as a bordered block diagonal (BBD) structure. BAS systems consist of N subsystems $\mathbf{S}_i$, where $i \in \mathcal{N} = \{1, 2, ..., N\}$, interconnected through a single common subsystem $\mathbf{S}_0$. Fig. 2.1 visually shows the structure of these systems where each subsystem $\mathbf{S}_i$ has a bidirectional interaction with $\mathbf{S}_0$ but no interaction with subsystem $\mathbf{S}_j$ where $i, j \in \mathcal{N}$ and $j \neq i$.

Each subsystem $\mathbf{S}_i$ can be represented in state-space form as

$$\begin{aligned} \mathbf{S}_i : \dot{x}_i &= A_{ii}x_i + A_{i0}x_0 + B_{ii}u_i + B_{i0}u_0 + V_{ii}d_i, \\ y_i &= C_{ii}x_i + D_{ii}u_i + W_{ii}d_i, \end{aligned} \tag{2.1}$$

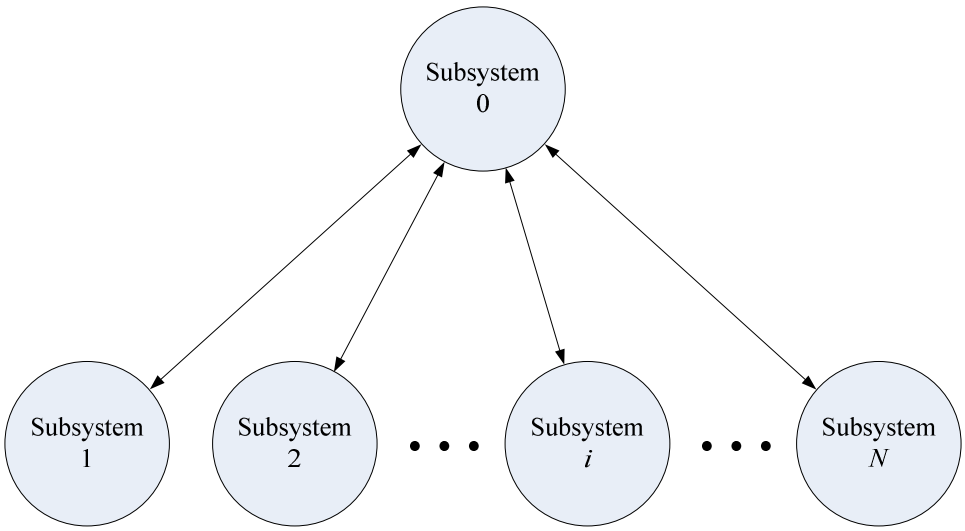

Figure 2.1 Subsystem interaction diagram for BAS systems.

where $x_i \in \mathbb{R}^{n_{s,i}}$, $u_i \in \mathbb{R}^{n_{u,i}}$, and $d_i \in \mathbb{R}^{n_{d,i}}$ are the state, control, and disturbance vectors for $\mathbf{S}_i$ and $x_0 \in \mathbb{R}^{n_{s,0}}$, $u_0 \in \mathbb{R}^{n_{u,0}}$ are the state and control vectors for $\mathbf{S}_0$. Note that it is assumed that the disturbances d_i are naturally decoupled. Additionally, throughout this work, it is assumed that all states are outputs of the system: $y = x$. Thus $C_{ii} = I_{n_{s,i} \times n_{s,i}}$, $D_{ii} = 0_{n_{s,i} \times n_{u,i}}$, and $W_{ii} = 0_{n_{s,i} \times n_{d,i}}$. The subsystem $\mathbf{S}_0$ can be represented in state-space form as

$$\mathbf{S}_0 : \dot{x}_0 = A_{00}x_0 + \sum_{i=1}^{N} A_{0i}x_i + B_{00}u_0 + \sum_{i=1}^{N} B_{0i}u_i + V_{00}d_0. \tag{2.2}$$

The subsystem representations from (2.1) and (2.2) can be combined to create the complete system $\mathbf{S}$, represented as

$$\begin{aligned} \mathbf{S} : \dot{x} &= Ax + Bu + Vd, \\ y &= x, \end{aligned} \tag{2.3}$$

where $x = \begin{bmatrix} x_1 & x_2 & \dots & x_N & x_0 \end{bmatrix}^T$, $u = \begin{bmatrix} u_1 & u_2 & \dots & u_N & u_0 \end{bmatrix}^T$, and $d = \begin{bmatrix} d_1 & d_2 & \dots & d_N & d_0 \end{bmatrix}^T$ are the state, control, and disturbance vectors for the entire system. Denoting $n_s = \sum_{i=1}^{N} n_{s,i} + n_{s,0}$,

$n_u = \sum_{i=1}^{N} n_{u,i} + n_{u,0}$ and $n_d = \sum_{i=1}^{N} n_{d,i} + n_{d,0}$, we have $x \in \mathbb{R}^{n_s}$, $u \in \mathbb{R}^{n_u}$, and $d \in \mathbb{R}^{n_d}$ with $A \in \mathbb{R}^{n_s \times n_s}$, $B \in \mathbb{R}^{n_s \times n_u}$, and $V \in \mathbb{R}^{n_s \times n_d}$. Writing out the A, B, and V matrices as

$$A = \left[\begin{array}{cccc|c} A_{11} & 0 & \cdots & 0 & A_{10} \\ 0 & A_{22} & \ddots & \vdots & A_{20} \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & A_{NN} & A_{N0} \\ \hline A_{01} & A_{02} & \cdots & A_{0N} & A_{00} \end{array}\right], \qquad B = \left[\begin{array}{cccc|c} B_{11} & 0 & \cdots & 0 & B_{10} \\ 0 & B_{22} & \ddots & \vdots & B_{20} \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & B_{NN} & B_{N0} \\ \hline B_{01} & B_{02} & \cdots & B_{0N} & B_{00} \end{array}\right], \tag{2.4}$$

$$V = \left[\begin{array}{cccc|c} V_{11} & 0 & \cdots & 0 & 0 \\ 0 & V_{22} & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & V_{NN} & 0 \\ \hline 0 & 0 & \cdots & 0 & V_{00} \end{array}\right], \tag{2.5}$$

it is easy to see why these systems are said to have a block arrow structure.

2.2 Literature Review

The majority of BAS research has focused on the exploitation of the structure for parallel computing purposes [1] [2]. However, there have been several efforts to utilize the structure of BAS systems in the development of control strategies for these systems. Most of this work has come from Dr. Groumpos and colleagues [3] [9] [10] [11], where a linear-quadratic regulator (LQR) approach is used to develop a static feedback control law which preserves the BAS form. In [9] it is found that the development of a BAS control law can be decomposed into the solution of multiple smaller independent algebraic Riccati equations. In addition to the reduction in size of each of these Riccati equations, the fact that these equations can be solved independently allows for the use of parallel processing, which further reduces computational time when compared to a centralized approach. It was found that the BAS approach proved a desirable compromise between centralized and completely decentralized control strategies by combining the low computational complexity of decentralized control with the high performance of centralized control. The controllability and stability of BAS systems using the proposed BAS

static feedback control law is analyzed in [3] and [10]. Additionally, in [11] it is found that a gradient-type algorithm can be used to optimize the BAS feedback gain and is shown to significantly improve the performance of the BAS control strategy; nearly achieving the same performance as the centralized approach.

The other set of research related to the control of BAS systems comes from Siljak and colleagues [4] [12], where a linear matrix inequality (LMI) approach is used to determine the BAS feedback gain. In [4], a graph-theoretic decomposition method is presented which allows any sparse system matrix to be reordered into a nested bordered block diagonal form. Both [4] and [12] use an LMI approach to develop output feedback control gains for nested BAS systems.

The previous literature has shown that a BAS control approach can provide a significant improvement in control performance when compared to decentralized control. However, the previous approaches, using LQR and LMI methods to design static feedback control laws, may not meet some of the practical needs of industrial applications. Examples of these needs include the ability to directly consider state, output, or actuator saturation and to predict the future state of the system. Therefore, this thesis extends the notions of BAS control design to a model predictive control (MPC) framework which allows for direct consideration of state, output, and actuator saturation as well as increased flexibility in the control design.

2.3 BAS System Analysis

Prior to the development of a control strategy of any system, the fundamental properties of controllability and observability, first presented in [13], need to be tested. Controllability means that each state of the system can be moved from any initial condition to any final condition in finite time under some control input. Observability means that each state can be determined in finite time based on knowledge of the inputs and outputs of the system. The standard tests for controllability and observability are

$$\text{rank } \mathcal{C} = \text{rank}\begin{bmatrix} B & AB & \dots & A^{n_s-1}B \end{bmatrix} = n_s, \tag{2.6}$$

and

$$\text{rank } \mathcal{O} = \text{rank}\begin{bmatrix} C & CA & \dots & CA^{n_s-1} \end{bmatrix}^T = n_s, \tag{2.7}$$

where n_s is the number of states in the system. Based on the duality of controllability and observability, only controllability is considered in the following analysis and discussion and the results can easily be adapted for observability.

For large-scale systems, determining the rank of $\mathcal{C} \in \mathbb{R}^{n_s \times (n_s \cdot n_u)}$, where n_u is the number of inputs to the system, may become a very difficult numerical problem to solve. Fortunately, as presented in [14] there is an alternative method to determine the controllability of large-scale systems. Here the notions of structured matrices and structural controllability are used to simplify the analysis.

Definition. [14] An $n \times m$ matrix $\tilde{M} = (\tilde{m}_{ij})$ is said to be a structured matrix if its elements $\tilde{m}_{ij}$ are either fixed zeros or independent free parameters.

For example, the scalar BAS matrix

$$M = \begin{bmatrix} -2 & 0 & 1 \\ 0 & -2 & 2 \\ 1 & 2 & -3 \end{bmatrix} \tag{2.8}$$

is admissible with respect to the structured matrix

$$\tilde{M} = \begin{bmatrix} * & 0 & * \\ 0 & * & * \\ * & * & * \end{bmatrix}. \tag{2.9}$$

Definition. [14] A pair of matrices $(\tilde{A}, \tilde{B})$ is said to be structurally controllable if there exists a controllable pair (A, B) such that $(A, B) \in (\tilde{A}, \tilde{B})$.

Thus a system is structurally controllable, if there exists a controllable system (A, B) with the given structure $(\tilde{A}, \tilde{B})$. By not taking into account the actual values in the A and B matrices, the test for structural controllability scales to large systems significantly better than the test for controllability found in (2.6). However, some systems may be found to be structurally controllable but are not controllable due to the correlation between parameters in the A and B matrices. This issue will be further addressed in Chapter 7.

Prior to presenting the conditions to test for structural controllability, several notions need to be introduced. First is the notion of *generic*, or *term, rank*. The generic rank of $\tilde{M}$, denoted as $\tilde{\rho}\left(\tilde{M}\right)$, is the maximal rank that $\tilde{M}$ can achieve through the appropriate selection of the numerical values for the undetermined elements of $\tilde{M}$. Second is the notion of *input reachability*. With additional details found in [14], the concept of input reachability comes from graph theory where a vertex v_i is reachable from v_j if there exists a path from v_j to v_i. The existence of this path is based on the structure of the graph and reachability can be tested using the notion of structured matrices presented above. For a generic system

$$\begin{aligned} \mathbf{S}: \dot{x} &= Ax + Bu, \\ y &= Cx, \end{aligned} \tag{2.10}$$

the Boolean matrices $\overline{A} = \left(\overline{a}_{ij}\right)$, $\overline{B} = \left(\overline{b}_{ij}\right)$, and $\overline{C} = \left(\overline{c}_{ij}\right)$ are defined with the elements

$$\overline{a}_{ij} = \begin{cases} 1, & a_{ij} \neq 0, \\ 0, & a_{ij} = 0, \end{cases} \quad \overline{b}_{ij} = \begin{cases} 1, & b_{ij} \neq 0, \\ 0, & b_{ij} = 0, \end{cases} \quad \overline{c}_{ij} = \begin{cases} 1, & c_{ij} \neq 0, \\ 0, & c_{ij} = 0. \end{cases} \tag{2.11}$$

Definition. [14] The interconnection matrix of $\mathbf{S}$ is a binary $\left(n_s + n_u + n_y\right) \times \left(n_s + n_u + n_y\right)$ matrix $E = \left(e_{ij}\right)$ defined as

$$E = \begin{bmatrix} \overline{A} & \overline{B} & 0 \\ 0 & 0 & 0 \\ \overline{C} & 0 & 0 \end{bmatrix}. \tag{2.12}$$

Using the interconnection matrix E the reachability matrix R is defined as

$$R = E \vee E^2 \vee \ldots \vee E^s, \tag{2.13}$$

where $s = n_s + n_u + n_y$, $E^k = E^{k-1} \wedge E$, and the Boolean operators $\wedge$ and $\vee$ represent *and* and *or* operations. An efficient method for calculating R is presented in [14]. For $d \geq 2$, E^d may be calculated as

$$E^d = \begin{bmatrix} \bar{A}^d & \bar{A}^{d-1}\bar{B} & 0 \\ 0 & 0 & 0 \\ \bar{C}\bar{A}^{d-1} & \bar{C}\bar{A}^{d-2}\bar{B} & 0 \end{bmatrix}. \tag{2.14}$$

From (2.13) and (2.14) the reachability matrix can be written as

$$R = \begin{bmatrix} F & G & 0 \\ 0 & 0 & 0 \\ H & \theta & 0 \end{bmatrix}. \tag{2.15}$$

From [14], we get the following theorem:

Theorem. A system $\mathbf{S}$ is input reachable if and only if the binary matrix G has no zero rows.

With the notions of generic rank and input reachability, the following theorem states conditions for structural controllability of a system with structure $\left(\tilde{A}, \tilde{B}\right)$ based on the results developed in [14].

Theorem. A pair $\left(\tilde{A}, \tilde{B}\right)$ is structurally controllable if and only if the system $\mathbf{S}$ is input reachable and

$$\tilde{\rho}\left(\begin{bmatrix} \tilde{A} & \tilde{B} \end{bmatrix}\right) = n_s. \tag{2.16}$$

While the theorem above holds for any system, several assumptions about the structure of BAS systems further simplify the analysis. For this thesis, it is assumed that each decoupled subsystem $\mathbf{S}_i$ and $\mathbf{S}_0$ is structurally controllable. This is to say that each $\left(\bar{A}_{ii}, \bar{B}_{ii}\right) \forall i \in \mathcal{N}$ and $\left(\bar{A}_{00}, \bar{B}_{00}\right)$ is input reachable and full generic rank. It is important to note that the coupling terms in the last row and last column of the A and B matrices for a BAS system cannot prevent the structural controllability of the system $\mathbf{S}$ and, therefore, if each of the $N+1$ subsystems is structurally controllable, then the entire system is structurally controllable. As previously stated, structural controllability does not take into account the potential relationships between terms in the A and B matrices and, therefore, a system may not actually be controllable despite being structurally controllable. In Chapter 7, an algebraic constraint on the system causes such a

relationship between terms in the A and B matrices, thus motivating the need to modify the system representation.

2.4 Control of BAS Systems

With the BAS system $\mathbf{S}$ from (2.3) it is important to classify some generic control objectives as well as the information and actuator constraints typically found for this class of systems.

2.4.1 Control Objectives

As with any large-scale system, BAS systems have multiple control objectives which are classified here as either *local* or *global* objectives. Local objectives refer to control objectives at the subsystem level such as regulating a subsystem state to a desired value. Global objectives refer to control objectives that either rely on multiple subsystems or the entire system. Examples of global objectives are the regulation of the difference between the states of two different subsystems or the minimization of the power consumed by the entire system. For this work, it is also important to classify control objectives as either *performance* or *efficiency* objectives. Performance objectives, which are typically concerned with states or outputs, include the state regulation or tracking of a desired reference value. Performance objectives have a well-defined desired outcome where it is easy to discern whether the objective is being met. Alternatively, efficiency objectives, which are typically concerned with inputs, refer to the minimization (or maximization) of a value which does not have a clearly achievable desired value. The term efficiency is used for these types of objectives as they often correspond to the minimization of total power consumption of the system where zero may be the desired power consumption but this objective is clearly not obtainable due to conflicting performance objectives. Efficiency objectives are considerably more difficult to achieve and evaluate than performance objectives.

A generic cost function of the form

$$J = \gamma_a \left(\gamma_b J_p + (1-\gamma_b) J_u \right) + (1-\gamma_a) J_{\Delta u} \tag{2.17}$$

is used for all of the controllers throughout this work. Here J_p, J_u, and $J_{\Delta u}$ refer to costs associated with performance, control input, and changes in control input, respectively. For this

work, the control input cost J_u is considered an efficiency objective. It is assumed that each actuator has a cost associated with the magnitude of the control input and the sum of these costs is represented by J_u. Therefore, the objective is to meet the performance requirements represented by J_p while minimizing J_u. The weightings γ_a and γ_b are used to adjusted the relative importance between the different objectives, where γ_a is used to determine the tradeoff between changes in control actions and the objectives and γ_b is used to determine the tradeoffs between the performance and efficiency objectives. This cost function and the associated control objectives are made more concrete for an example system in Section 2.5 and the Model Predictive Control (MPC) algorithm used to minimize this cost function is detailed in Chapter 3.

2.4.2 Information Constraints

For large-scale systems, information constraints often restrict the types of control architectures which can be used to minimize the cost function (2.17). Assuming state-feedback control, *centralized* control architectures utilize complete state information and a complete system model when determining the control inputs to the entire system. It is well known that a centralized control approach to solving a multi-objective control problem results in a Pareto optimal solution [15], where it is impossible to make any term of the cost function smaller without making another term larger. Unfortunately, centralized control approaches to large-scale systems are typically infeasible. This is due to the large communication and computational costs associated with using information of all the system states and a complete system model in the minimization of the cost function for the system. Therefore, control of large-scale systems is often done in a decentralized manner. *Decentralized* control architectures utilize only local state information and a model of only the local subsystem to determine the local control efforts. This decentralization reduces the single large control problem into a set of smaller control problems. These control problems can be solved independently of one another, allowing the problems to be solved in parallel with reduced computational costs.

For large-scale BAS systems, information constraints can be categorized as either constraints on *communication* or *plant knowledge*. Communication constraints restrict the information, typically state or input values, that can be used to make control decisions by the

various controllers for the system. Plant knowledge constraints often arise in large-scale systems with multiple subsystems. While the dynamics of each subsystem may be well understood and accurately modeled, often it may be more difficult to develop an accurate model for the interactions between subsystems. Thus, some control architectures may be constrained to make control decisions with limited knowledge of the interconnection between subsystems in the plant. Chapter 5 develops several control architectures under various information constraints and the performance of these architectures is evaluated for an example system in Chapter 6.

2.4.3 Actuator, State, and Output Constraints

In addition to constraints on information, actuator, state, and output constraints are another key aspect of BAS systems which can significantly affect the control of these systems. In this work, all actuators with inputs u are constrained to have a minimum value $u_{\min}$ and a maximum value $u_{\max}$ such that

$$u_{\min} \le u(t) \le u_{\max} \quad \forall\, t \in [t_0, \infty], \tag{2.18}$$

where t_0 is the initial time. The BAS with constrained actuators presents an interesting control problem. The inputs u_0 of the common subsystem $\mathbf{S}_0$ have the ability to significantly affect the states of the each subsystem $\mathbf{S}_i$. If the inputs u_0 are chosen poorly, the ability for each subsystem $\mathbf{S}_i$ to meet its performance objectives can be compromised due to the constraints on the inputs u_i. The effects of actuator saturation are covered in greater detail and demonstrated for an example system in Chapter 6.

MPC provides the capability to constrain the states and outputs of the system in addition to constraining the actuator inputs. In practice, it is common to place minimum and maximum bounds on states and outputs instead of forcing them to track a desired value. These constraints are of the form

$$x_{\min} \le x(t) \le x_{\max} \quad \forall\, t \in [t_0, \infty], \tag{2.19}$$

$$y_{\min} \le y(t) \le y_{\max} \quad \forall\, t \in [t_0, \infty]. \tag{2.20}$$

Additionally, MPC is able to enforce algebraic relationships between states of the form

$$Ax(t) = b \quad \forall\ t \in [t_0, \infty]. \tag{2.21}$$

The equality relationship in (2.21) can also be relaxed to an inequality relationship such that

$$Ax(t) \leq b \quad \forall\ t \in [t_0, \infty]. \tag{2.22}$$

While these state and output constraints can be very useful in practice, this work will only consider the effects of actuator constraints in the control of BAS systems. The use of state and output constraints is left for future work.

2.5 Example System

Throughout the rest of this thesis, it is beneficial to use an example system to demonstrate the controller designs in Chapters 3 and 4 and the control architecture development and comparison in Chapters 5 and 6. This example system represents a system with a BAS and was designed to have many of the features found in real large-scale BAS systems such as unknown disturbances, actuator saturation, and performance and efficiency control objectives. In order to effectively convey the ideas in the following chapters, the system was also designed to be linear and relatively small, with only 3 subsystems $\mathbf{S}_i$ and one subsystem $\mathbf{S}_0$.

2.5.1 Example System Description

Fig. 2.2 shows the example system which is used throughout the chapters that follow. The system is represented by an electrical circuit with 3 subsystems $\mathbf{S}_i \ \ i \in \mathcal{N} = \{1,2,3\}$ and a common subsystem $\mathbf{S}_0$. Each subsystem $\mathbf{S}_i$ has two dynamic states V_i and V_{ai}, two capacitors C_i and C_{ai}, two variable resistances $\tilde{R}_i$ and $\tilde{R}_{ai}$ with actuator inputs u_i and u_{ai}, respectively, a fixed resistance R_{ei}, and a current sink I_{Li} which acts as a disturbance. The common subsystem $\mathbf{S}_0$ has two states V_g and V_e, two capacitors C_g and C_e, a variable resistance $\tilde{R}_g$ with actuator input u_g, a current source I_t with actuator input u_t, and a voltage source V_h which acts as a disturbance.

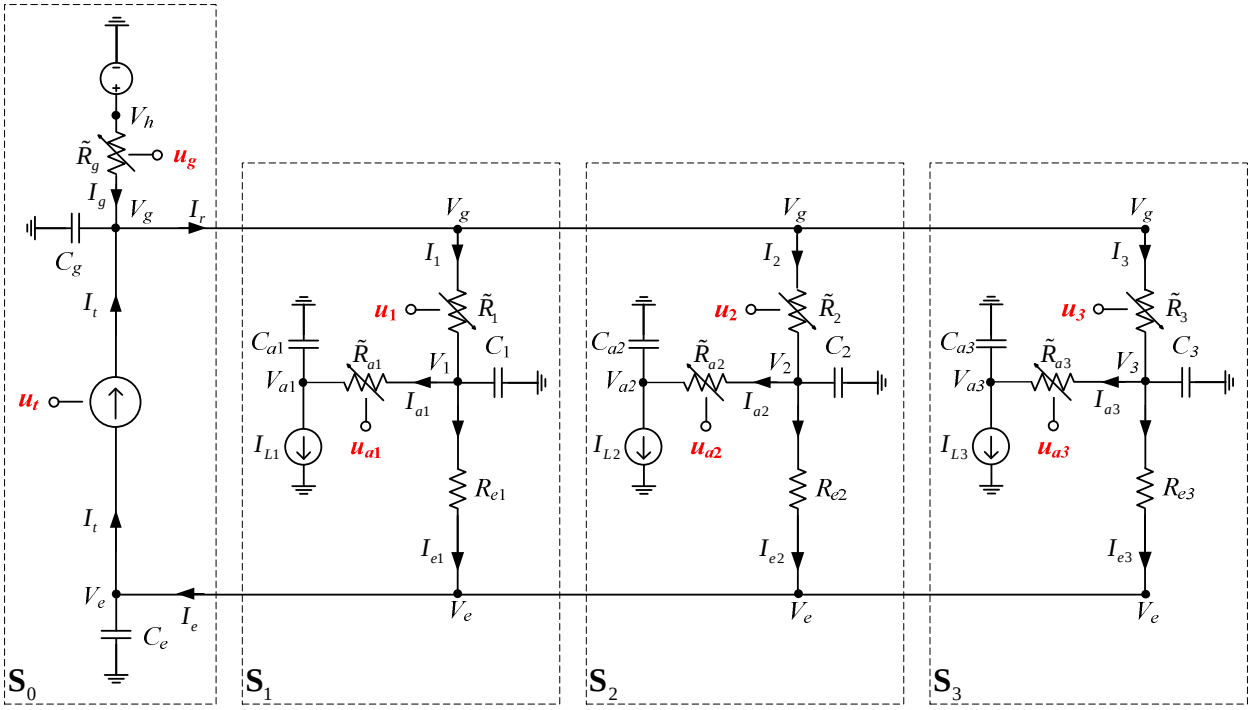

Figure 2.2 Electrical circuit diagram for example BAS system.

From the circuit diagram in Fig. 2.2, the following differential and algebraic equations represent the dynamics of the system and are used to develop a state-space model representation of the system. Each subsystem $\mathbf{S}_i$ has two differential equations

$$C_i \dot{V}_i = I_i - I_{ai} - I_{ei} \quad \forall\, i \in \mathcal{N}, \tag{2.23}$$

$$C_{ai} \dot{V}_{ai} = I_{ai} - I_{Li} \;\forall\, i \in \mathcal{N}. \tag{2.24}$$

The variable resistors $\tilde{R}_i$ and $\tilde{R}_{ai}$ are represented as

$$V_g - V_i = I_i \tilde{R}_i\left(u_i\right) \;\forall\, i \in \mathcal{N}, \tag{2.25}$$

$$V_i - V_{ai} = I_{ai} \tilde{R}_{ai}\left(u_{ai}\right) \forall\, i \in \mathcal{N}. \tag{2.26}$$

In order to make these equations linear, the voltage drop across each of the resistors is expressed as a linear combination of the current through the resistor and the control input, written as

$$V_g - V_i = I_i R_i - K_i u_i \quad \forall\, i \in \mathcal{N}, \tag{2.27}$$

$$V_i - V_{ai} = I_{ai}R_{ai} - K_{ai}u_{ai} \;\forall\, i \in \mathcal{N}, \tag{2.28}$$

where R_i, R_{ai}, K_i, and K_{ai} are positive constants of the linearization and R_i and R_{ai} can be thought of as nominal resistances. The resistor R_{ei} is simply expressed as

$$V_i - V_e = I_{ei}R_{ei} \;\forall\, i \in \mathcal{N}. \tag{2.29}$$

Arranging (2.23) – (2.29) into a subsystem state-space form yields

$$\begin{aligned}\mathbf{S}_i : \begin{bmatrix}\dot{V}_i \\ \dot{V}_{ai}\end{bmatrix} = &\begin{bmatrix} -\dfrac{\bar{R}_i}{C_i} & \dfrac{1}{C_i R_{ai}} \\ \dfrac{1}{C_{ai}R_{ai}} & -\dfrac{1}{C_{ai}R_{ai}}\end{bmatrix}\begin{bmatrix}V_i \\ V_{ai}\end{bmatrix} + \begin{bmatrix}\dfrac{1}{C_i R_i} & \dfrac{1}{C_i R_{ei}} \\ 0 & 0\end{bmatrix}\begin{bmatrix}V_g \\ V_e\end{bmatrix} + \\ &+\begin{bmatrix}\dfrac{K_i}{C_i R_i} & -\dfrac{K_{ai}}{C_i R_{ai}} \\ 0 & \dfrac{K_{ai}}{C_{ai}R_{ai}}\end{bmatrix}\begin{bmatrix}u_1 \\ u_{ai}\end{bmatrix} + \begin{bmatrix}0 \\ -\dfrac{1}{C_{ai}}\end{bmatrix}\begin{bmatrix}I_{Li}\end{bmatrix} \quad \forall\, i \in \mathcal{N},\end{aligned} \tag{2.30}$$

where $\bar{R}_i = \dfrac{1}{R_i} + \dfrac{1}{R_{ai}} + \dfrac{1}{R_{ei}}$, and is written more compactly as

$$\mathbf{S}_i : \dot{x}_i = A_{ii}x_i + A_{i0}x_0 + B_{ii}u_i + V_{ii}d_i \;\forall\, i \in \mathcal{N}. \tag{2.31}$$

The common subsystem $\mathbf{S}_0$ has two differential equations

$$C_g\dot{V}_g = I_t + I_g - I_r, \tag{2.32}$$

$$C_e\dot{V}_e = I_e - I_t. \tag{2.33}$$

The variable resistor $\tilde{R}_g$ is represented as

$$V_h - V_g = I_g\tilde{R}_g, \tag{2.34}$$

with the linearization

$$V_h - V_g = I_g R_g - K_g u_g, \tag{2.35}$$

where R_g and K_g are positive constants and R_g can be thought of as the nominal resistance. The current source is given by

$$I_t = \alpha_1 V_g + \alpha_2 V_e + \alpha_3 u_t, \tag{2.36}$$

where α_1, α_2, and α_3 are positive coefficients. Note that this is not an ideal current source, as the current depends on the voltages V_g and V_e. Finally, from Kirchhoff's current law, we have

$$I_e = \sum_{i=1}^{n} I_{ei}, \tag{2.37}$$

$$I_r = \sum_{i=1}^{n} I_i. \tag{2.38}$$

Arranging (2.32) – (2.38) into a subsystem state-space form yields

$$\begin{aligned} \mathbf{S}_0 : \begin{bmatrix} \dot{V}_g \\ \dot{V}_e \end{bmatrix} = & \begin{bmatrix} -\frac{1}{C_g}\left(\frac{1}{R_g} + \sum_{i=1}^{3}\frac{1}{R_i} - \alpha_1\right) & \frac{\alpha_2}{C_g} \\ -\frac{\alpha_1}{C_e} & -\frac{1}{C_e}\left(\sum_{i=1}^{3}\frac{1}{R_{ei}} + \alpha_2\right) \end{bmatrix}\begin{bmatrix} V_g \\ V_e \end{bmatrix} + \sum_{i=1}^{3}\begin{bmatrix} \frac{1}{C_g R_i} & 0 \\ \frac{1}{C_e R_{ei}} & 0 \end{bmatrix}\begin{bmatrix} V_i \\ V_{ai} \end{bmatrix} + \\ & + \begin{bmatrix} \frac{\alpha_3}{C_g} & \frac{K_g}{C_g R_g} \\ -\frac{\alpha_3}{C_e} & 0 \end{bmatrix}\begin{bmatrix} u_t \\ u_g \end{bmatrix} + \sum_{i=1}^{3}\begin{bmatrix} -\frac{K_i}{C_g R_i} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} u_i \\ u_{ai} \end{bmatrix} + \begin{bmatrix} \frac{1}{C_g R_g} \\ 0 \end{bmatrix} V_h, \end{aligned} \tag{2.39}$$

and is written more compactly as

$$\mathbf{S}_0 : \dot{x}_0 = A_{00} x_0 + \sum_{i=1}^{3} A_{0i} x_i + B_{00} u_0 + \sum_{i=1}^{3} B_{0i} u_i + V_{00} d_0. \tag{2.40}$$

Combining (2.31) and (2.40), the complete system is

$$\mathbf{S}:\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_0\end{bmatrix}=\begin{bmatrix}A_{11} & 0 & 0 & A_{10}\\ 0 & A_{22} & 0 & A_{20}\\ 0 & 0 & A_{33} & A_{30}\\ A_{01} & A_{02} & A_{03} & A_{00}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_0\end{bmatrix}+\begin{bmatrix}B_{11} & 0 & 0 & 0\\ 0 & B_{22} & 0 & 0\\ 0 & 0 & B_{33} & 0\\ B_{01} & B_{02} & B_{03} & B_{00}\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\\ u_0\end{bmatrix}+\begin{bmatrix}V_{11} & 0 & 0 & 0\\ 0 & V_{22} & 0 & 0\\ 0 & 0 & V_{33} & 0\\ 0 & 0 & 0 & V_{00}\end{bmatrix}\begin{bmatrix}d_1\\ d_2\\ d_3\\ d_0\end{bmatrix}, \tag{2.41}$$

or

$$\mathbf{S}:\dot{x}=Ax+Bu+Vd. \tag{2.42}$$

2.5.2 Example System Control Objectives

The example system was designed to have control objectives representative of the type of objectives found in many large-scale BAS systems. These objectives consist of local and global performance objectives for state reference tracking and a global efficiency objective. The local performance objective $J_{p,i}$ is used to have the state V_{ai} track a desired reference r_{ai} for each subsystem $\mathbf{S}_i$, which is expressed as

$$J_{p,i}=\left(V_{ai}-r_{ai}\right)^2. \tag{2.43}$$

The global performance objective is to track a desired reference r_{Ie} for the current I_e. From (2.29) and (2.37), it is clear that the current I_e can be expressed as

$$I_e=\sum_{i=1}^{3}\frac{V_i-V_e}{R_{ei}}. \tag{2.44}$$

Since I_e is a function of states and parameters from multiple subsystems, the control of I_e is a global performance objective $J_{p,\mathrm{g}}$, written as

$$J_{p,\mathrm{g}}=\left(I_e-r_{Ie}\right)^2. \tag{2.45}$$

In addition to the performance objectives, each actuator is assumed to consume a resource κ_j (e.g. energy) as a quadratic function of the actuator input u_j,

$$\kappa_j = a_j u_j^{\,2} + b_j u_j + c_j. \tag{2.46}$$

The global efficiency objective J_u is to minimize the sum of the resources consumed by all actuators

$$J_u = \sum_{j=1}^{n_u} \kappa_j, \tag{2.47}$$

where n_u is the number of actuators in the system and $n_u = 8$ for the example system. The efficiency objective is a global objective because attempting to minimize the resource consumption for each actuator individually does not result in the lowest resource consumption possible due to the coupling between subsystems. This idea is further explained and demonstrated in the decentralized control section 5.2.

An addition objective $J_{\Delta u}$ is used to penalize changes in the control inputs in time. This objective is necessary to prevent the actuators from changing too rapidly, which can cause instability. Finally, these control objectives can be combined resulting in the control problem

$$\begin{aligned}
\underset{u}{\text{minimize}} \quad & J = \int_{t=t_0}^{T} \gamma_a \left(\gamma_b \left(\sum_{i=1}^{3} J_{p,i}(t) + J_{p,g}(t) \right) + (1-\gamma_b) \sum_{j=1}^{8} \kappa_j(t) \right) + (1-\gamma_a) J_{\Delta u}, \\
\text{subject to} \quad & u(t) \in \mathcal{U}, \;\; x(t) \in \mathcal{X}, \;\; t \in [t_0, T], \\
& \dot{x}(t) = Ax(t) + Bu(t) + Vd(t), \;\; t \in [t_0, T], \\
& x(t_0) = x_0 \in \mathcal{X},
\end{aligned} \tag{2.48}$$

where t_0 and T are the initial and final times, $\mathcal{U}$ and $\mathcal{X}$ are sets of admissible control input and state values, and x_0 is the initial state at time t_0. For the example system, the actuator inputs are constrained but the states are not, therefore $\mathcal{X} = \mathbb{R}^8$. This control problem is augmented into the form used for Model Predictive Control (MPC) in Chapter 3.

Chapter 3
Model Predictive Control (MPC)

Model Predictive Control (MPC) is a receding-horizon optimal control framework which uses a dynamic model of a system to predict the future response of the system. By solving a finite-time horizon, open-loop, optimal control problem using the current state of the system, MPC determines a sequence of control decisions which minimize the specified cost function over the prediction horizon. The first element of this control sequence is applied to the system and the procedure is repeated at the next time instance. Aspects such as stability and robustness are thoroughly developed in the literature [5]. MPC has been widely adopted in industry due to its ability to explicitly consider hard constraints on the inputs, states, and outputs of the system. Additional features that make MPC attractive for control applications are the abilities to perform multi-input multi-output (MIMO) control, utilize a wide range of cost functions, and predict the future state of the system. For this work, linear MPC is used as opposed to nonlinear MPC. While most practical systems are nonlinear, linear MPC requires a linear representation of the system. Despite the fact that the linear approximation of the nonlinear system is only valid for a limited range of operating conditions, linear MPC has significantly lower computational costs than nonlinear MPC. For additional details on nonlinear MPC please refer to [16].

Fig. 3.1 demonstrates the MPC process for a given time instance. Starting at the current sample k, the finite-time horizon consists of N_p time samples with the time between samples Δt. The number of time samples N_p is known as the prediction horizon and denotes the number of future time steps for which the system states are predicted. Similarly, the control

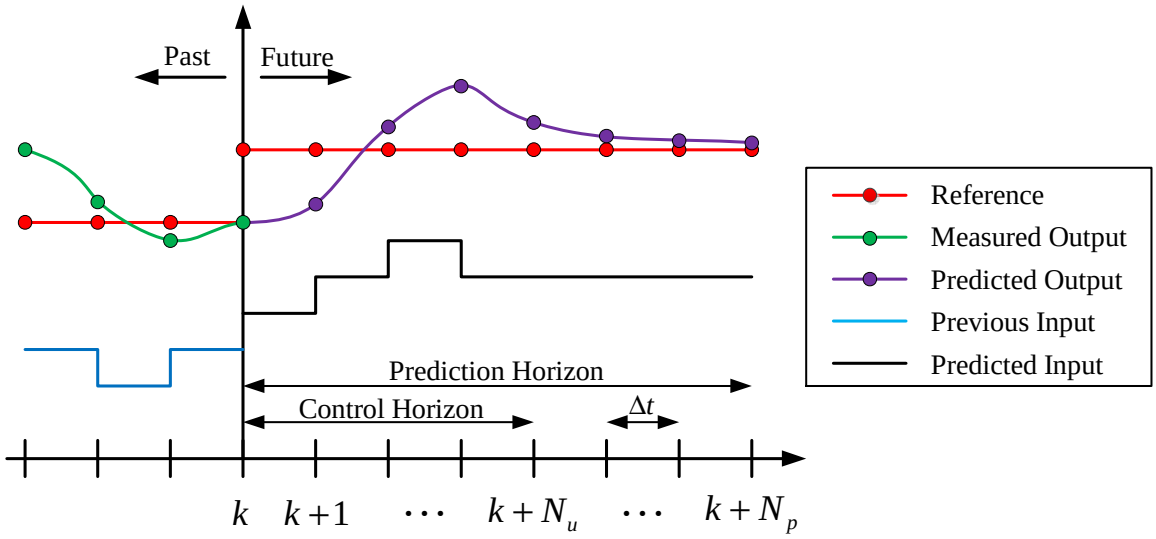

Figure 3.1 Model predictive control.

horizon N_u denotes the number of future time steps for which control decisions are determined by solving the optimization problem. Note that $N_u \le N_p$ and if $N_u < N_p$ we have

$$u(k+N_u) = u(k+N_u+1) = \ldots = u(k+N_p). \tag{3.1}$$

While MPC provides the flexibility to consider references and disturbances that change over the prediction horizon, as presented in [17], this work considers references and disturbances to be constant. Section 3.1 details the MPC formulation used for the various control strategies presented throughout this thesis and Section 3.2 demonstrates how this formulation is used to develop the controllers for the example system from Chapter 2.

3.1 Basic MPC Formulation

MPC uses a discrete system model of the form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Vd(k), \\ y(k) &= Cx(k) + Du(k) + Wd(k), \end{aligned} \tag{3.2}$$

where $x \in \mathbb{R}^{n_s}$, $y \in \mathbb{R}^{n_y}$, $u \in \mathbb{R}^{n_u}$, and $d \in \mathbb{R}^{n_d}$ with matrices A, B, C, D, V, W of the appropriate sizes. For simplicity, the D and W matrices are considered to be zero for this work. The generic MPC formulation solves the control problem

$$
\begin{aligned}
\underset{U}{\text{minimize}} \quad & J = \sum_{j=1}^{N_p} f\left(x(k), x(k+j), y(k+j), u(k+j-1)\right), \\
\text{subject to} \quad & u(k+j-1) \in \mathcal{U}, \quad x(k+j) \in \mathcal{X}, \quad j \in \left[1, N_p\right], \\
& x(k+j) = Ax(k+j-1) + Bu(k+j-1) + Vd(k+j-1), \quad j \in \left[1, N_p\right], \\
& y(k+j) = Cx(k+j), \quad j \in \left[1, N_p\right], \\
& x(k) = x_0 \in \mathcal{X},
\end{aligned} \tag{3.3}
$$

where U is the set of control inputs over the control horizon, $\mathcal{U}$ and $\mathcal{X}$ are sets of admissible control input and state values, and x_0 is the initial state at sample k. The cost f, at each time $k+j$, is composed of costs associated with performance J_p, efficiency J_u, and changes in control input actuation $J_{\Delta u}$. Additionally, each of these terms is the sum of costs from each subsystem $\mathbf{S}_i$ and $\mathbf{S}_0$. Therefore, the cost function from (3.3) can be expressed as

$$
J = \sum_{j=1}^{N_p} \left[\sum_{i=0}^{N} \left(\gamma_a \left(\gamma_b J_{p,i} + (1-\gamma_b) J_{u,i} \right) + (1-\gamma_a) J_{\Delta u,i} \right) \right], \tag{3.4}
$$

where γ_a and γ_b are weightings used to assign the relative importance between the objectives and control inputs and between the performance and efficiency objectives.

When solving this control problem, it is common to rewrite the cost function as only a function of the initial state $x(k)$ and the control inputs over the control horizon $u(k+l)$, where $l \in \left[0, N_u - 1\right]$. Thus, the discrete model (3.2) is used to express all future states and outputs of the system as a function of the initial state at sample k and the control inputs U over the control horizon

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + Vd, \\
x(k+2) &= A^2 x(k) + ABu(k) + Bu(k+1) + (A+I)Vd, \\
&\vdots \\
x(k+N_u) &= A^{N_u} x(k) + A^{N_u-1} Bu(k) + \ldots + Bu(k+N_u-1) + \left(A^{N_u-1} + \ldots + A + 1\right)Vd, \\
&\vdots \\
x(k+N_p) &= A^{N_p} x(k) + A^{N_p-1} Bu(k) + \ldots + A^{N_p-N_u} Bu(k+N_u-1) + \left(A^{N_p-1} + \ldots + A + 1\right)Vd.
\end{aligned} \tag{3.5}
$$

Note that the disturbance d is assumed to be constant over the entire prediction horizon. From (3.5), the system response over the prediction horizon can be expressed in a lifted form as

$$X = Tx(k) + SU + Rd, \tag{3.6}$$

where

$$X = \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N_p) \end{bmatrix} \in \mathbb{R}^{(n_s \cdot N_p)}, \quad U = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_u-1) \end{bmatrix} \in \mathbb{R}^{(n_u \cdot N_u)}, \tag{3.7}$$

$$T = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix} \in \mathbb{R}^{(n_s \cdot N_p) \times (n_s)}, \quad R = \begin{bmatrix} V \\ (A+1)V \\ \vdots \\ \left(A^{N_p-1} + \ldots + A + 1\right)V \end{bmatrix} \in \mathbb{R}^{(n_s \cdot N_p) \times (n_d)}, \tag{3.8}$$

$$S = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & B & 0 \\ A^{N_u-1}B & \cdots & AB & B \\ \vdots & \ddots & \ddots & \vdots \\ A^{N_p-2}B & \ddots & \ddots & A^{N_p-N_u-1}B \\ A^{N_p-1}B & \cdots & A^{N_p-N_u-1}B & A^{N_p-N_u}B \end{bmatrix} \in \mathbb{R}^{(n_s \cdot N_p) \times (n_u \cdot N_u)}. \tag{3.9}$$

The outputs of the system can be expressed as

$$Y = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_p) \end{bmatrix} = PX, \tag{3.10}$$

where $Y \in \mathbb{R}^{(n_y \cdot N_p)}$, $P \in \mathbb{R}^{(n_y \cdot N_p) \times (n_x \cdot N_p)}$, and

$$P = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & C \end{bmatrix}. \tag{3.11}$$

Now the cost function from (3.3) can be expressed as

$$J = U^T HU + F^T U, \tag{3.12}$$

where $H \in \mathbb{R}^{(n_u \cdot N_u)\times(n_u \cdot N_u)}$, $F \in \mathbb{R}^{(n_u \cdot N_u)}$ are functions of T, S, R, P, $x(k)$, and d. In addition to minimizing this cost function, the solution must satisfy the constraints on the actuators

$$U_{\min} \le U \le U_{\max}, \tag{3.13}$$

and constraints on the states

$$X_{\min} \le X \le X_{\max}. \tag{3.14}$$

The constraints on the states can be converted to constraints on the inputs using (3.6)

$$X_{\min} - Tx(k) - Rd \le SU \le X_{\max} - Tx(k) - Rd \tag{3.15}$$

A similar transformation can be done for constraints on the outputs.

3.2 MPC Formulation for Example System

Now we develop the MPC framework based on the steps presented in the previous section. With the continuous system model (2.42), restated here as

$$\mathbf{S}: \dot{x} = Ax + Bu + Vd, \tag{3.16}$$

the first step is to define the outputs of the system. Based on the control objectives presented in Section 2.5.2, it is valuable to have I_e be an output of the system. Therefore, with the state vector $x = \begin{bmatrix} V_1 & V_{a1} & V_2 & V_{a2} & V_3 & V_{a3} & V_g & V_e \end{bmatrix}^T$ and the output vector $y = \begin{bmatrix} V_1 & V_{a1} & V_2 & V_{a2} & V_3 & V_{a3} & V_g & I_e \end{bmatrix}^T$, the C matrix is

$$C = \left[\begin{array}{cccccccc|c} & & & & & & & & \\ & & & & I_{7\times 7} & & & & 0_{7\times 1} \\ & & & & & & & & \\ \hline \frac{1}{R_{e1}} & 0 & \frac{1}{R_{e2}} & 0 & \frac{1}{R_{e3}} & 0 & 0 & & -\left(\frac{1}{R_{e1}}+\frac{1}{R_{e2}}+\frac{1}{R_{e3}}\right) \end{array}\right]. \tag{3.17}$$

Next, in order to achieve perfect reference tracking for the desired outputs, it is typical to convert the states into tracking error states. Often, however, only a subset of the states have a desired value, which are defined as $\tilde{y} \in \mathbb{R}^{n_r}$, and the matrix $M \in \mathbb{R}^{n_r \times n_y}$ is used to isolate these states (i.e. $\tilde{y} = My$). For the example system, the states V_{ai}, $i = 1,2,3$, and I_e have desired values, resulting in

$$M = \text{blkdiag}\left(\begin{bmatrix}0 & 1\end{bmatrix},\begin{bmatrix}0 & 1\end{bmatrix},\begin{bmatrix}0 & 1\end{bmatrix},\begin{bmatrix}0 & 1\end{bmatrix}\right). \tag{3.18}$$

The error states are defined as

$$e = y - M^{-1}r, \tag{3.19}$$

where M^{-1} denotes a pseudo-inverse since M is rarely invertible. With the references r assumed to be constant, the error system model $\hat{\mathbf{S}}$ can be written as

$$\hat{\mathbf{S}}: \dot{e} = CAC^{-1}e + CBu + \begin{bmatrix} CV & CAC^{-1}M^{-1} \end{bmatrix}\begin{bmatrix} d \\ r \end{bmatrix}, \tag{3.20}$$

or

$$\hat{\mathbf{S}}: \dot{e} = \hat{A}e + \hat{B}u + \hat{V}\hat{d}. \tag{3.21}$$

Now, we discretize the system with a sample time of Δt resulting in the discrete model $\mathbf{S}_d$

$$\mathbf{S}_d : e(k+1) = A_d e(k) + B_d u(k) + V_d d(k). \tag{3.22}$$

In addition to having error states, perfect reference tracking also requires integral states z. The system in (3.22) is further augmented to include these integral states

$$\hat{\mathbf{S}}_d : \begin{bmatrix} e(k+1) \\ z(k+1) \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ M \cdot \Delta t & I \end{bmatrix}\begin{bmatrix} e(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} V_d \\ 0 \end{bmatrix} d(k). \tag{3.23}$$

As mentioned in Section 2.5.2, when designing a controller to achieve reference tracking (as opposed to stabilizing an equilibrium) it is often valuable to penalize changes in the control inputs in time instead of penalizing the magnitude of the control input. Thus, through a final augmentation, the system (3.23) is transformed to have inputs in terms of $\Delta u(k)$ where $u(k) = u(k-1) + \Delta u(k)$. Defining an additional set of states as $x_u(k) = u(k-1)$, we have

$$\begin{aligned} \overline{\mathbf{S}}_d : \begin{bmatrix} e(k+1) \\ z(k+1) \\ x_u(k+1) \end{bmatrix} &= \begin{bmatrix} A_d & 0 & B_d \\ M \cdot \Delta t & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} e(k) \\ z(k) \\ x_u(k) \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \\ I \end{bmatrix} \Delta u(k) + \begin{bmatrix} V_d \\ 0 \\ 0 \end{bmatrix} d(k), \\ \overline{y}(t) &= \begin{bmatrix} M & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} e(k) \\ z(k) \\ x_u(k) \end{bmatrix}, \end{aligned} \tag{3.24}$$

or

$$\begin{aligned} \overline{\mathbf{S}}_d : \overline{x}(k+1) &= \overline{A}\overline{x}(k) + \overline{B}\Delta u(k) + \overline{V}d(k), \\ \overline{y}(t) &= \overline{C}\overline{x}(k). \end{aligned} \tag{3.25}$$

Note that $\overline{y}$ only contains the error states which have desired values and the integral states of those errors. As was done in (3.6), the system response over the prediction horizon can be expressed in lifted form as

$$\begin{aligned} \overline{X} &= \overline{T}\overline{x}(k) + \overline{S}\Delta U + \overline{R}d, \\ \overline{Y} &= \overline{P}\overline{X} \end{aligned} \tag{3.26}$$

Now the cost function for the example system from (2.48) can be written in the lifted system representation as

$$J = \overline{Y}^T Q_1 \overline{Y} + \Delta U^T Q_2 \Delta U + K, \tag{3.27}$$

where $Q_1 \in \mathbb{R}^{(2n_r \cdot N_p) \times (2n_r \cdot N_p)}$ is a diagonal matrix containing the weightings for the reference tracking error and integral states and $Q_2 \in \mathbb{R}^{(n_u \cdot N_u) \times (n_u \cdot N_u)}$ is a diagonal matrix for the weightings on the changes in control inputs. In (3.27), K represents the cost associated to the resource

consumption of each actuator. As stated in (2.46), these actuator costs are quadratic functions of the actuator inputs, rewritten here as

$$\kappa_j = a_j u_j{}^2 + b_j u_j + c_j. \tag{3.28}$$

Often, system models and controllers are developed about a nominal operating condition and the control inputs determined by the controller are deviations δu from the nominal actuator input u^0, where the superscript denotes a nominal, not a power. Therefore, it is important to be able to express the actuator costs κ_j as a function of a_j, b_j, c_j, u_j^0, and δu_j. Through a simple change of coordinates the actuator costs in (3.28) can be written as

$$\begin{aligned} \kappa_j &= a_j \delta u_j{}^2 + \left(2a_j u_j^0 + b_j\right)\delta u_j + \left(a\left(u_j^0\right)^2 + b u_j^0 + c_j\right), \\ \kappa_j &= \bar{a}_j \delta u_j{}^2 + \bar{b}_j \delta u_j + \bar{c}_j. \end{aligned} \tag{3.29}$$

Note that from here on the deviation input, δu, is written as u to simplify the notation. Also, it is important to remember that δu is a deviation from a nominal condition, whereas Δu is a change in the input from one sample time to the next.

Additionally, since these actuator costs are functions of the magnitude of the actuators inputs U over the prediction horizon and not the changes in the inputs ΔU, the magnitudes of the inputs are written as functions of the changes in the inputs and the magnitude of the input at the current sample time $u(k)$ as

$$U = \bar{N}\Delta U + \bar{n}u(k), \tag{3.30}$$

where $\bar{N} \in \mathbb{R}^{(n_u \cdot N_p)\times(n_u \cdot N_u)}$, $\bar{n} \in \mathbb{R}^{(n_u \cdot N_p)\times n_u}$, and

$$\bar{N} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ I & \cdots & I & I \\ \vdots & \ddots & \vdots & \vdots \\ I & \cdots & I & I \end{bmatrix}, \quad \bar{n} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.31}$$

The actuator resource consumption costs K can be written as

$$K = U^T Q_a U + q_b^T U, \tag{3.32}$$

where $Q_a \in \mathbb{R}^{(n_u \cdot N_p)\times(n_u \cdot N_p)}$ is a diagonal matrix containing the $\overline{a}_j$ terms from (3.29) and $q_b \in \mathbb{R}^{(n_u \cdot N_p)}$ is a vector containing the $\overline{b}_j$ terms. Note that the $\overline{c}_j$ terms, while they do affect the cost, do not influence the optimal control sequence, and can be omitted moving forward.

It is beneficial to rewrite the cost function (3.27) as a quadratic function of the control decisions. This allows the optimization problem to be solved very quickly using the *quadprog* function in the MATLAB Optimization Toolbox [18]. Using (3.26), (3.30), and (3.32), the lifted cost function (3.27) can be written as a quadratic function of ΔU

$$J = \Delta U^T H \Delta U + F^T \Delta U, \tag{3.33}$$

where

$$\begin{aligned} H &= \overline{S}^T \overline{P}^T Q_1 \overline{P}\overline{S} + Q_2 + \overline{N}^T Q_a \overline{N}, \\ F &= 2\overline{S}^T \overline{P}^T Q_1 \overline{P}\overline{T}\overline{x}(k) + 2\overline{S}^T \overline{P}^T Q_1 \overline{P}\overline{R}d + 2\overline{N}^T Q_a \overline{n}u(k) + \overline{N}^T q_b. \end{aligned} \tag{3.34}$$

Since actuator constraints act on U and not ΔU, using (3.30), actuator constraints of the form

$$U_{\min} \le U \le U_{\max}, \tag{3.35}$$

can be written as constraints on ΔU as

$$U_{\min} - \overline{n}u(k) \le \overline{N}\Delta U \le U_{\max} - \overline{n}u(k). \tag{3.36}$$

This MPC formulation for the example system utilizes the entire model to determine the control decisions for each actuator in the system and is used for the centralized control approach in Chapter 5. Additionally, in Chapter 5, the same procedure is used to develop MPC controllers for several decentralized control strategies using alternative representations of the system.

MPC can be very effective when an accurate model of the system is available. However, the presence of unknown or unmodeled disturbances and unmodeled system nonlinearity may significantly degrade the performance of MPC. The use of integrator states, as detailed above, is commonly used to overcome model inaccuracy and can be successfully used to meet the performance objectives for the system. However, model inaccuracy can also cause model-based control strategies to operate the system away from the most efficient operating conditions, thus

degrading the ability of the controller to meet the efficiency objective for the system. The following chapter presents a model-free control strategy known as extremum seeking control (ESC), which can be used in conjunction with MPC to provide greater system efficiencies, especially in the presence of unknown disturbances and system nonlinearity.

Chapter 4

Extremum Seeking Control

Extremum seeking control (ESC) is an adaptive feedback control algorithm used to maximize (minimize) a system output y by driving a system input u to an optimal value u^*, while utilizing very little information about the system. With the first rigorous stability proof in [19], ESC has become widely used in a variety of applications including thermoacoustic oscillations [20], wind turbines [21], and PID controller tuning [22]. This work focuses on single-input single-output (SISO) gradient-based ESC which is developed in Section 4.1. Section 4.2 outlines the selection process for the parameter used by the ESC algorithm. Alternative ESC formulations, along with various augmentations to the standard ESC algorithm, are presented in Section 4.3 and a few of these augmentations are used in the control of the example system as shown in Section 4.4.

4.1 Basic ESC Formulation

Based on the analysis from [19], consider a general nonlinear plant of the form,

$$\begin{aligned} \dot{x} &= f(x,u), \\ y &= h(x), \end{aligned} \tag{4.1}$$

where $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}$, and $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$ are smooth. Using a smooth control law, $u = \alpha(x,\theta)$, the closed loop system becomes,

$$\dot{x} = f\left(x, \alpha(x,\theta)\right). \tag{4.2}$$

It is assumed that there exists a function $l:\mathbb{R}\to\mathbb{R}^n$ such that $f\left(x,\alpha\left(x,\theta\right)\right)=0$ if and only if $x=l\left(\theta\right)$. Additionally, for each $\theta\in\mathbb{R}$, the equilibrium $x=l\left(\theta\right)$ is locally exponentially stable. It is also assumed that there exists a unique $\theta^*\in\mathbb{R}$ such that

$$\begin{aligned}\left(h\circ l\right)'\left(\theta^*\right)&=0,\\ \left(h\circ l\right)''\left(\theta^*\right)&<0.\end{aligned}\tag{4.3}$$

Thus, our objective is to drive θ to θ^* in order to maximize $y=h\left(l\left(\theta\right)\right)$ without knowledge of θ^*, f, h, or l.

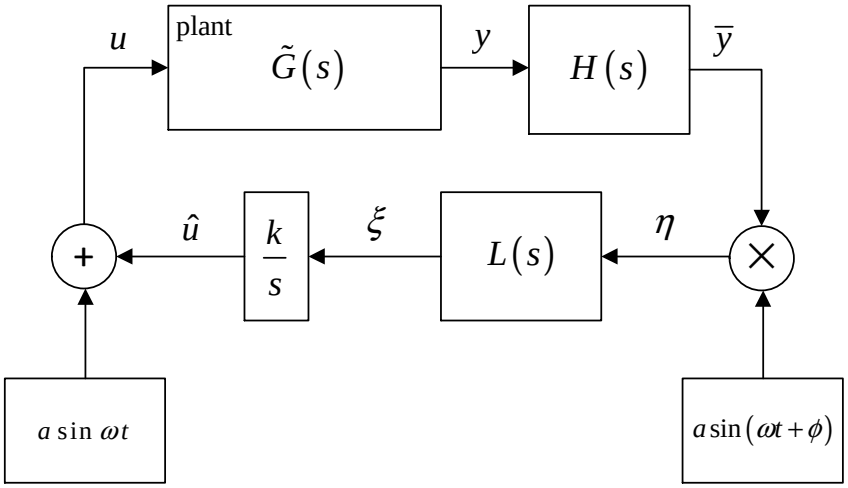

Figure 4.1. ESC schematic.

Fig. 4.1 shows a schematic of a basic gradient-based extremum seeking controller. The plant is represented by a transfer function $\tilde{G}(s)$, where the tilde denotes the fact that this transfer function changes as a function of the input u and disturbances to the system. It is assumed that $\tilde{G}(s)$ changes in such a way as to preserve a convex relationship between u and y, allowing information about the gradient to be used to drive $u\to u^*$. The following algorithm outlines in greater detail how the various signals in Fig. 4.1 are calculated and what they represent.

Gradient-based ESC Algorithm:

1. With the input

$$u=\hat{u}+a\sin\omega t,\tag{4.4}$$

the objective is to drive $\hat{u} \to u^*$, where ω is the perturbation frequency and a is the perturbation amplitude.

2. Given input u, the plant output y can be considered of the form

$$y = \hat{y} + b\sin\left(\omega t + \phi_p\right) + \gamma, \tag{4.5}$$

where:

(i) $\hat{y}$ changes with $\hat{u}$ and the plant dynamics, but is assumed to be changing slowly with respect to ω,

(ii) $b = aG_p\left(u, \omega\right)$ where G_p is the plant gain which is a function of the input u and ω,

(iii) ϕ_p is a phase shift caused by the plant dynamics which depends on ω, and

(iv) γ is the noise in the measurement (which is negligible for the simulation studies in this thesis).

3. Note: For a static plant, $\phi_p = 0$ if $\hat{u} < u^*$ and $\phi = 180$ if $\hat{u} > u^*$. However, with a dynamic plant, this may not be the case.

4. The output y is passed through a high-pass filter $H\left(s\right)$ to remove $\hat{y}$. The resulting signal is

$$\overline{y} = hb\sin\left(\omega t + \phi_p + \phi_H\right), \tag{4.6}$$

where $h = mag\left(H\left(j\omega\right)\right)$ and $\phi_H = phase\left(H\left(j\omega\right)\right)$.

5. Multiplying $\overline{y}$ by the demodulation signal $a\sin\left(\omega t + \phi\right)$ produces

$$\begin{aligned} \eta &= ahb\sin\left(\omega t + \phi\right)\sin\left(\omega t + \phi_p + \phi_H\right) \\ &= \frac{ahb}{2}\left[\cos\left(\phi_p + \phi_H - \phi\right) - \cos\left(2\omega t + \phi_p + \phi_H + \phi\right)\right]. \end{aligned} \tag{4.7}$$

6. The signal η is passed through the low-pass filter $L\left(s\right)$ which is designed to attenuate the $\cos\left(2\omega t + \phi_p + \phi_H + \phi\right)$ term. Thus $\xi \approx \frac{ahb}{2}\cos\left(\phi_p + \phi_H - \phi\right)$.

7. The signal ξ is scaled by k and integrated to produce $\hat{u}$.

From this algorithm, it is clear that $\hat{u}$ will increase if $\xi > 0$ which occurs when $\cos\left(\phi_p + \phi_H - \phi\right) > 0$. Similarly, $\hat{u}$ will decrease if $\xi, \cos\left(\phi_p + \phi_H - \phi\right) < 0$. When $\phi_p + \phi_H - \phi = 90^o$, $\cos\left(\phi_p + \phi_H - \phi\right) = 0$ and $\hat{u}$ will remain constant. Ideally, $\phi_p + \phi_H - \phi = 90^o$ when $\hat{u} = u^*$, however, the ESC parameters must be tuned correctly for this to occur.

4.2 ESC Parameter Selection

When developing an ESC controller, the performance of the controller is significantly affected by the choice of the perturbation frequency ω, perturbation amplitude a, scaling factor k, phase shift ϕ, and the high and low-pass filters, $H(s)$ and $L(s)$. First, the amplitude a must be chosen so that b, from (4.5), is distinguishable above the noise γ. Additionally, the larger the amplitude, the faster $\hat{u}$ can be driven to u^*. However, a large amplitude will also cause large perturbations in u once $\hat{u}$ gets close to u^*. If desired, the perturbation signal amplitude can be adapted as presented in [20]. The amplitude can be made a function of ξ such that when ξ is large, a is large, allowing for faster converges to u^*. When ξ is small, indicating $\hat{u}$ is close to u^*, a can be made small so that u does not deviate far from u^*.

The perturbation frequency can greatly affect the convergence of the ESC controller. While ω needs to be significantly slower than the dynamics of the system for stability purposes, see [19], in [23] it was found that the choice of ω can also effect the value to which $\hat{u}$ converges. If ω is poorly chosen, typically if ω is too fast, $\hat{u}$ may converge to a point very far from u^*. From the analysis above, this means that $\phi_p + \phi_H \neq 90^o$ when $\hat{u} = u^*$. Fortunately, the demodulation signal can be phase shifted by ϕ to compensate such that $\phi_p + \phi_H - \phi = 90^o$ when $\hat{u} = u^*$. Therefore, the choice of ω and ϕ need to be coordinated.

The choice of the scaling factor k directly affects the convergence rate of the algorithm. In order to insure stability, k needs to be relatively small [19], but large enough to achieve an acceptable convergence rate. If increasing k cannot provide the desired convergence rate, alternative approaches such as the use of dynamic compensators [24] and Newton-based methods [20] can be used to improve stability and achieve faster convergence.

Finally, the choice of the high- and low-pass filters is also of importance. The high-pass filter is used to remove the slowly changing $\hat{y}$ component of the output signal from (4.5). Therefore, the output of the filter $\bar{y}$ mainly contains a phase-shifted sine wave at the perturbation frequency. The design of $H(s)$ is not unique, but a first-order filter of the form

$$H(s) = \frac{s}{s + \omega_h}, \tag{4.8}$$

is often sufficient, where ω_h determines the cut-off frequency for the filter. Typically, $\omega_h = \omega$ is acceptable, but if $\bar{y}$ changes quickly, it may be necessary to make $\omega_h > \omega$. While the low-pass filter is not actually required, the filter is typically used to remove the $\cos\left(2\omega t + \phi_p + \phi_H + \phi\right)$ term in η from (4.7) and to help prevent $\hat{u}$ from changing too quickly. Once again, a first-order filter of the form

$$L(s) = \frac{\omega_l}{s + \omega_l}, \tag{4.9}$$

is often sufficient, where ω_l determines the cut-off frequency for the filter. While $\omega_l = \omega$ is typically acceptable, in choosing ω_l it is important to ensure that the perturbation in η is sufficiently attenuated while still allowing $\hat{u}$ to adequately track changes in u^* caused by disturbances to the system.

While the above procedure for determining the various ESC parameters is useful, the tuning of these parameters is highly dependent on the dynamics of the particular system being controlled. The parameters chosen for the example system are presented in Chapter 6.

4.3 ESC Augmentations

In Section 4.1, a basic gradient-based ESC approach is given. However, there are several commonly used augmentations and alternative approaches which provide greater control performance, typically in terms of convergence rate and robustness to disturbances. First, the use of dynamics compensators, as developed in [24] and [25], has been shown to allow for faster adaption through improvements in the relative degree and phase response of the system.

Alternative methods for gradient estimation have also been proposed which provide faster convergence. Extended Kalman filters (EKF) are used for this purpose in [26] and an observer based method is presented in [20], which not only identifies the gradient but also the curvature of the system allowing for the use of a Newton-like algorithm. There has also been some work where gradient information is not used and a trust region-based approach is applied instead [27].

In addition to these alternative approaches, there are several augmentations that can be used to improve the performance of any ESC algorithm. In [21], three such augmentations are presented, including anti-windup, integrator resetting, and high-pass filter resetting. Anti-windup [28] is a widely used technique to prevent controllers with integral action from 'winding up' in the presence of actuator constraints. Both the integrator and high-pass filter resetting techniques are used to improve the transient response of the ESC algorithm following a large abrupt change in the plant output due to a disturbance. The perturbation signal amplitude can also be dynamic. In [20], a Dither Signal Amplitude Schedule (DSAS) is proposed which changes the magnitude of the perturbation signal based on the gradient estimation. If the gradient is large, implying that the input is far from the optimal, the perturbation amplitude increases to improve the convergence rate. However, once the gradient is small and the input is close to the optimal, the amplitude is reduced so that the input oscillates within a small region of the optimal value. Finally, the shape of the perturbation signal can also be modified. In [29], it was found that a square wave can provide faster convergence when compared to sine and triangular wave perturbations.

4.4 ESC Formulation for Example System

With the different approaches and augmentations presented in the previous section, the schematic presented in Fig. 4.1 and the algorithm detailed in Section 4.1 are used with two modifications to make ESC much more suitable in practice. Evaluating the potential convergence improvements provided by the other approaches is left for future work. Fig. 4.2 shows the modified ESC schematic with two additional features.

First an anti-windup feature is added to retain the ESC functionality even when the input u saturates. The anti-windup scheme used comes from [28]. Additionally, in practice, the plant output y may change quickly due to a disturbance to the system. This causes $\bar{y}$ to change

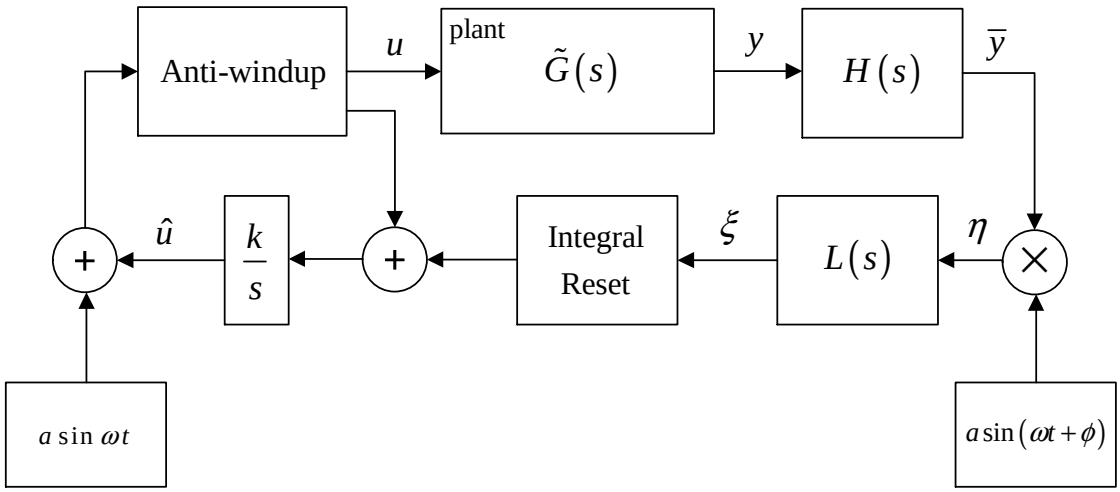

Figure 4.2. Modified ESC schematic.

quickly and, therefore, may not be adequately removed with the high-pass filter. This can cause the magnitude of ξ to become very large, causing large changes in $\hat{u}$, which may cause the system to go unstable. Therefore, an integrator resetting scheme [21] is used such that

$$\xi = \begin{cases} \xi & if \quad |\xi| < \xi_0 \\ 0 & if \quad |\xi| \geq \xi_0 \end{cases}, \tag{4.10}$$

where ξ_0 is chosen based on values of ξ determined to prevent the system from going unstable.

With this ESC formulation and the MPC formulation from Chapter 3, the following chapter develops several control architectures which are used to meet the performance and efficiency control objectives for BAS systems.

Chapter 5
Controller Architectures

With the class of large-scale BAS systems presented in Chapter 2, there are numerous control architectures available which can be used to meet the various local and global performance and efficiency objectives. Despite having the same control objectives, different control architectures attempt to meet these objectives under different constraints on the information available to the controller. As presented in Chapter 2, the two types of information constraints relevant to BAS systems are constraints on communication and constraints on plant knowledge. These information constraints are very important to consider when developing and evaluating a control architecture. In industry, communication and plant knowledge are often constrained when developing a control architecture to make the control design and implementation more practical by reducing setup and commissioning cost and avoiding excessive communication and computational costs. Therefore, it is important to understand how imposing different types of constraints on information can improve the ability to implement a control architecture and the associated potential reductions in control performance due to these constraints.

This chapter focuses on 3 such control architectures which are denoted as *centralized*, *decentralized*, and *BAS*. In a notional sense, Fig. 5.1a shows the relative degrees of plant knowledge and communication required for the 3 architectures. The centralized approach has no constraints on plant knowledge and communication, whereas the decentralized approach is constrained to use the least amount of plant model and communication. The BAS approach uses the same plant knowledge as the centralized approach but requires significantly less communication.

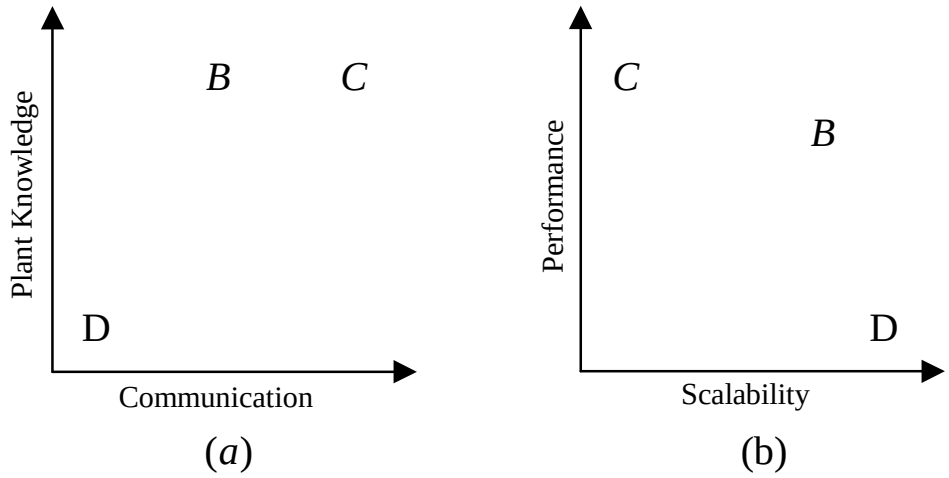

Figure 5.1 Qualitative comparison between centralized, C, decentralized, D, and BAS, B, control architectures – a) relative plant knowledge and communication, b) relative performance and scalability.

As will be demonstrated using the example system in Chapter 6, the varying degrees of plant knowledge and communication for each of the 3 architectures result in different levels of performance and scalability, as shown in Fig. 5.1b. Here, performance refers to the ability to meet both the performance and efficiency objectives as defined in Section 2.4. Scalability refers to the feasibility and practicality of applying each control architecture to a large-scale system. While the centralized control approach achieves the best performance, the large communication and plant knowledge requirements restrict the scalability of this architecture. The decentralized control approach is very scalable but performs significantly worse than the centralized approach. By utilizing additional plant knowledge, the BAS approach performs significantly better than the decentralized approach, while still remaining relatively scalable.

The centralized, decentralized, and BAS control architectures are presented in Sections 5.1, 5.2, and, 5.3 respectively. The issue of model uncertainty is addressed in Section 5.4, where extremum seeking control is proposed as a possible solution.

5.1 Centralized Control Architecture

Fig. 4.1 shows a schematic of the centralized control architecture.

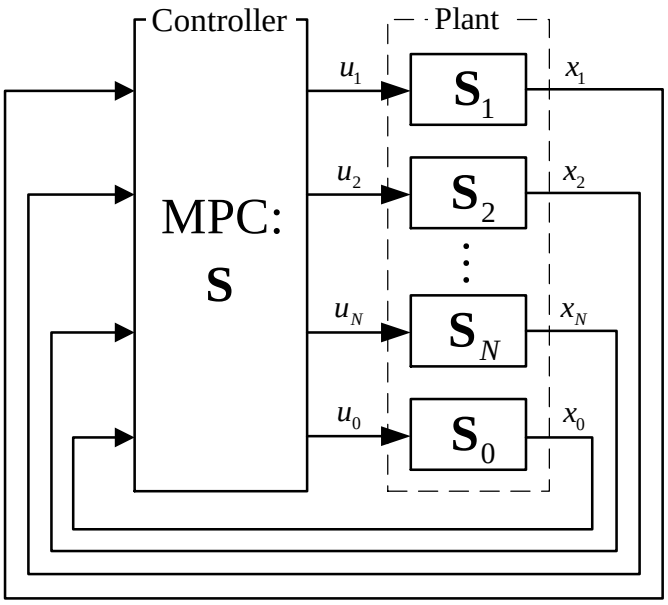

Figure 5.2 Centralized control architecture.

For the plant $\mathbf{S}$ with BAS structure presented in Chapter 2 and rewritten here as

$$\mathbf{S}:\left[\begin{array}{c}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_N\\ \hline \dot{x}_0\end{array}\right]=\left[\begin{array}{cccc|c}A_{11} & 0 & \cdots & 0 & A_{10}\\ 0 & A_{22} & \ddots & \vdots & A_{20}\\ \vdots & \ddots & \ddots & 0 & \vdots\\ 0 & \cdots & 0 & A_{NN} & A_{N0}\\ \hline A_{01} & A_{02} & \cdots & A_{0N} & A_{00}\end{array}\right]\left[\begin{array}{c}x_1\\ x_2\\ \vdots\\ x_N\\ \hline x_0\end{array}\right]+\left[\begin{array}{cccc|c}B_{11} & 0 & \cdots & 0 & B_{10}\\ 0 & B_{22} & \ddots & \vdots & B_{20}\\ \vdots & \ddots & \ddots & 0 & \vdots\\ 0 & \cdots & 0 & B_{NN} & B_{N0}\\ \hline B_{01} & B_{02} & \cdots & B_{0N} & B_{00}\end{array}\right]\left[\begin{array}{c}u_1\\ u_2\\ \vdots\\ u_N\\ \hline u_0\end{array}\right]$$
$$+\left[\begin{array}{cccc|c}V_{11} & 0 & \cdots & 0 & 0\\ 0 & V_{22} & \ddots & \vdots & 0\\ \vdots & \ddots & \ddots & 0 & \vdots\\ 0 & \cdots & 0 & V_{NN} & 0\\ \hline 0 & 0 & \cdots & 0 & V_{00}\end{array}\right]\left[\begin{array}{c}d_1\\ d_2\\ \vdots\\ d_N\\ \hline d_0\end{array}\right], \tag{5.1}$$

or

$$\mathbf{S}:\dot{x}=Ax+Bu+Vd, \tag{5.2}$$

the centralized control architecture uses the complete system model (A, B, V) as well as the complete set of states x to determine the control inputs u for the system. Additionally, it is assumed that the controller has access to all the disturbances d. Chapter 7 considers the control performance of a system when this is not the case. By using information about the entire system,

the centralized architecture produces a Pareto optimal solution. However, centralized architectures can rarely be applied to large-scale systems. The extensive communication and computational requirements associated with requiring access to all of the states and a complete model of the system makes centralized control infeasible for many large-scale applications. Therefore, the centralized control architecture is used to represent the best case scenario in terms of controller performance with the understanding that this is not a practical solution for many large-scale systems.

5.1.1 Centralized Control Problem

The centralized control architecture solves a single control problem. This control problem is the same that was presented in Chapter 2, and is rewritten here as

$$\begin{aligned}
\underset{U}{\text{minimize}} \quad & J = \sum_{j=1}^{N_p}\left[\sum_{i=0}^{N}\left(\gamma_a\left(\gamma_b J_{p,i} + (1-\gamma_b) J_{u,i}\right) + (1-\gamma_a) J_{\Delta u,i}\right)\right], \\
\text{subject to} \quad & u(k+j-1) \in \mathcal{U}, \quad x(k+j) \in \mathcal{X}, \quad j \in \left[1, N_p\right], \\
& x(k+j) = Ax(k+j-1) + Bu(k+j-1) + Vd(k+j-1), \quad j \in \left[1, N_p\right], \\
& y(k+j) = Cx(k+j), \quad j \in \left[1, N_p\right], \\
& x(k) = x_0 \in \mathcal{X}.
\end{aligned} \tag{5.3}$$

The centralized control architecture has access to (A, B, V), which model the dynamics of the entire system $\mathbf{S}$. Using this model and knowledge of all the states in the system $x(k)$ at time instance k, the centralized control problem is solved to produce the complete set of control inputs u.

5.1.2 Centralized Formulation for Example System

The MPC formulation developed in Chapter 2 for the example system is used for the centralized control formulation and is outlined here. First, the complete system model $\mathbf{S}$

$$\begin{aligned}
\mathbf{S}: \dot{x} &= Ax + Bu + Vd, \\
y &= Cx,
\end{aligned} \tag{5.4}$$

is used by the MPC controller where $x \in \mathbb{R}^8$, $u \in \mathbb{R}^8$, $d \in \mathbb{R}^4$, $y \in \mathbb{R}^8$, and C is

$$C = \left[\begin{array}{cccccccc|c} & & & I_{7\times 7} & & & & & 0_{7\times 1} \\ \hline \frac{1}{R_{e1}} & 0 & \frac{1}{R_{e2}} & 0 & \frac{1}{R_{e3}} & 0 & 0 & & -\left(\frac{1}{R_{e1}}+\frac{1}{R_{e2}}+\frac{1}{R_{e3}}\right) \end{array}\right]. \tag{5.5}$$

Since there are only 4 reference values, $r \in \mathbb{R}^4$, the matrix M is used to select which outputs have desired reference values,

$$M = \text{blkdiag}\left(\begin{bmatrix}0 & 1\end{bmatrix},\begin{bmatrix}0 & 1\end{bmatrix},\begin{bmatrix}0 & 1\end{bmatrix},\begin{bmatrix}0 & 1\end{bmatrix}\right). \tag{5.6}$$

The system is then augmented into the form shown in (3.24) to include error states e, integral states z, and changes in control input Δu. Next, the system is written in a lifted form as shown in (3.26). The cost function is rewritten as a function of the lifted output $\bar{Y}$, the lifted inputs $\Delta\bar{U}$, the weightings Q_1 and Q_2, and the actuator resource consumption costs K. The weighting matrix Q_1 is used to penalize the magnitudes of the error and integral states and weighting matrix Q_2 is used to penalize the changes in control inputs. The actuator cost term K represents the efficiency objective to be minimized and is a quadratic function of the lifted inputs U, the matrix Q_a, and the vector q_b, as defined in (3.32). The actuator constraints are written in a lifted form and converted to be a function of ΔU and the actuator input $u(k)$ at sample time k, as shown in (3.36). Finally, the optimization problem is solved to determine the sequence of inputs which satisfy (3.36) and minimize J (3.33).

5.2 Decentralized Control Architecture

The decentralized control architecture is designed to be much more scalable than the centralized architecture but at the cost of control performance. Fig. 5.3 shows a schematic of the decentralized control architecture.

Instead of using a complete system model and knowledge of all the states, decentralized control acts at the subsystem level. For a BAS system (4.1) containing $N+1$ subsystems, N subsystems $\mathbf{S}_i$, defined as

$$\mathbf{S}_i : \dot{x}_i = A_{ii}x_i + B_{ii}u_i + V_{ii}d_i, \tag{5.7}$$

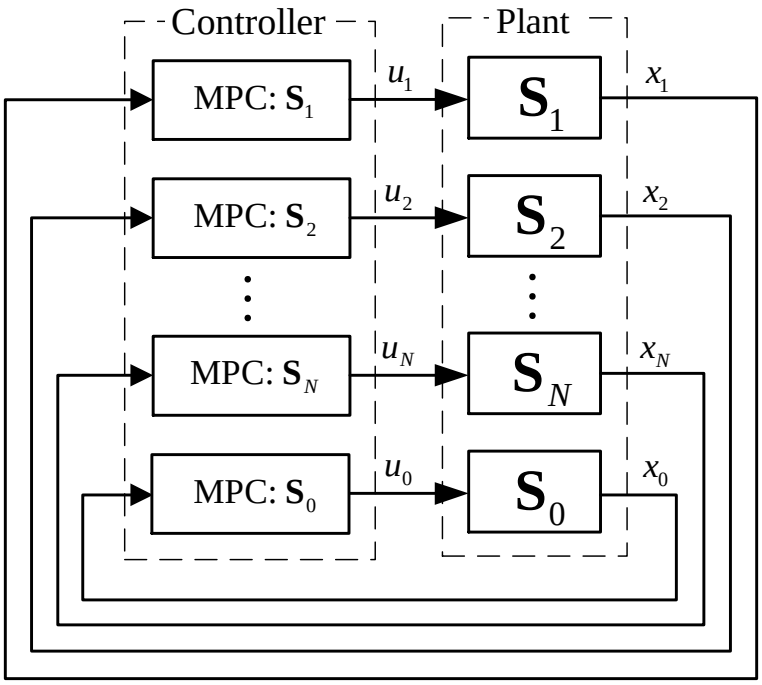

Figure 5.3 Decentralized control architecture.

and one subsystem $\mathbf{S}_0$, defined as

$$\mathbf{S}_0 : \dot{x}_0 = A_{00}x_0 + B_{00}u_0 + V_{00}d_0, \tag{5.8}$$

are used by $N+1$ decentralized controllers. These controllers are designed independently and use only local state information to make control decisions for the actuators of that subsystem.

The decentralized control architecture solves $N+1$ significantly smaller control problems, which drastically improves the scalability of the control architecture. However, by not utilizing information about the interconnections A_{i0}, A_{0i}, B_{i0}, and B_{0i} of the subsystems, the performance of the decentralized control solution is often significantly degraded when compared to the centralized approach. If the subsystems are strongly interconnected, decentralized control may cause the system to go unstable or require the decentralized controller to be significantly less aggressive. Additionally, if there is a system-wide efficiency objective, decentralized control often results in much lower efficiency. Therefore, the decentralized control architecture is used to represent a worst case scenario in terms of controller performance but is the most scalable architecture and the simplest to implement for large-scale systems.

5.2.1 Decentralized Control Problem

For the decentralized control architecture, the single control problem from (5.3) is broken into $N+1$ smaller control problems corresponding to the N subsystems $\mathbf{S}_i$ and the common subsystem $\mathbf{S}_0$. The control problem for the $\mathbf{S}_i$ subsystems is

$$\begin{aligned}
\underset{U_i}{\text{minimize}} \quad & J = \sum_{j=1}^{N_p}\left[\gamma_a\left(\gamma_b J_{p,i} + (1-\gamma_b) J_{u,i}\right) + (1-\gamma_a) J_{\Delta u,i}\right], \\
\text{subject to} \quad & u_i(k+j-1) \in \mathcal{U}_i, \quad x_i(k+j) \in \mathcal{X}_i, \quad j \in \left[1, N_p\right], \\
& x_i(k+j) = A_{ii} x_i(k+j-1) + B_{ii} u_i(k+j-1) + V_{ii} d_i(k+j-1), \quad j \in \left[1, N_p\right], \\
& y_i(k+j) = C_{ii} x_i(k+j), \quad j \in \left[1, N_p\right], \\
& x_i(k) = x_{0,i} \in \mathcal{X}_i.
\end{aligned} \tag{5.9}$$

The control problem for the $\mathbf{S}_0$ subsystem is

$$\begin{aligned}
\underset{U_0}{\text{minimize}} \quad & J = \sum_{j=1}^{N_p}\left[\gamma_a\left(\gamma_b J_{p,0} + (1-\gamma_b) J_{u,0}\right) + (1-\gamma_a) J_{\Delta u,0}\right], \\
\text{subject to} \quad & u_0(k+j-1) \in \mathcal{U}_0, \quad x_0(k+j) \in \mathcal{X}_0, \quad j \in \left[1, N_p\right], \\
& x_0(k+j) = A_{00} x_0(k+j-1) + B_{00} u_0(k+j-1) + V_{00} d_0(k+j-1), \quad j \in \left[1, N_p\right], \\
& y_0(k+j) = C_{00} x_0(k+j), \quad j \in \left[1, N_p\right], \\
& x_0(k) = x_{0,0} \in \mathcal{X}_0.
\end{aligned} \tag{5.10}$$

The individual controllers in the decentralized control architecture only have access to parts of each matrix which model only the dynamics of the corresponding subsystem $\mathbf{S}_i \forall i \in [0, N]$. Using these subsystem models and knowledge of only the local states $x_i(k)$ at time instance k, the decentralized control problems are solved to produce only the local control inputs u_i.

5.2.2 Decentralized Formulation for Example System

The decentralized control formulation follows the same procedure as defined for the centralized control formulation in Section 5.1.2. For the N subsystems $\mathbf{S}_i$, the subsystem model $\mathbf{S}_i$

$$\begin{aligned} \mathbf{S}_i : \dot{x}_i &= A_{ii}x_i + B_{ii}u_i + V_{ii}d_i, \\ y_i &= C_{ii}x_i, \end{aligned} \tag{5.11}$$

is used by the corresponding MPC controller where $x_i \in \mathbb{R}^2$, $u_i \in \mathbb{R}^2$, $d_i \in \mathbb{R}^1$, $y_i \in \mathbb{R}^2$, and $C_{ii} = I_{2\times 2}$. Since there is only one reference value for each subsystem, $r_i \in \mathbb{R}^1$, the matrix $M_i = \begin{bmatrix} 0 & 1 \end{bmatrix}$ is used to select the output with the desired reference value. Similarly, the system model $\mathbf{S}_0$

$$\begin{aligned} \mathbf{S}_0 : \dot{x}_0 &= A_{00}x_0 + B_{00}u_0 + V_{00}d_0, \\ y_0 &= C_{00}x_0, \end{aligned} \tag{5.12}$$

is used by the corresponding MPC controller where $x_0 \in \mathbb{R}^2$, $u_0 \in \mathbb{R}^2$, $d_0 \in \mathbb{R}^1$, $y_0 \in \mathbb{R}^2$, and C_{00} is

$$C_{00} = \begin{bmatrix} 1 & 0 \\ 0 & -\left(\frac{1}{R_{e1}} + \frac{1}{R_{e2}} + \frac{1}{R_{e3}} \right) \end{bmatrix}. \tag{5.13}$$

Once again, since there is only one reference value for each subsystem, $r_0 \in \mathbb{R}^1$, the matrix $M_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}$ is used to select the output with the desired reference value. The remainder of the steps in Section 5.1.2 can be directly applied to these decentralized subsystems to complete the decentralized control formulation.

5.3 BAS Control Architecture

The BAS control approach directly utilizes the distinct structure of BAS systems. Fig. 5.4 shows a schematic of the BAS control architecture.

Whereas the decentralized control approach had $N+1$ controllers, the BAS architecture only has N controllers corresponding to the N subsystems $\mathbf{S}_i$. Based on the system (4.1), N new *BAS* subsystems, denoted with a prime, are defined as

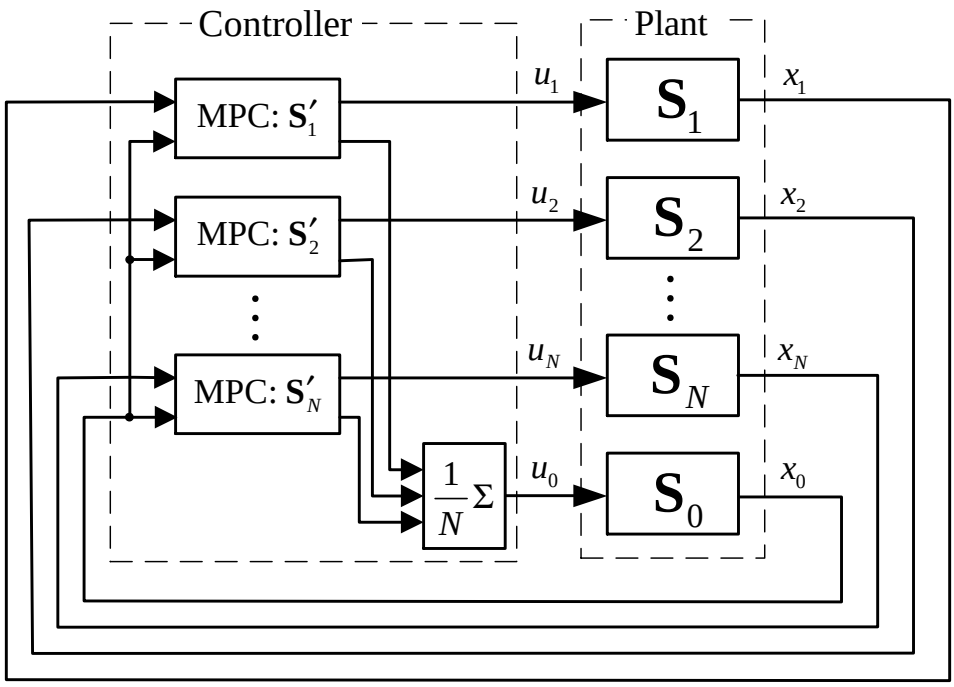

Figure 5.4 BAS control architecture.

$$\mathbf{S}_i' : \begin{bmatrix} \dot{x}_i \\ \dot{\tilde{x}}_0 \end{bmatrix} = \begin{bmatrix} A_{ii} & A_{i0} \\ A_{0i} & A_{00} \end{bmatrix} \begin{bmatrix} x_i \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} B_{ii} & B_{i0} \\ B_{0i} & B_{00} \end{bmatrix} \begin{bmatrix} u_i \\ \tilde{u}_0 \end{bmatrix} + \begin{bmatrix} V_{ii} & 0 \\ 0 & V_{00} \end{bmatrix} \begin{bmatrix} d_i \\ d_0 \end{bmatrix}, \tag{5.14}$$

or

$$\mathbf{S}_i' : \dot{x}_i' = A_{ii}' x_i' + B_{ii}' u_i' + V_{ii}' d_i'. \tag{5.15}$$

Each of these BAS subsystems includes one of the original $\mathbf{S}_i$ subsystems and the common $\mathbf{S}_0$ subsystem. Thus, knowledge of the subsystem interconnections A_{i0}, A_{0i}, B_{i0}, and B_{0i} is included in the BAS subsystem models and available to each BAS controller. Using state information x_i and x_0 and the corresponding BAS subsystem model, each controller makes control decisions for the actuator inputs u_i of the subsystem $\mathbf{S}_i$ and u_0 of the subsystem $\mathbf{S}_0$. Note that the states and inputs corresponding to $\mathbf{S}_0$ are denoted with a tilde. With an MPC framework, the tilde is used to highlight the fact that each of the N controllers may predict a unique set of future values for x_0 and may determine different control inputs u_0. This means that there are N sets of control signals $\tilde{u}_0$ for the subsystem $\mathbf{S}_0$. Thus, the control decisions $\tilde{u}_0$

need to be combined into a single set of control inputs u_0. For this work, the N input signals $\tilde{u}_0$ are averaged to produce u_0, written as

$$u_0 = \frac{1}{N}\sum_{i=1}^{N} \tilde{u}_0 \tag{5.16}$$

With the knowledge of how the $\mathbf{S}_i$ and $\mathbf{S}_0$ subsystems interact, the BAS controller does not suffer from the same stability issues that may arise from the decentralized approach and can be designed to be significantly more aggressive. From the construction of the BAS controller, state and input information only need to be passed between the common subsystem $\mathbf{S}_0$ and each subsystem $\mathbf{S}_i$. There is no need to pass information among subsystems $\mathbf{S}_i$, allowing this architecture to scale to systems where N is large.

5.3.1 BAS Control Problem

For the BAS control architecture, the single control problem from (5.3) is broken into N smaller control problems corresponding to the N subsystems $\mathbf{S}_i$. The control problem for each subsystem $\mathbf{S}_i$ is

$$\begin{aligned}
\underset{U_i'}{\text{minimize}} \quad & J = \sum_{j=1}^{N_p}\Big[\gamma_a\left(\gamma_b J'_{p,i} + (1-\gamma_b) J'_{u,i}\right) + (1-\gamma_a) J'_{\Delta u,i}\Big], \\
\text{subject to} \quad & u_i'(k+j-1) \in \mathcal{U}_i', \quad x_i'(k+j) \in \mathcal{X}_i', \quad j \in \left[1, N_p\right], \\
& x_i'(k+j) = A_{ii}' x_i'(k+j-1) + B_{ii}' u_i'(k+j-1) + V_{ii}' d_i'(k+j-1), \quad j \in \left[1, N_p\right], \\
& y_i'(k+j) = C_{ii}' x_i'(k+j), \quad j \in \left[1, N_p\right], \\
& x_i'(k) = x_{0,i}' \in \mathcal{X}_i.
\end{aligned} \tag{5.17}$$

Using the subsystem models from (5.15) and knowledge of only the local states $x_i(k)$ and the common states $x_0(k)$ at time instance k, the individual controllers in the BAS control architecture solve the BAS control problems to produce the local control inputs u_i and the common control inputs $\tilde{u}_0$. These common inputs are combined according to (5.16).

5.3.2 BAS Formulation for Example System

As with the decentralized control formulation, the BAS control formulation follows the procedure from Section 5.1.2. First, the BAS system model $\mathbf{S}_i'$

$$\begin{aligned} \mathbf{S}_i' : \dot{x}_i' &= A_{ii}' x_i' + B_{ii}' u_i' + V_{ii}' d_i', \\ y_i' &= C_{ii}' x_i', \end{aligned} \tag{5.18}$$

is used by the corresponding MPC controller where $x_i' \in \mathbb{R}^4$, $u_i' \in \mathbb{R}^4$, $d_i' \in \mathbb{R}^2$, $y_i' \in \mathbb{R}^4$, and C is

$$C_{ii}' = \left[\begin{array}{ccc|c} & I_{3\times 3} & & 0_{3\times 1} \\ \hline \frac{1}{R_{ei}} & 0 & 0 & -\left(\frac{1}{R_{e1}} + \frac{1}{R_{e2}} + \frac{1}{R_{e3}}\right) \end{array}\right]. \tag{5.19}$$

Since there are 2 reference values, $r \in \mathbb{R}^2$, the matrix M_i' is used to select which outputs have desired reference values,

$$M_i' = \text{blkdiag}\left(\begin{bmatrix} 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \end{bmatrix}\right). \tag{5.20}$$

Once again, the remainder of the steps in Section 5.1.2 can be directly applied to these BAS subsystem to complete the BAS control formulation.

5.4 ESC Control Architecture

It is important to note that the relationships presented in Fig. 5.1 only hold when the plant knowledge used by the various controllers is perfect. However, this is often not the case. When controlling most systems, there are numerous sources of uncertainty that can cause significant differences between the system representation used when determining control decisions and the true system behavior. Examples of these sources of uncertainty are model inaccuracies from the system identification process, unknown or unmodeled disturbances, or system nonlinearity. While these uncertainties do affect the ability to meet the performance objectives for the system, feedback controllers are often designed with integral action to overcome model uncertainty. These uncertainties do have a major effect, however, when it comes to meeting the system-level efficiency objectives. Since there is not a well-defined, achievable desired value for these

objectives, integral action cannot be used to help meet these objectives. Therefore, the set of control decision determined to be optimal for the model used by the controller, may not be optimal for the actual system. This idea is demonstrated in Section 7.6, where it is shown that the centralized control approach may be more sensitive to model uncertainty.

With the degradation in system efficiency due to model inaccuracy, a model-free control approach is desirable and can be used to drive the system to the most efficient operating condition while still meeting the performance objectives. Extremum seeking control (ESC), presented in Chapter 4, is a model-free adaptive control approach used to drive a system input u to an optimal input u^* which minimizes (or maximizes) a system output $\underline{y}$.

While ESC can be used to control an actuator input directly, as discussed in Chapter 4, ESC is an adaptive algorithm that requires a perturbation signal to be sufficiently slower than the dynamics of the system. This means the actuator input controlled by ESC is not able to respond quickly to disturbances to the system. Therefore, it is desirable to combine the fast transient performance of the model-based MPC approaches presented above with the ability to achieve greater system efficiency through ESC in the presence of model uncertainty. Any model-based control strategy can be augmented with ESC as demonstrated for the BAS architecture in Fig. 5.5. For this study, only single-input ESC is considered, meaning only one actuator input signal can be augmented by ESC. The augmented inputs signal is denoted as $\underline{u}$ and is the sum of the control signal $\hat{u}$ from the MPC controller and the control signal $\tilde{u}$ from the ESC controller,

$$\underline{u} = \hat{u} + \tilde{u}. \tag{5.21}$$

The remaining un-augmented control inputs are denoted as $\overline{u}$.

Clearly, the choice of $\underline{u}$ is not unique. However, $\underline{u}$ must be chosen strategically in order to achieve the greatest effect and the ability to drive the system from the operating condition determined by the MPC control to the truly optimal operating condition. Typically, the efficiency objective is a global objective and, therefore, it is necessary that $\underline{u}$ has the ability to sufficiently affect the entire system. If $\underline{u}$ is isolated and not coupled to the rest of the system, augmenting $\underline{u}$ does not provide the necessary change in system operation. Therefore, for a BAS system, intuition says to choose one of the common system inputs u_0 as $\underline{u}$. Additionally, if one

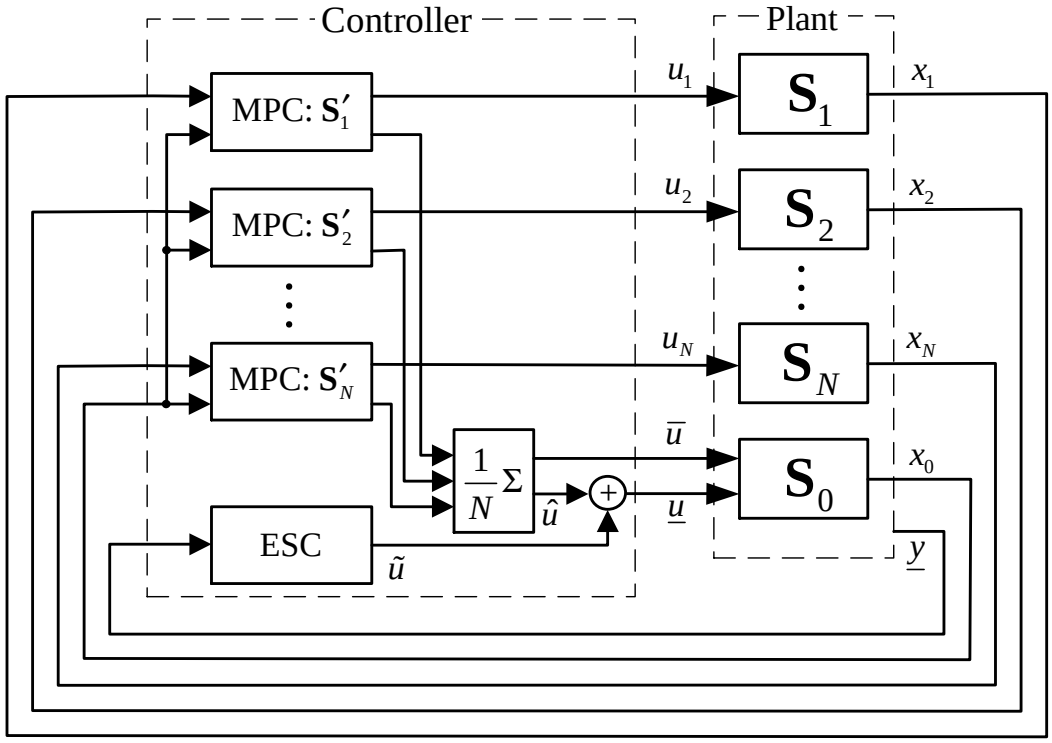

Figure 5.5 Augmented control architecture with ESC.

of the u_0 has a strong connection to one of the performance objectives for $\mathbf{S}_0$, it is suggested not to choose that input as the augmented input $\underline{u}$. From here, the decision of $\underline{u}$ is highly dependent on the system being controlled and this decision is left to the control designer.

5.4.1 ESC Control Problem

Since the ESC algorithm is used to simply modify a control input signal from an MPC controller, either the centralized control problem (5.3), the decentralized control problems (5.9) and (5.10), or the BAS control problem (5.17), can be used. Once the control signal $\hat{u}$ is chosen, the ESC algorithm presented in Chapter 4 is used to generate a control signal $\tilde{u}$ which drives $\underline{u}$ to minimize the efficiency cost term

$$J_u = \sum_{i=0}^{N} J_{u,i}, \tag{5.22}$$

which is calculated as the output $\underline{y}$ of the system.

5.4.2 ESC Formulation for Example System

For the example system, u_g is chosen as the control input to be augmented by the ESC controller. As stated above, it is important to augment a control input which sufficiently affects the entire system. From (2.39), it is clear that u_g directly affects V_g, which influences V_e and all of the states V_i and V_{ai}, as seen in (2.30). The input u_t was not chosen to be augmented due to its direct influence on V_e which is used in the calculation of I_e. Since the control of I_e is one of the global performance objectives for the system, it is important to utilize u_t to effectively meet this objective. The output $\underline{y}$ from the system, to be minimized by the ESC algorithm, is the sum of the actuator resource consumption costs, J_u. This output is calculated in real-time based on the actuator inputs u_j and the cost function for each actuator, as seen in (2.46) and (2.47).

With the input and output for the ESC algorithm determined, the only additional step is to determine the perturbation frequency ω, perturbation amplitude a, scaling factor k, phase shift ϕ, and the high and low-pass filters, $H(s)$ and $L(s)$. Since tuning these parameters is based on the dynamics of the actual system, the selection of these parameters is presented in Chapter 6, where the example system from Chapter 3 is given a set of values and simulated in order to evaluate the performance of the various control architectures developed in this chapter.

Chapter 6

Illustrative Example System

Throughout Chapters 2-5, an example system has been used to demonstrate the distinct structure of BAS systems, the development of MPC and ESC controllers, and the formulation of centralized, decentralized, and BAS control architectures. In this Chapter, numerical values are assigned to this example system in order to demonstrate the analysis of BAS systems and the functionality of the various control approaches.

In Section 6.1, parameter values are assigned to the example system and the controllability of the system is tested. The performance of the various control architectures is compared for two different scenarios in Section 6.2.1. ESC is used in Section 6.2.2 to further improve the performance of the BAS control strategy. Finally the scalability of the control architectures is analyzed in Section 6.2.3. Appendix A contains the MATLAB code used to generate and control the example system.

6.1 System Parameters and Analysis

Table 6.1 contains the values for the example system parameters presented in Chapter 2. While the 3 subsystems have the same structure, the parameters were chosen to make each subsystem have different nominal conditions and dynamic responses. For example, at the nominal operating condition (all actuators inputs at 50 volts), the amount of current entering each subsystem is significantly different where $I_1 = 3$, $I_2 = 1.3$, and $I_3 = 2.3$. Additionally, the capacitances for the subsystems range from $C_1 = 0.06$ to $C_3 = 0.01$ and $C_{a1} = 2$ to $C_{a3} = 5$.

Table 6.1 Example System Parameters

	Subsystem				
Parameter	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	**Parameter**	$\mathbf{S}_0$
C_i	0.060	0.030	0.010	C_e	0.010
C_{ai}	2.000	4.000	5.000	C_g	0.050
R_i	15.00	22.00	20.00	R_g	5.000
K_i	0.300	0.172	0.120	K_g	0.200
R_{ai}	10.00	45.00	20.00	α_1	0.002
K_{ai}	0.180	0.280	0.460	α_2	0.002
R_{ei}	300.0	400.0	50.00	α_3	0.008
	Initial Condition				
State	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	**State**	$\mathbf{S}_0$
V_i	50	60	40	V_g	80
V_{ai}	30	20	25	V_e	20
	Resource Costs				
Actuator	a_i	b_i	c_i		
u_i	0.01	0	0		
u_{ai}	0.01	0	0		
u_t	0.01	0	0		
u_g	0.02	0	0		

With the parameter values from Table 6.1, the complete system model **S** from (2.41) is now written as

$$\mathbf{S}: \dot{x} = Ax + Bu + Vd, \tag{6.1}$$

where

$$A = \left[\begin{array}{cccccc|cc} -2.833 & 1.667 & & & & & 1.111 & 0.056 \\ 0.050 & -0.050 & & 0_{2\times2} & & 0_{2\times2} & 0 & 0 \\ & & -2.339 & 0.741 & & & 1.515 & 0.083 \\ & 0_{2\times2} & 0.006 & -0.006 & & 0_{2\times2} & 0 & 0 \\ & & & & -12.000 & 5.000 & 5.000 & 2.000 \\ & 0_{2\times2} & & 0_{2\times2} & 0.010 & -0.010 & 0 & 0 \\ \hline 1.333 & 0 & 0.909 & 0 & 1.000 & 0 & -7.202 & 0.040 \\ 0.333 & 0 & 0.250 & 0 & 2.000 & 0 & -0.200 & -2.7833 \end{array}\right], \quad (6.2)$$

$$B = \left[\begin{array}{cccccc|cc} 0.333 & -0.300 & & & & & & \\ 0 & 0.009 & & 0_{2\times2} & & 0_{2\times2} & & 0_{2\times2} \\ & & 0.261 & -0.207 & & & & \\ & 0_{2\times2} & 0 & 0.002 & & 0_{2\times2} & & 0_{2\times2} \\ & & & & 0.600 & -2.300 & & \\ & 0_{2\times2} & & 0_{2\times2} & 0 & 0.0046 & & 0_{2\times2} \\ \hline -0.400 & 0 & -0.1564 & 0 & -0.120 & 0 & 0.160 & 0.800 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.800 & 0 \end{array}\right], \quad (6.3)$$

$$V = \begin{bmatrix} 0 & 0_{2\times1} & 0_{2\times1} & 0_{2\times1} \\ -0.500 & & & \\ 0_{2\times1} & 0 & 0_{2\times1} & 0_{2\times1} \\ & -0.250 & & \\ 0_{2\times1} & 0_{2\times1} & 0 & 0_{2\times1} \\ & & -0.200 & \\ 0_{2\times1} & 0_{2\times1} & 0_{2\times1} & 0 \\ & & & 4.000 \end{bmatrix}. \quad (6.4)$$

Table 6.1 also show the coefficients a_i, b_i, and c_i for the quadratic resource consumption costs for each actuator. For simplicity, the b_i and c_i terms are all 0. Therefore, the consumption cost for each actuator is 0 when $u_i = 0$. The efficiency objective attempts to minimize the sum of the 8 actuator consumption costs while still meeting the performance objectives. For the following studies, the performance objectives are to track the nominal operating conditions (i.e. $V_{a1} = 30$, $V_{a2} = 20$, $V_{a3} = 25$, and $I_e = 0.6$) in the presence of

disturbances. With the nominal input of 50 volts to each actuator, the minimum and maximum constraints for all of the actuators are 0 and 100 volts.

Following the process presented in Chapter 2, it is straightforward to determine that the example system is structurally controllable. With the Boolean representation of the B matrix from (6.3) written as

$$\bar{B} = \left[\begin{array}{cccccc|c} 1 & 1 & & & & & \\ 0 & 1 & & 0_{2\times2} & & 0_{2\times2} & 0_{2\times2} \\ & & 1 & 1 & & & \\ 0_{2\times2} & & 0 & 1 & & 0_{2\times2} & 0_{2\times2} \\ & & & & 1 & 1 & \\ 0_{2\times2} & & 0_{2\times2} & & 0 & 1 & 0_{2\times2} \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 1 \quad 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \quad 0 \end{array}\right], \tag{6.5}$$

it is clear that the G matrix of the reachability matrix R from (2.15) will not have any zero rows and, therefore, the system is input reachable. Additionally, the system has full generic rank (i.e. $\tilde{\rho}\left(\left[\tilde{A} \quad \tilde{B}\right]\right) = 8$). This is easily verified using the fact that each of the block matrices along the diagonal B_{11}, B_{22}, B_{33}, and B_{00} has a generic rank of 2. Since the system is input reachable and full generic rank, the example system is structurally controllable. For this system, structural controllability implies controllability. An example of when this is not the case is presented in Chapter 7.

6.2 Control Architecture Comparison

In this section, the centralized, decentralized, and BAS control strategies are compared under two conditions. First, the MPC control architectures from Chapter 5 are implemented for two different scenarios to analyze how each control approach meets the performance and efficiency objectives. Second, the BAS architecture is augmented with ESC, as developed in Chapters 4 and 5, in order to improve system efficiency for the BAS control approaches. Finally, the different control approaches are compared in terms of scalability.

6.2.1 Baseline Controller Performance

Two scenarios are used to evaluate the performance of the centralized, decentralized, and BAS control strategies. Scenario 1 demonstrates the control performance when each control architecture is able to meet the performance objectives. Scenario 2 compares the control architectures when this is not the case due to actuator saturation. For the MPC controllers used by each control architecture, a sample time of $\Delta t = 10$ seconds is used with a control horizon of $N_u = 15$ steps and a prediction horizon of $N_p = 30$ steps.

6.2.1.1 Scenario 1

Fig. 6.1 shows the disturbances I_{Li} and V_h over the 80 minute simulation for Scenario 1. These disturbances are roughly 10 – 20% deviations from the nominal conditions and were designed to sufficiently test the performance of the various control architectures. For the following comparisons, each controller has access to the disturbance information when making control decisions. The effects of unknown disturbances are studied for a more realistic system in Chapter 7.

For comparison purposes, the open-loop responses of the system due to the disturbances from Fig. 6.1 are shown in Fig. 6.2 and 6.3. All of the actuators are held constant at the nominal

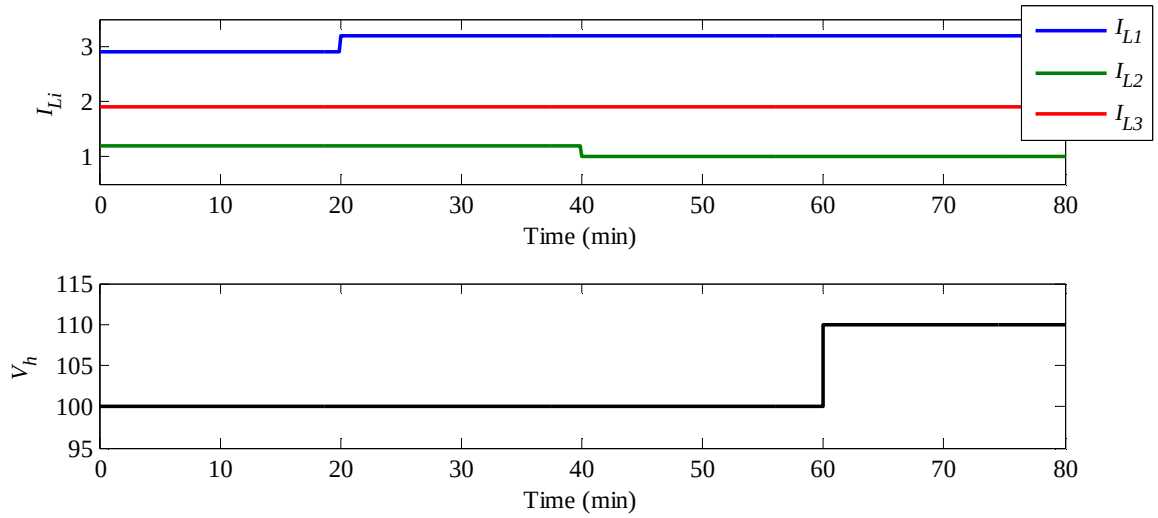

Figure 6.1 System disturbances I_{Li} and V_h for Scenario 1.

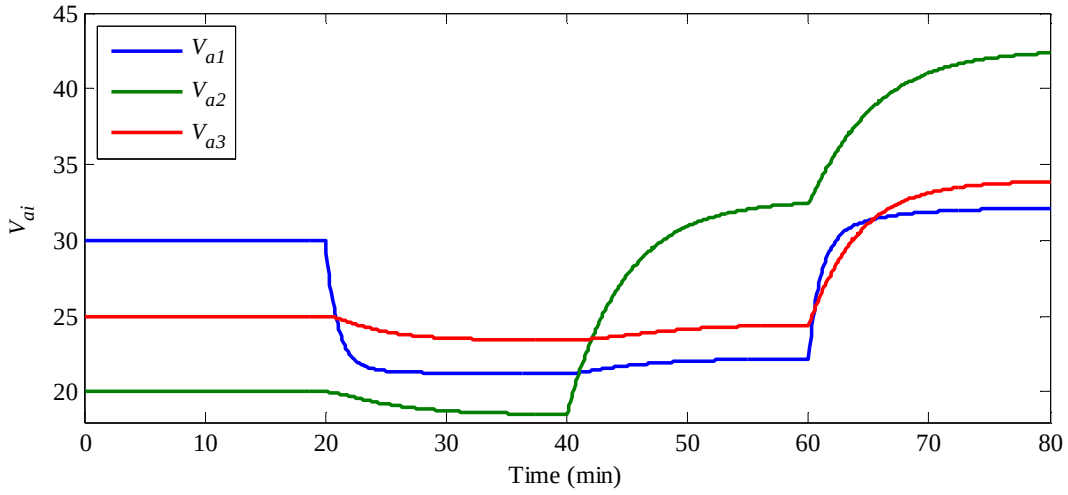

Figure 6.2 Open-loop response for V_{ai} due to the disturbances for Scenario 1.

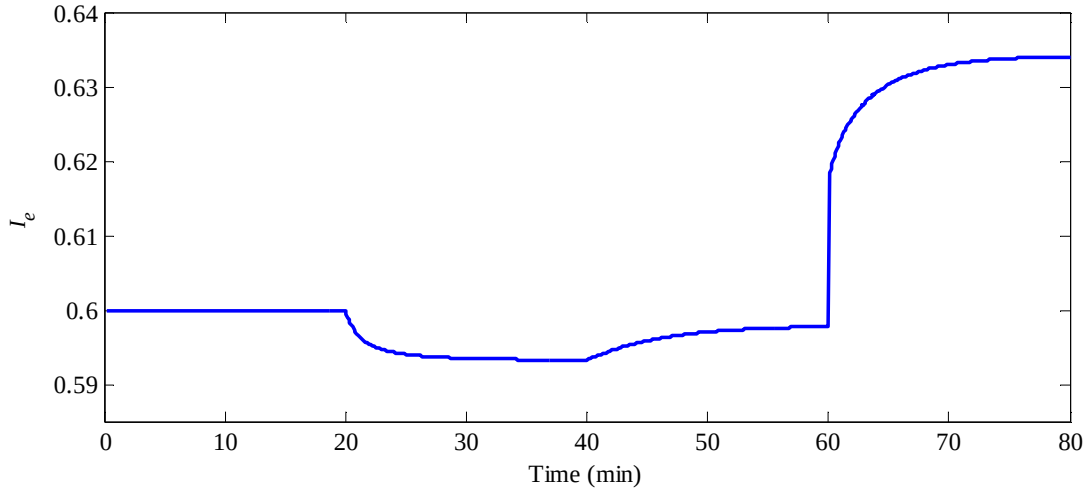

Figure 6.3 Open-loop response for I_e due to the disturbances for Scenario 1.

input values of 50 volts. With the first disturbance at 20 minutes into the simulation, V_{ai} and I_e, which represent the performance objectives, deviate significantly from the nominal conditions.

Figs. 6.4 and 6.5 show the closed-loop responses in V_{ai} and I_e for each of the three control architectures. Clearly, each architecture is able to meet the objective, but there is a difference in the transient responses. As expected, the deviations from the reference values following a disturbance are significantly smaller for the centralized controller opposed to the

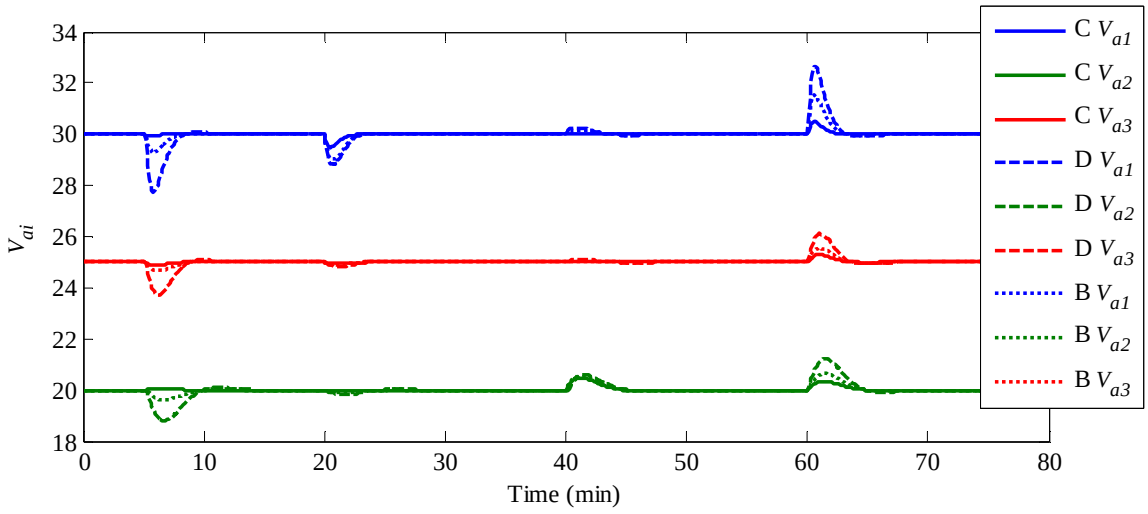

Figure 6.4 Ability of each control architecture to meet local performance objectives by tracking the desired value for V_{ai} for Scenario 1.

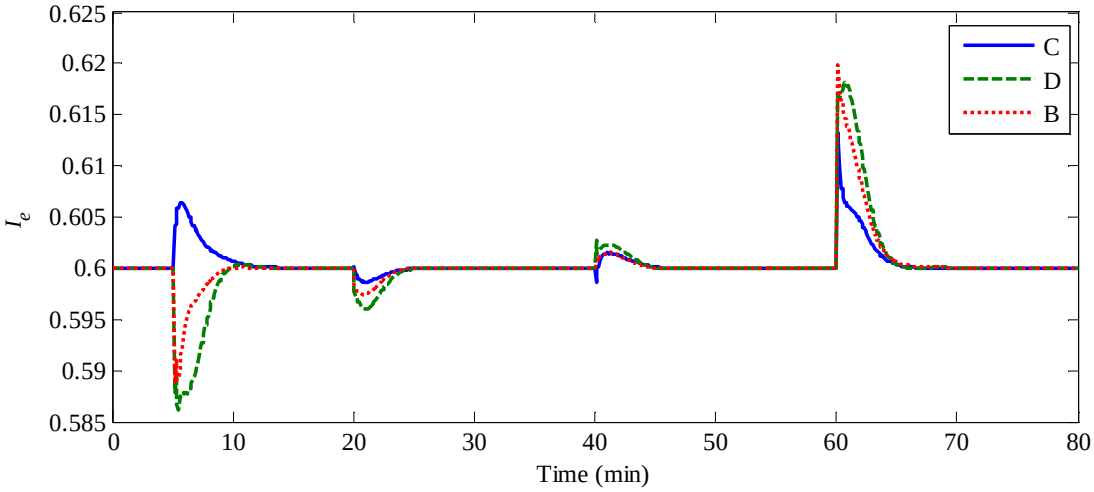

Figure 6.5 Ability of each control architecture to meet the global performance objective by tracking the desired value for I_e for Scenario 1.

decentralized approach due to the coupling in the system. The BAS approach consistently performs better than the decentralized approach and worse that the centralized approach.

The major difference between the control architectures, however, is seen when comparing the efficiency objective. Fig. 6.6 shows that the centralized controller is able to achieve a significantly lower actuator cost than the decentralized and BAS approaches. It is

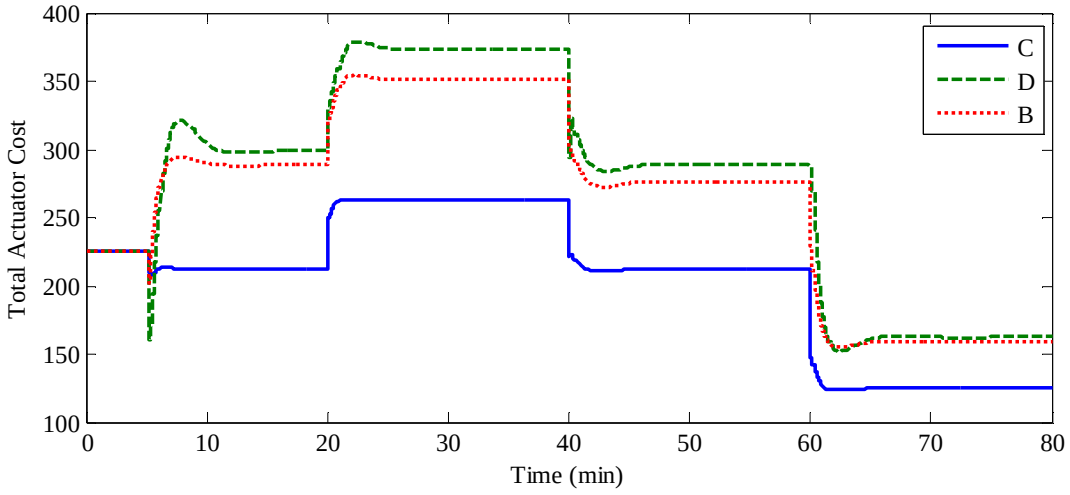

Figure 6.6 Ability of each control architecture to meet the global efficiency objective by minimizing the total actuator cost for Scenario 1.

interesting to note that the centralized controller is the only architecture that was able to improve the system efficiency in the first 20 minutes of simulation after the controller is turned on at 5 minutes into the simulation. The other two architectures result in a large increase in actuator cost when the controllers are turned on, even though the disturbances and references for the system are constant for the first 20 minutes. After integrating the total actuator cost over the 80 minutes of simulation, the cost for the decentralized approach was 35% higher than that of the centralized approach. With a 30% higher cost, the BAS approach was only slightly better than the decentralized. In order to see how the different control approaches resulted in such different efficiencies, despite the fact that they performed very similarly at meeting the performance objectives, the actuator inputs for each control architecture are shown in Figs. 6.7, 6.8, and 6.9. From these figures, it is clear that the major difference between the control architectures, which led to the difference in efficiencies, is the control of actuator u_g, seen in Fig. 6.9. Both the decentralized and BAS approaches drastically reduce u_g, saturating in the decentralized case, while the centralized approach maintains u_g relatively close to the nominal value. While, the decentralized and BAS approaches achieve significantly smaller costs for the u_g actuator, the reduction in u_g forces the u_i and u_{ai} actuators to increase as seen in Figs. 6.7 and 6.8. The

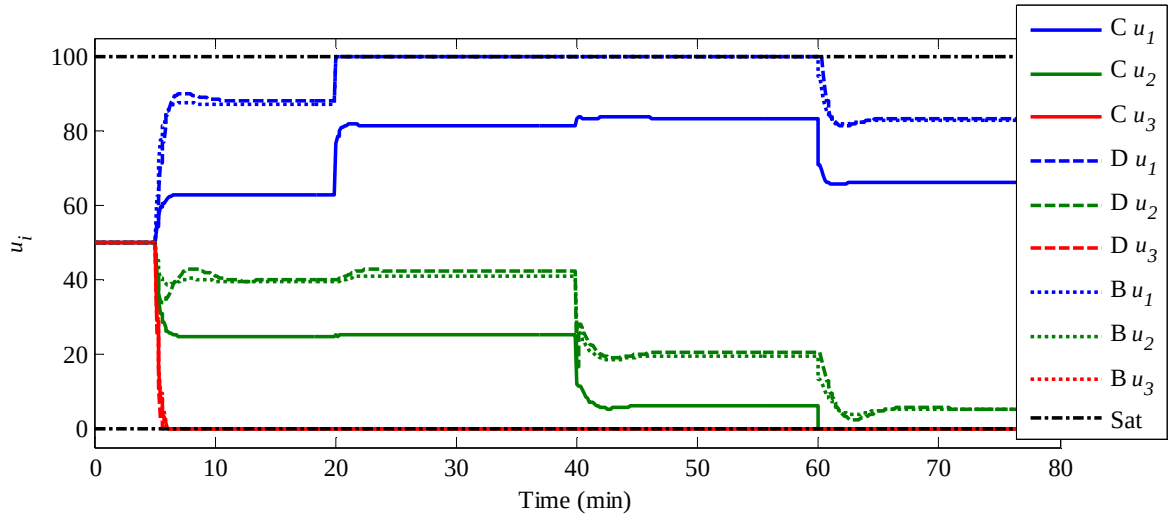

Figure 6.7 Actuator inputs u_i from each control architecture for Scenario 1.

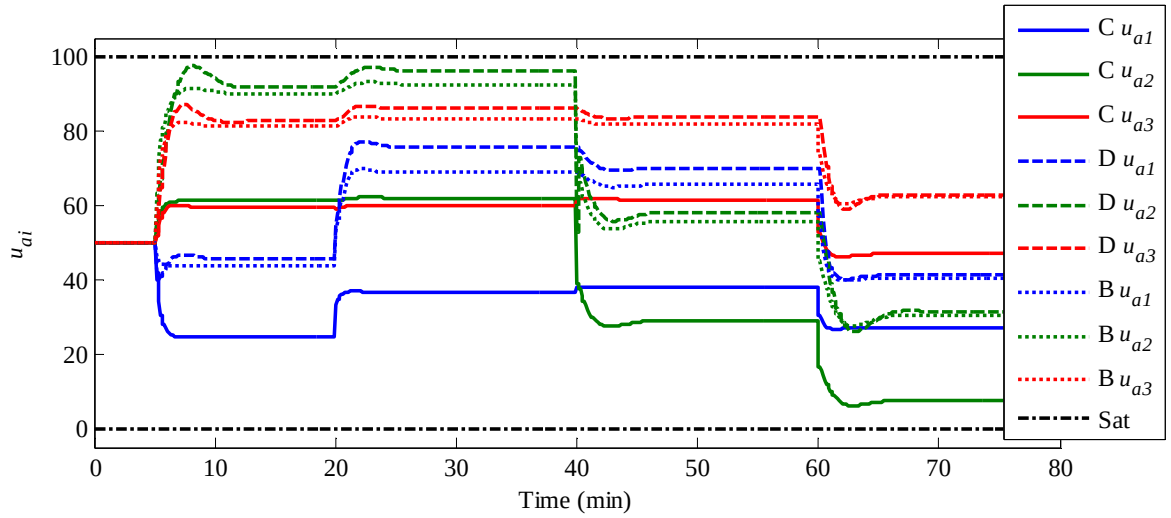

Figure 6.8 Actuator inputs u_{ai} from each control architecture for Scenario 1.

increase in actuator costs associated with the increase in u_i and u_{ai} outweighs the reduction in cost from the reduction of u_g, thus the increase in the total actuator cost.

While the BAS approach did perform slightly better than the decentralized approach in terms of meeting both the performance and efficiency objectives, it is surprising that the BAS approach did not perform better given the fact that the coupling terms in the A and B are directly considered when making control decisions. From Figs. 6.7, 6.8, and 6.9, it is clear that there is not a unique combination of actuator inputs u_g, u_i, and u_{ai} which can achieve the

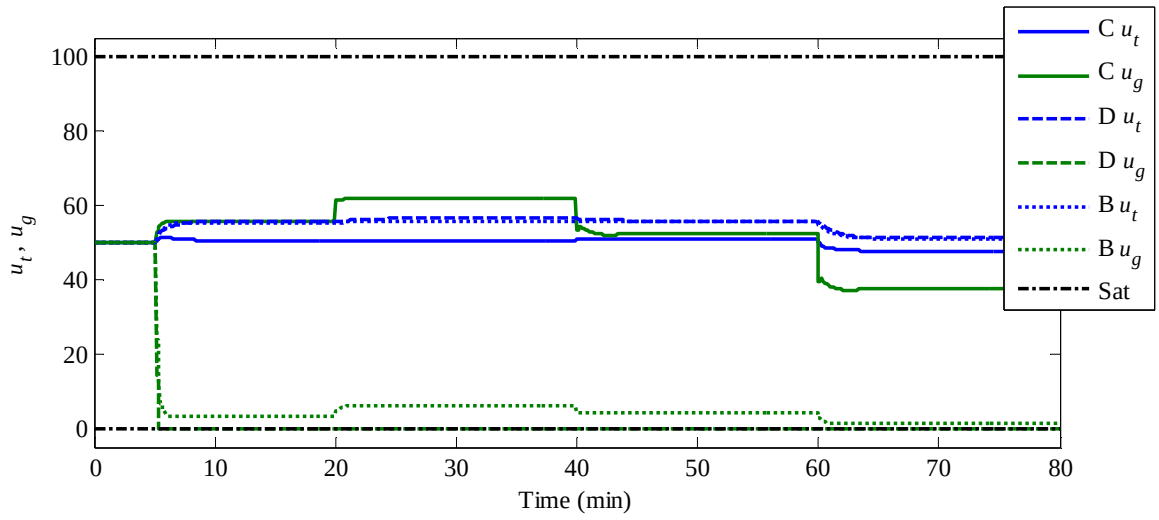

Figure 6.9 Actuator inputs u_t and u_g from each control architecture for Scenario 1.

performance objectives. However, different combinations of these actuators do result in different system efficiencies. With the decentralized control approach, each controller tries to minimize the actuator costs for only its corresponding subsystem actuators. This results in an equilibrium far from the optimal solution achieved by the centralized approach. The BAS approach is able to perform slightly better because each of the BAS controllers tries to find a balance which minimizes the common actuator costs and the costs of the actuators for the corresponding subsystem $\mathbf{S}_i$. The reason the BAS controller does not perform closer to the centralized controller is due to the fact that each controller only considers the trade-off between the common actuators and a set of actuators corresponding only to a single subsystem $\mathbf{S}_i$ and not the set of actuators for all the subsystems. With this insight, a simple modification to the BAS approach can significantly improve its efficiency performance. When calculating the matrices Q_a and q_b used in the MPC formulation, multiplying the actuator cost coefficients a_i and b_i, for the only actuators u_i and u_{ai}, by the number of subsystems N ($N = 3$ for the example system) can cause the BAS approach to perform much more like the centralized approach. This is because each of the BAS controllers must now determine the trade-off between the common actuator costs and the costs corresponding to 3 times the actuator costs for a single subsystem. While each subsystem is different, this multiplication better approximates the effect of the common actuators on all of the actuators u_i and u_{ai}. This modification has a slight negative

effect on the performance objectives but significantly improves the system efficiency as shown in Fig. 6.10. Now, after integrating the total actuator costs over the entire simulation, the cost for the BAS approach is only 4% greater than that of the centralized approach. Fig. 6.11 shows that the control signal for u_g is now much closer to that from the centralized approach. If the subsystems $\mathbf{S}_i$ were more similar, this modification would provide even greater improvement.

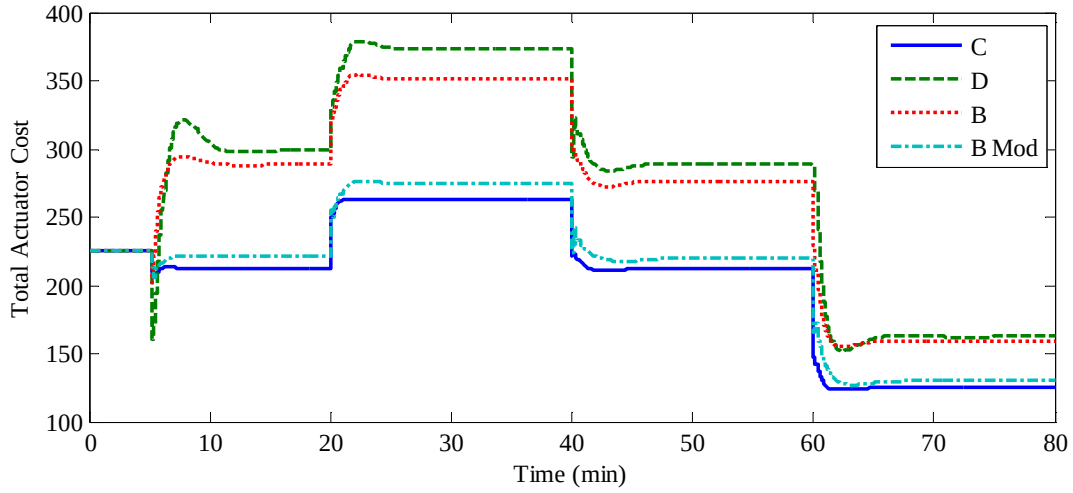

Figure 6.10 Improved system efficiency achieved by the modified BAS controller for Scenario 1.

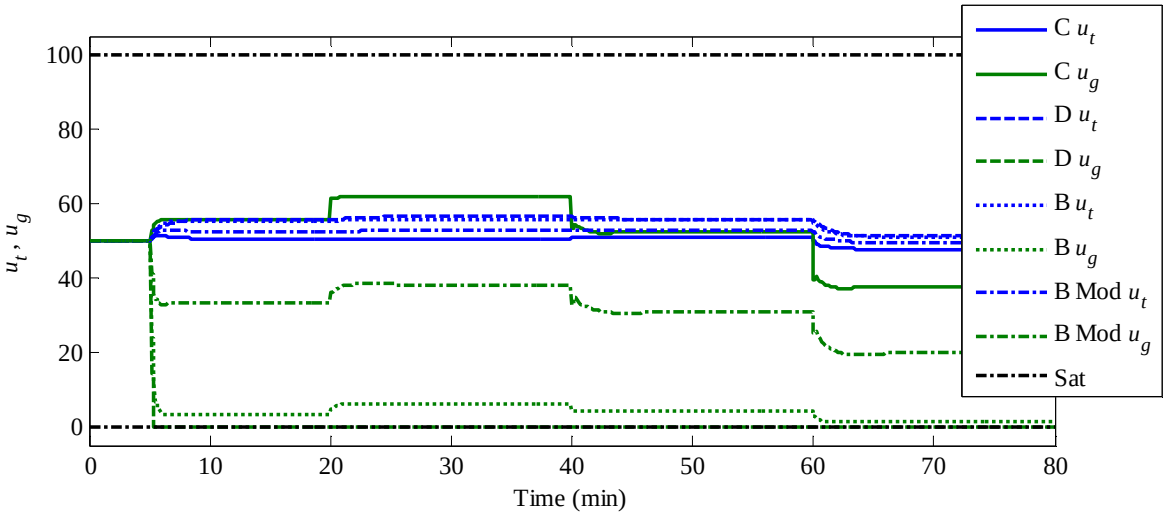

Figure 6.11 Change in u_g when using the modified BAS controller for Scenario 1.

6.2.1.2 Scenario 2

Scenario 2 presents a situation where not all of the control architectures are able to meet the performance objectives due to actuator constraints. Fig. 6.12 shows the disturbances I_{Li} and V_h over the 80 minute simulation for Scenario 2. The disturbances are the same as for Scenario 1 with the exception of I_{L1} which is increased to 3.5 instead of 3.2. This increase in I_{L1} causes both actuators u_1 and u_{a1} to saturate in the decentralized control approach as seen in Figs. 6.13 and 6.14. With both actuators for $\mathbf{S}_1$ saturated, the performance objective for V_{a1} cannot be met and V_{a1} deviates far from the desired value as seen in Fig. 6.15.

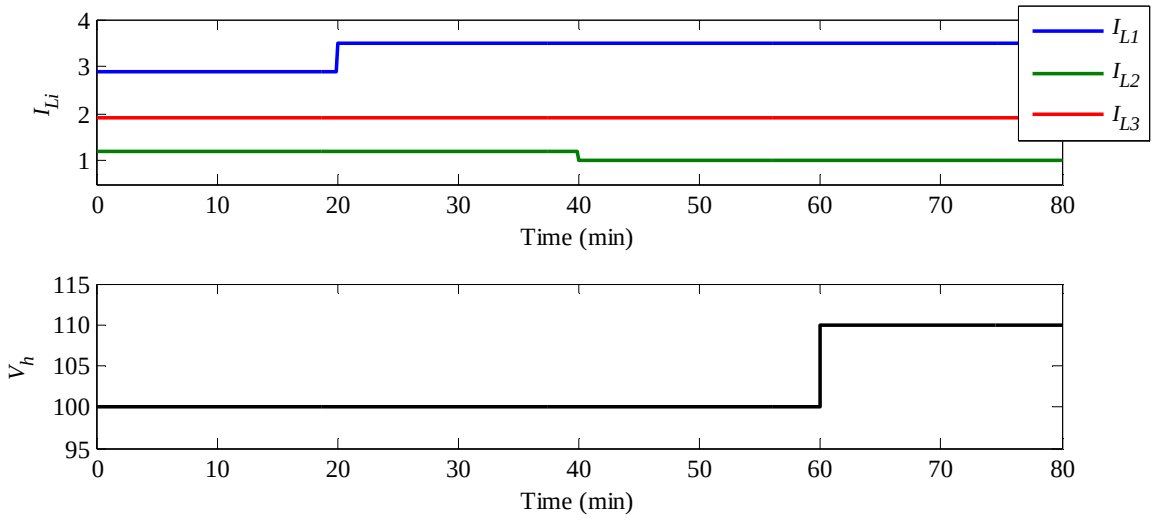

Figure 6.12 System disturbances I_{Li} and V_h for Scenario 2.

This scenario highlights one of the major benefits of the BAS control structure. From Scenario 1, the decentralized and unmodified BAS control approaches performed very similarly. However, now when the decentralized approach is unable to meet the performance objectives, the unmodified BAS approach remains able to meet the objective despite that fact that both u_1 and u_{a1} saturate. Looking at the common actuator inputs in Fig. 6.16, the major difference between the BAS and decentralized approach is the value of u_g. When the disturbance I_{L1} changes 20 minutes into the simulation, the BAS control architecture is able to increase the u_g input significantly in order to help meet the local performance for $\mathbf{S}_1$. Since the performance

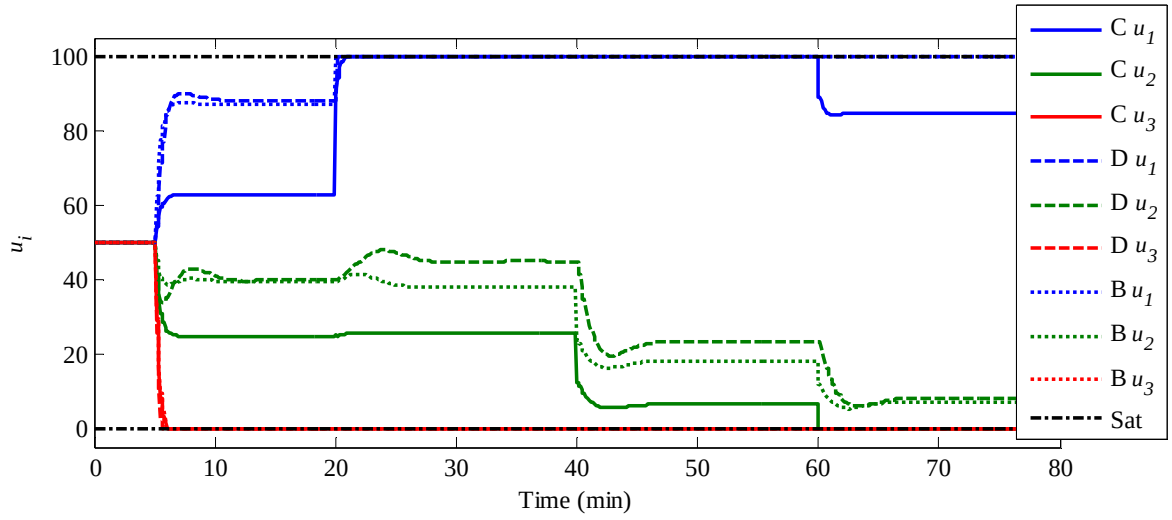

Figure 6.13 Actuator inputs u_i from each control architecture for Scenario 2.

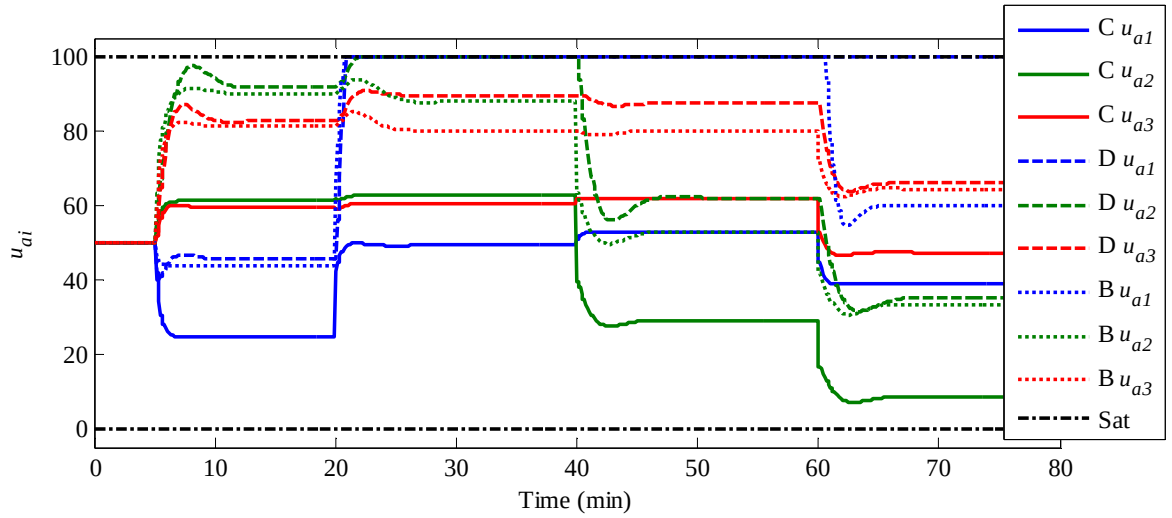

Figure 6.14 Actuator inputs u_{ai} from each control architecture for Scenario 2.

objectives for $\mathbf{S}_0$ and $\mathbf{S}_1$ are considered by the MPC controller for $\mathbf{S}_1$, the BAS approach is able to utilize the common subsystem actuators to help meet the objectives for $\mathbf{S}_i$. The decentralized control approach, however, does not know the coupling between the subsystems $\mathbf{S}_0$ and $\mathbf{S}_i$ and therefore, is unable to utilize the common actuators to meet the objectives for $\mathbf{S}_i$.

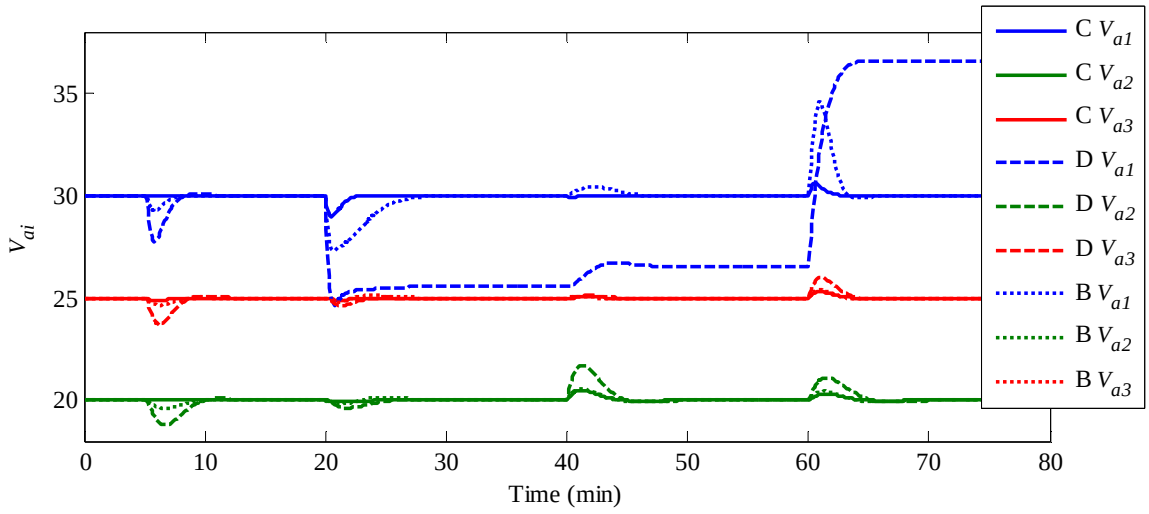

Figure 6.15 Ability of each control architecture to meet local performance objectives by tracking the desired value for V_{ai} for Scenario 2.

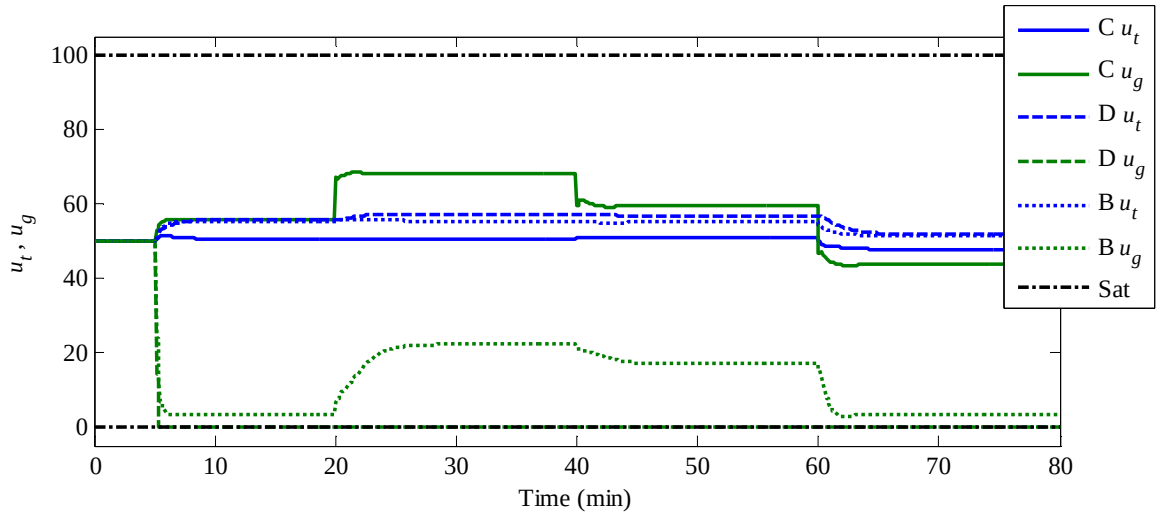

Figure 6.16 Actuator inputs u_t and u_g from each control architecture for Scenario 2.

6.2.2 Augmented Controller Performance via ESC

From Section 6.2.1, it is clear that the BAS control architecture is able to provide improved control performance when compared to the decentralized approach both in terms of meeting performance objectives as well as maximizing system efficiency. Additionally, the BAS approach is able to achieve significantly greater efficiency by modifying the actuator costs. However, if the subsystems are too dissimilar, this modification may not provide a significant

efficiency improvement. If the BAS approach cannot be modified, the efficiency of the approach may be significantly lower than that of the centralized approach. Fortunately, extremum seeking control can be used to augment the BAS control architecture to improve the system efficiency. Following the formulation from Chapter 5, the BAS control architecture from the previous section was modified to include an ESC controller. The ESC parameters used for the controller are presented in Table 6.2.

Table 6.2 ESC Parameters

Parameter	Value
ω	$\pi/100$ rad.
a	5
k	3ω
ϕ	$15\pi/180$ rad.
ω_h	10ω
ω_l	0.1ω

Due to the fact that ESC is an adaptive control technique and is relatively slow to adapt following a disturbance, the following figures show the results for a 240 minute simulation, where the same disturbances as Scenario 1 from above are used but are stretched in time by a factor of 3, as shown in Fig. 6.17.

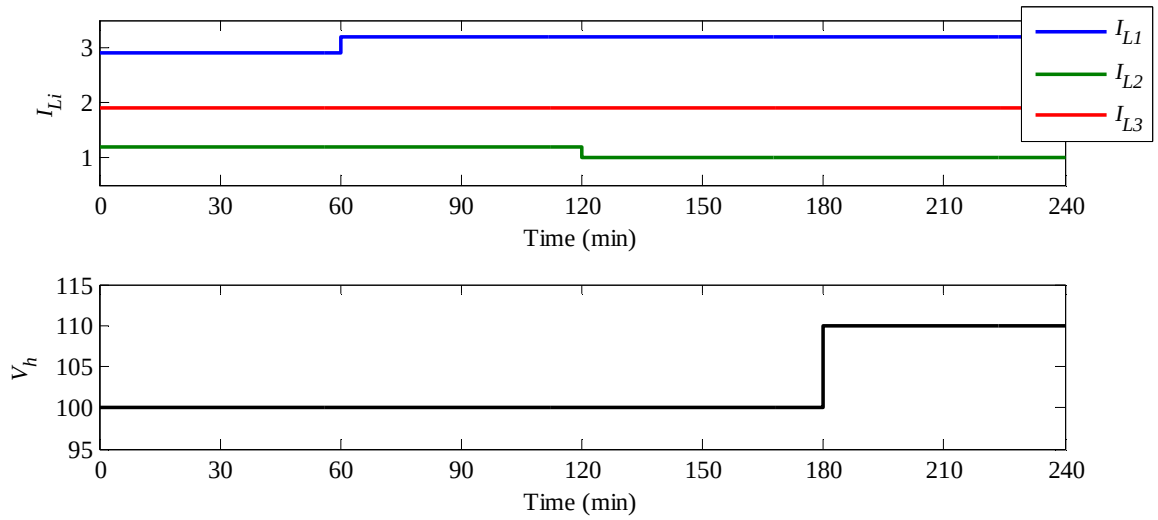

Figure 6.17 System disturbances I_{Li} and V_h for ESC Scenario.

First, from Fig. 6.18, it is clear that the perturbation from the ESC degrades the ability of the MPC controllers to meet the performance objectives. The majority of the oscillations in V_{ai} are small and are centered about the desired reference value. However, the disturbance in V_h at 180 minutes causes a rather large oscillation in V_{a2} which does not settle out until 45 minutes later. The reason for this is actuator saturation of u_2 and u_{a2} during the transient that follows the disturbance in V_h, as seen in Figs. 6.19 and 6.20. These figures show that the centralized solution saturates u_2 and brings u_{a2} close to saturation. From these figures, it is clear that the ESC approach often overshoots the centralized solution but then converges very close to the centralized solution. After the disturbance in V_h, the ESC approach overshoots and saturates u_{a2} until approximately 220 minutes when both u_{a2} and u_g, seen in Fig. 6.21, converge to the centralized solution.

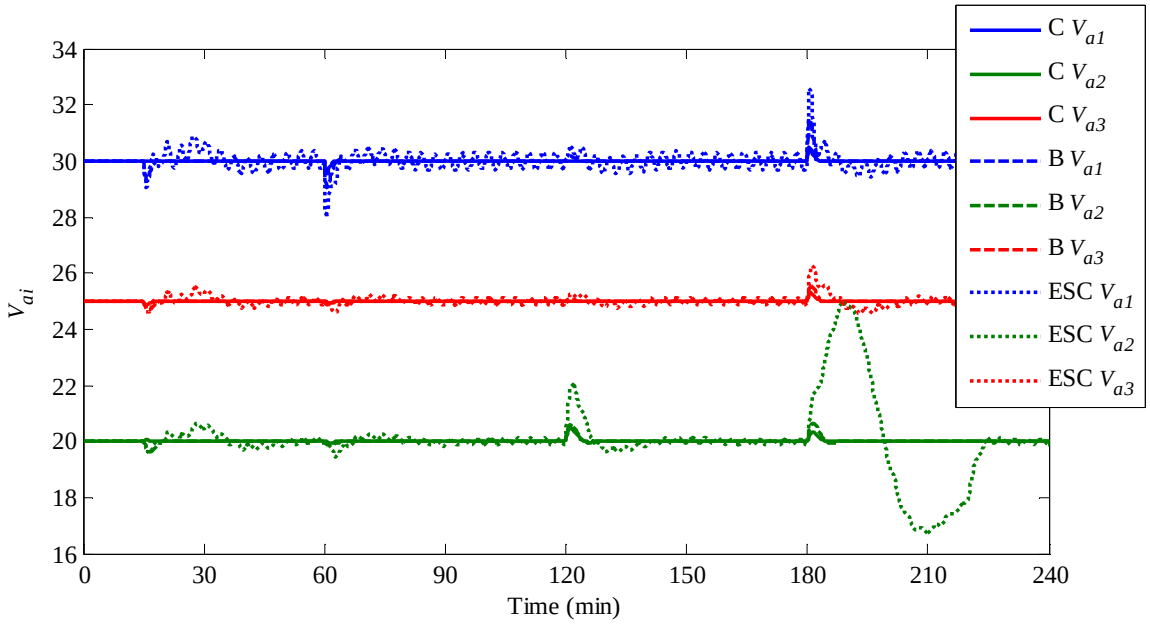

Figure 6.18 Ability of each control architecture to meet local performance objectives by tracking the desired value for V_{ai} for ESC Scenario.

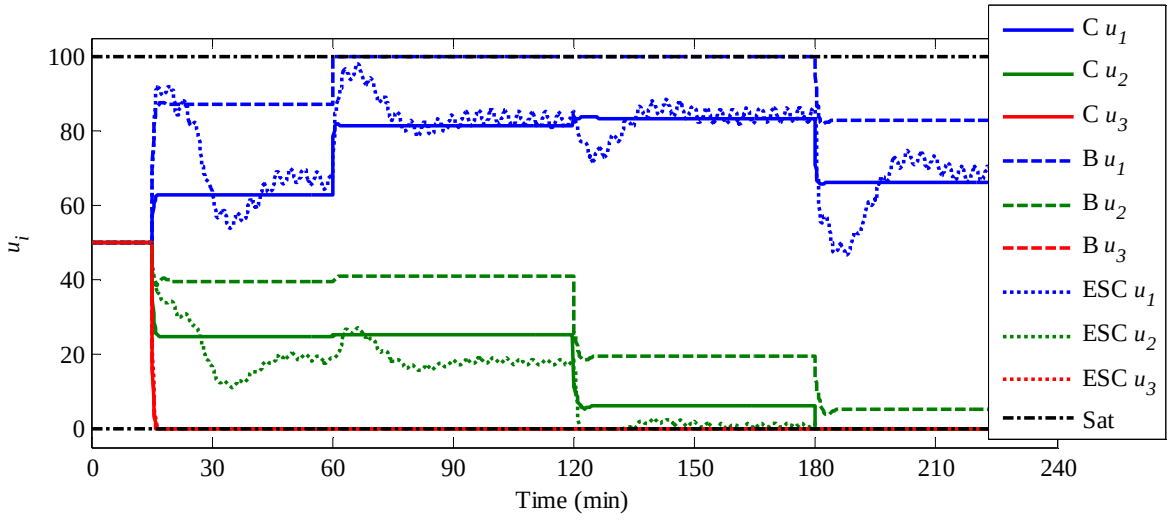

Figure 6.19 Actuator inputs u_i from each control architecture for Scenario 1.

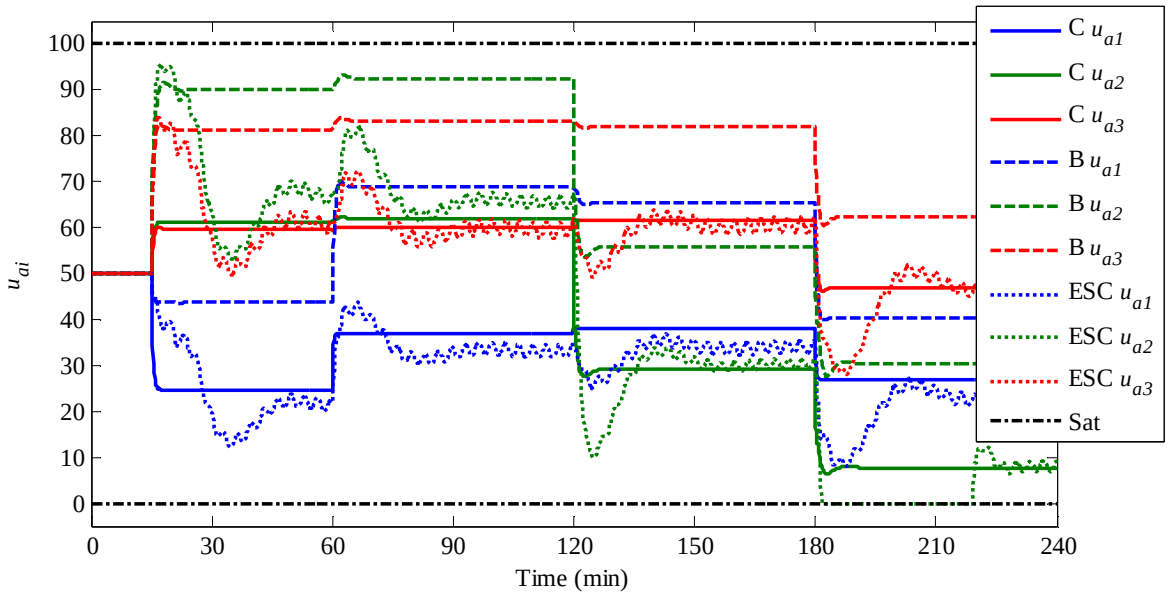

Figure 6.20 Actuator inputs u_{ai} from each control architecture for Scenario 1.

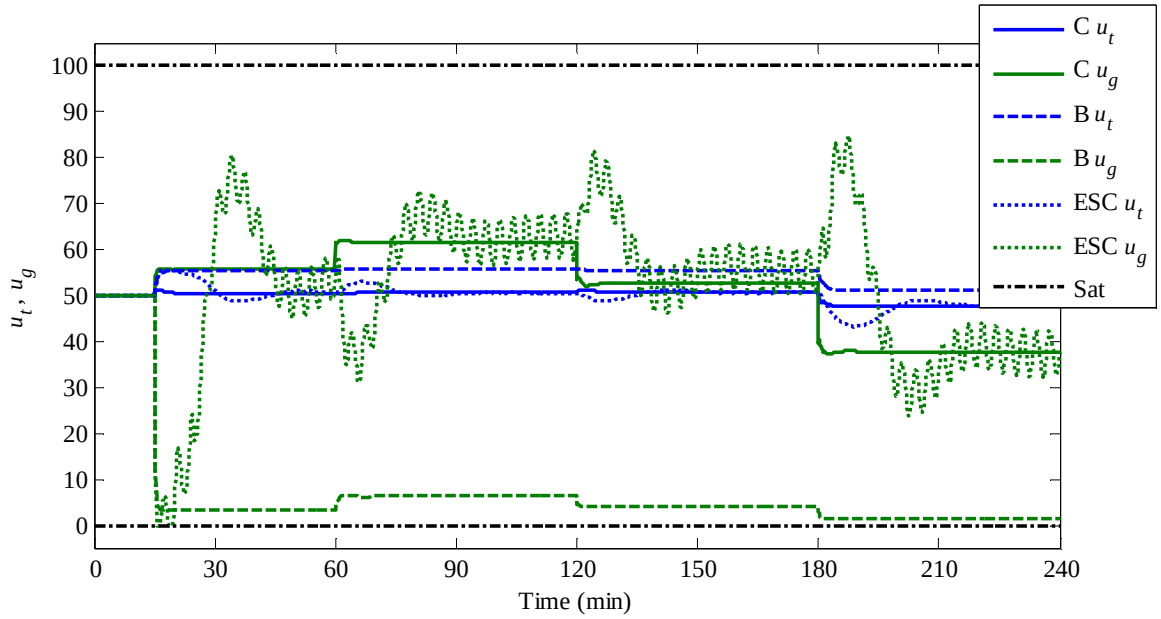

Figure 6.21 Actuator inputs u_t and u_g from each control architecture for Scenario 1.

From Figs. 6.19, 6.20, and 6.21, it is clear that the ESC augmentation is able to drive the actuator inputs of the BAS approach toward the inputs from the centralized solution. This results in a drastic improvement in the system efficiency as seen in Fig. 6.22. Fifteen minutes into the simulation when the controller are activated, Fig. 6.22 shows that initially the total actuator costs for the BAS and the ESC augmented BAS approaches are very similar and are both much larger than that of the centralized approach. However, after 30 minutes into the simulation, the ESC controller has augmented the control inputs enough as to drive down the actuator costs, converging very close to the centralized solution. For each disturbance to the system, the ESC controller is able to significantly reduce the actuator cost after a short transient. When comparing the integral of the total actuator costs, the unmodified BAS approach results in a 30% higher cost than the centralized approach. The BAS approach with the ESC augmentation, however, is only 5% higher. Therefore, the ESC augmentation presents a trade-off. Fig. 6.18 shows that the ESC approach can adversely affect the ability to meet the performance objectives during a transient, while Fig. 6.22 shows that the ESC augmentation can significantly improve the total system efficiency.

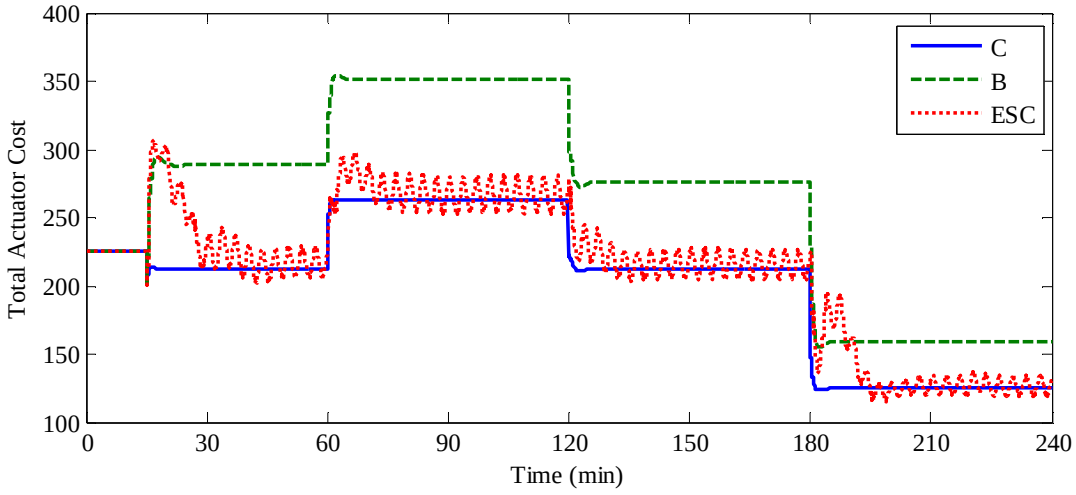

Figure 6.22 Improved system efficiency achieved by the modified BAS controller for Scenario 1.

6.2.3 Scalability

Sections 6.2.1 and 6.2.2 have demonstrated the differences between the various control architectures in terms of the ability to meet the performance and efficiency objectives for the system. When developing control strategies for large-scale systems, the scalability of the approach also needs to be considered. In this case, scalability refers to the associated communication and computational costs for a given control architecture. Fig. 5.1 showed that the centralized control approach is the least scalable and the decentralized approach is the most. The BAS (and the ESC augmented BAS) approach falls between these two. This section quantifies the communication and computation cost for the example system to demonstrate this result.

6.2.3.1 Communication

A centralized control strategy requires knowledge of all system states when determining the control decision for any given input. For some systems, especially large-scale system which may consist of subsystems which are spatially distributed, the communication of all these states may be expensive and restrict the ability to use a centralized approach. Additionally, a centralized approach is very sensitive to faults, where a sensor failure has the ability to adversely affect the control decisions for the entire system. Alternatively, decentralized control requires

very little communication since only local state information is used to make control decisions. This also makes decentralized control more robust to faults, where a sensor failure only affects the control decisions for the corresponding subsystem and not the entire system. The BAS control approach requires more communication than the decentralized approach but less communication than the centralized approach. The BAS architecture only requires the communication of the common subsystem states to each of the N subsystems. The fact that the states of the N subsystems are not communicated to each other makes the BAS approach more robust to faults than the centralized approach.

6.2.3.2 Computation

One of the major motivations for a decentralized control approach is the reduced computational costs when compared to a centralized solution. In practice, excessive computational costs can make a centralized approach infeasible; thus, the motivation for decentralized control. Dividing the centralized control problem into several smaller problems accomplishes two things; each control problem has fewer states and inputs resulting in less computational cost and each control problem may be solved in parallel allowing each problem to be solved on a separate processor. Having multiple processors can also reduce communication costs since each processor can be physically located near the corresponding subsystem. The reduction in computational costs is now demonstrated for the example system.

With the centralized control approach for the example system, the system model has 8 states and 8 inputs. Additionally, a control horizon of 15 time steps and a prediction horizon of 30 time steps are used by the MPC formulation. This results in a large optimization problem where the lifted control vector $U \in \mathbb{R}^{120}$ is solved for every 10 seconds. For the decentralized control approach, the centralized control problem is broken into 4 smaller control problems where each has only 2 states and 2 inputs, resulting in a lifted control vector $U \in \mathbb{R}^{30}$. Similarly, the BAS control approach, decomposes the central control problem into 3 smaller control problems, where each has 4 states and 4 inputs, resulting in a lifted control vector $U \in \mathbb{R}^{60}$. Fig. 6.23 shows the computational time required at each sample time for Scenario 1 from Section 6.2.1. Note that the y-axis is a log scale and since the decentralized and BAS approaches have multiple MPC controllers, the largest computation time at each sample time is shown in the

figure. The maximum computation time for the centralized, decentralized, and BAS approaches are 2.089, 0.299, and 0.317 seconds, respectively. The average computation time for each is 0.036, 0.005, 0.010 seconds, respectively. With the maximum computation time for the centralized solution nearly an order of magnitude larger than that of the decentralized and BAS solutions, it is clear why a centralized approach may not be feasible in some applications.

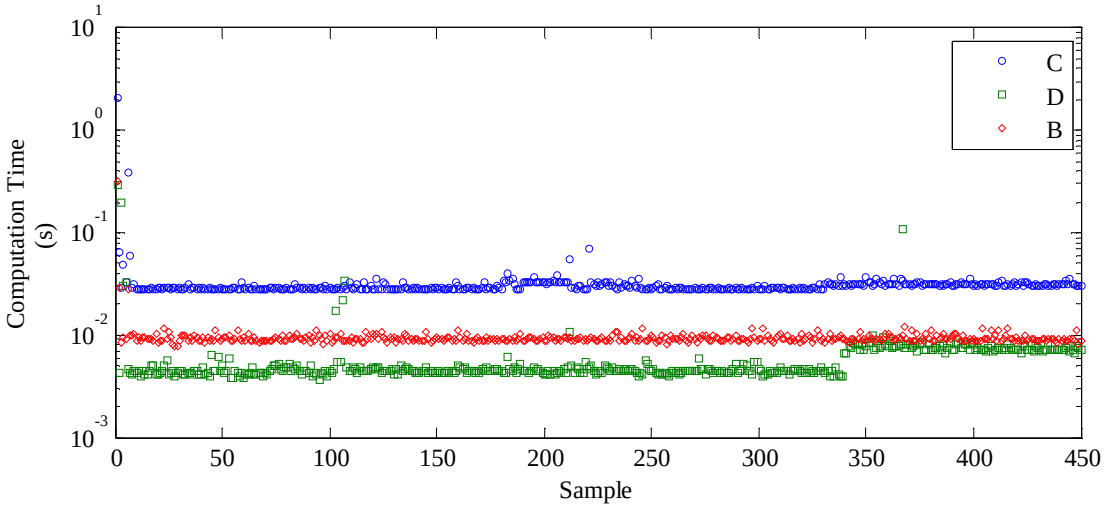

Figure 6.23 Computation time for each control approach for Scenario 1.

For a BAS system, as the number of subsystems $\mathbf{S}_i$ increases, the computational cost for the centralized solution grows rapidly. However, for the BAS control architecture the computational costs simply grow linearly with the number of subsystems and as previously mentioned, the BAS approach can take advantage of parallel processing. If parallel processing is used, the computation cost for each processor remains constant and only the number of processors increases as the number of subsystems increases. Fig. 6.24 shows the growth in computational costs for the centralized and BAS control strategies for an increasing number of subsystems. The data in this figure was generated using a system very similar to the example system with the slight difference of having identical subsystems. With each subsystem having 2 states and 2 inputs and the MPC having a control horizon of 15 steps and a prediction horizon of 30 steps, the number of subsystems $\mathbf{S}_i$ was varied between 2 and 30. The two plots show the same data using linear and logarithmic y-axes. The BAS data assumes the computations are performed on a single processor and, therefore, the computational costs grow on the order of N.

However, for the centralized approach the computation costs grow on the order of N^2. While the computational costs are similar when there are few subsystems, once the number of subsystems reaches 30, the centralized computation cost is over an order of magnitude greater than that of the BAS approach. Clearly the BAS control approach scales significantly better than the centralized approach.

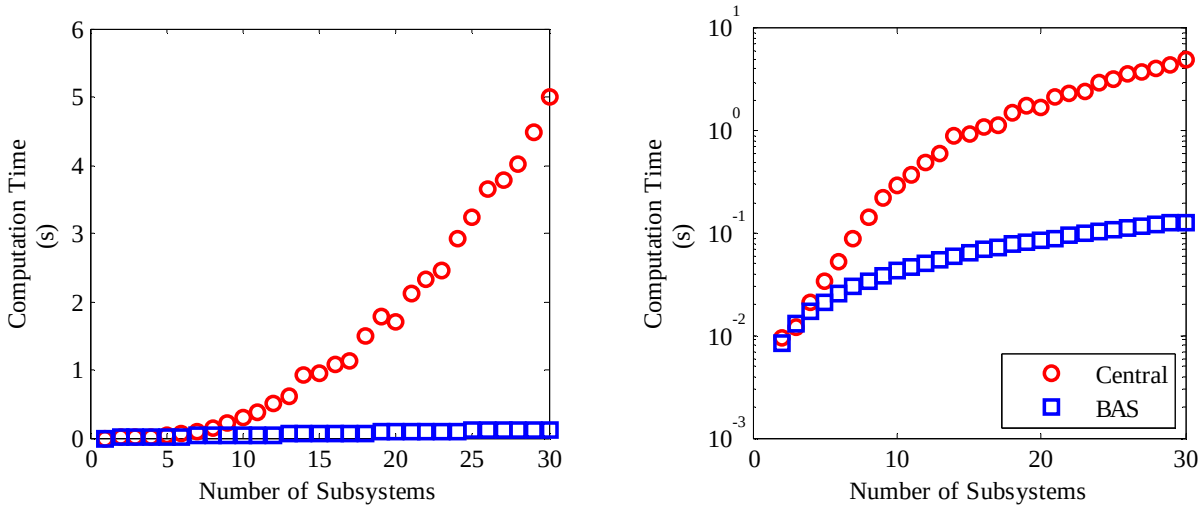

Figure 6.24 Computation time for each control approach for Scenario 1.

This chapter has demonstrated the advantages of the BAS and ESC control strategies for a linear example system. The following chapter evaluates these control strategies for a more realistic system which includes additional challenges such as unknown disturbances and system nonlinearity.

Chapter 7
VRF System

The example system developed in Chapter 2 has been used to demonstrate the methods and ideas presented through this thesis, however, these techniques must be applicable to real-world systems in order to be of value. While there are many different systems that can naturally be modeled with a BAS structure, this chapter demonstrates how the BAS control approach can be applied to variable refrigerant flow (VRF) systems. Section 7.1 provides a detailed background of previous modeling and control efforts for VRF systems and motivates the need for improved control strategies. A gray-box modeling approach is developed in Section 7.2 and the state-space BAS structure of VRF systems is presented in Section 7.3. Model validation is performed in Section 7.4. The control architecture and controller design are developed in Section 7.5 and simulation results are presented in Section 7.6, which are used to evaluate the control performance. Appendix B contains the MATLAB code used to generate and control the VRF system.

7.1 VRF Background

VRF systems utilize the vapor compression cycle (VCC) to transfer heat from one location to another. VRF systems are also known as multi-evaporator vapor compression systems (ME-VCS) and are becoming widely used to provide the air-conditioning and refrigeration needs for buildings. A single VRF system can remove heat from multiple rooms or zones of a building and reject this heat to the outdoor environment. Prior to analyzing VRF systems in greater detail, it is important to have an understanding of VCC basics. Fig. 7.1a shows a four component single-evaporator vapor compression system (VCS) consisting of a

compressor, condenser, electronic expansion valve (EEV), and evaporator. The corresponding VCC is shown on a pressure-enthalpy (P-h) diagram in Fig. 7.1b. Low temperature, low pressure refrigerant vapor enters the compressor at (1), at which point the refrigerant is compressed, causing a drastic increase in pressure and temperature. The refrigerant vapor at (2) then enters the condenser where the high temperature refrigerant loses heat to the lower temperature air passing through the heat exchanger. As the refrigerant loses heat, the refrigerant condenses from a vapor into a liquid and is typically completely liquid by the time the refrigerant exits the condenser at (3). Then, the refrigerant enters the EEV, where the refrigerant is suddenly expanded causing a quick drop in pressure and temperature. This expansion turns the liquid refrigerant entering the EEV into a two-phase mixture of vapor and liquid. This mixture at (4) enters the evaporator where the refrigerant absorbs heat from the warmer air passing through the heat exchanger. This absorption of heat cause the remainder of the liquid to evaporate and by the time the refrigerant exits the evaporator it is entirely vapor and the cycle repeats. Through this process, a VCS system is able to remove heat from a room or space (the low temperature reservoir) and expel the heat to the high temperature reservoir even if the temperature of this environment is significantly higher than the temperature of the room or space.

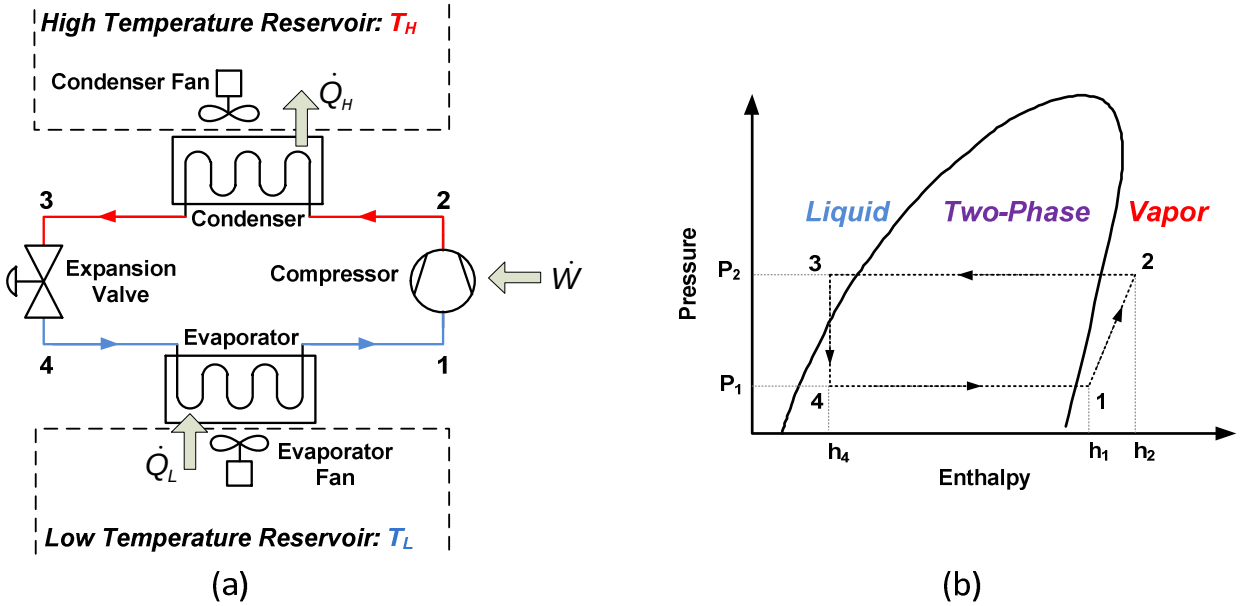

Figure 7.1 Single-evaporator VCS – a) 4 component system schematic, b) P-h diagram of vapor compression cycle.

VCSs are of particular interest due to their relatively high efficiencies. VCSs are heavily embedded in today's society and are used to meet a large variety of cooling needs ranging from household refrigerators and air-conditioning systems to aircraft and large-scale data centers. Therefore, there has been extensive research into the modeling and control of these systems. A detailed review of previous research efforts can be found in [30].

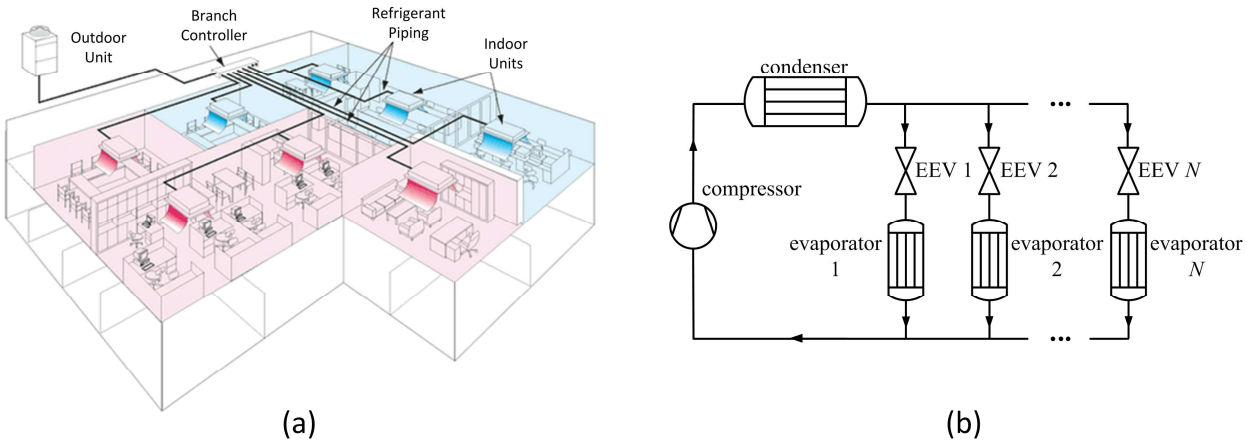

Figure 7.2 VRF system – a) Industrial system used to heat and cool multiple rooms in a building [31], b) schematic of VRF system.

VRF systems are very similar to the single-evaporator VCS system shown in Fig. 7.1. While still using a single compressor and condenser, a VRF system has multiple EEVs and multiple evaporators which allow a single system to directly cool multiple rooms. Fig. 7.2a shows how a VRF system is used to directly cool (or heat) multiple rooms in a building and Fig. 7.2b shows a schematic of an N evaporator VRF system. With multiple evaporators, the total refrigerant flow rate from the compressor is divided into N different flow paths after exiting the condenser. The EEV apertures are used to determine how much of the total refrigerant flow is sent to each evaporator. This division of refrigerant flow rate is a major source of coupling in the system which needs to be considered when developing a decentralized control strategy. An additional source of coupling comes from the joining of refrigerant flows downstream of the evaporators. Prior to entering the compressor, the refrigerant flows from each evaporator are combined in a junction, at which point the refrigerant flows converge to a single pressure. While each evaporator can operate at a different pressure, these pressures are highly coupled due to the

combination of flows at the junction. Further discussion and analysis of these sources of coupling can be found in [7].

Significantly less research has focused on the modeling and control of multi-evaporator systems. Due to the high degree of coupling between evaporators, the majority of multi-evaporator control research utilizes Multiple-Input Multiple-Output (MIMO) control strategies which rely on a linear model of the system to make control decisions. Thus, the first challenge to controlling VRF systems is developing an appropriate linear model. Most of the previous work has used black-box models developed using traditional data-driven system identification techniques. Unfortunately, black-box modeling techniques suffer from several disadvantages, the most important of which is the fact that they are not scalable (with respect to the number of evaporators in the system) since the identified model is specific to the system configuration at the time data is gathered. Section 7.2 develops a gray-box modeling approach (based on the fluid dynamic modeling effort from [32]) for both the fluid and thermal dynamics of a N evaporator VRF system.

Once an appropriate linear model is identified, the second challenge is to design a control architecture which is scalable to VRF systems with a large number of evaporators. Several control strategies have been proposed in the literature for dual- and triple-evaporator VRF systems. Model-based cascaded control approaches for dual- and triple-evaporator systems are presented in [33] and [34], which take advantage of the time scale separation between the refrigerant thermal dynamics and the room air thermal dynamics. A linear-quadratic regulator (LQR) approach is used to control a dual-evaporator system in [35] and is one of the few efforts in which the model used for control came from the linearization of a nonlinear model of the system as opposed to a black-box approach. A decentralized hierarchical control approach for a dual-evaporator system is developed in [36]. At the lower level, decentralized MPC controllers use the EEV and secondary fluid flow rate to control cooling capacity and superheat for each evaporator. The compressor speed and discharge valve aperture are controlled using proportional-integral (PI) controllers to regulate the evaporator pressures. At the higher level, a global controller, using MPC, determines the cooling capacity and pressure set points for each evaporator in order to maximize system efficiency. While these control approaches have been demonstrated for dual- and triple-evaporator system, a scalable control architecture for arbitrarily

large VRF systems is still needed. The BAS control approach developed in Chapter 5 is applied to a 5-evaporator VRF system in Section 2.4 and the ability of this control approach to meet both the performance and efficiency objectives for these systems is demonstrated in Section 7.6.

7.2 System Modeling

In this thesis, a simulated VRF system is used in place of an experimental system. While experimental work is of interest and will be the focus of future work, the simulated system used for the following modeling and control efforts captures a wide-range of system dynamics and features found in the physical systems. The VRF system is modeled and simulated using the AFRL (Air Force Research Laboratory) Transient Thermal Modeling and Optimization (ATTMO) toolbox [37], which is based on the Thermosys toolbox [38] from the University of Illinois at Urbana-Champaign (UIUC). ATTMO is a Simulink® based toolbox which uses a modular approach wherein each component of a VCS is modeled independently. The dynamic heat exchanger models use a lumped parameter moving boundary approach to model the condenser with three refrigerant fluid zones (superheat, two-phase, subcooled) and the evaporator with two refrigerant fluid zones (two-phase, superheat). Each component calculates its own refrigerant outlet enthalpy. The heat exchanger and flow junction models calculate the system pressures and the compressor and valve models calculate the refrigerant mass flow rates. For this study, the heat exchangers are of the tube-and-fin configuration and the secondary fluid is air. Validation efforts for ATTMO can be found in [37] and validation efforts for Thermosys can be found in [38], which uses a very similar modeling approach. ATTMO is able to capture both the nonlinear and transient dynamics of VCC systems and, therefore, serves as an appropriate substitute for an experimental system. ATTMO, however, does not simulate the signal noise found in experimental system and, thus, the effect of signal noise on control performance is left for future work. ATTMO serves an efficient platform to conduct control design and analysis due to its ability to simulate systems significantly faster than real-time. With speed-ups ranging from 10x to well over 100x (depending on the operating conditions and the degree of the transient behavior), ATTMO offers drastic reductions in control development time. Please refer to [37] for additional details.

7.2.1 Fluid Dynamics

Fig. 7.3 shows the electrical circuit schematic used to represent the fluid dynamics for the VRF system. Table 7.1 lists the corresponding electrical analogue for each component of the fluid system. The schematic depicts a generic N evaporator system which results in $N+2$ differential equations. These equations represent the dynamics for the condenser pressure P_c, the N evaporator pressures P_i, and the junction pressure P_q downstream of the evaporators and are written as

$$C_c \dot{P}_c = \dot{m}_k' - \dot{m}_k + \kappa_c \left(\dot{Q}_{rwc} - \dot{Q}_{wac} \right), \tag{7.1}$$

$$C_i \dot{P}_i = \dot{m}_i - \dot{m}_i' + \kappa_i \left(\dot{Q}_{awi} - \dot{Q}_{wri} \right) \ \forall \ i \in \mathcal{N}, \tag{7.2}$$

$$C_q \dot{P}_q = \sum_{i=1}^{n} \dot{m}_i' - \dot{m}_k'. \tag{7.3}$$

Note that (7.1) and (7.2) include several heat transfer rate terms denoted by $\dot{Q}$ which will be described in detail in the following section. Additionally, $\dot{P}$ represents a derivative state which should not be confused with refrigerant mass flow rate $\dot{m}$ or heat transfer rate $\dot{Q}$, which are algebraic quantities of the system. For notational purposes $i \in \mathcal{N} = \{1, 2, ..., N\}$. It is important to note that all quantities used to model the fluid and thermal dynamics for the system are deviations from a nominal operating condition.

The compressor is modeled as a current source and provides a refrigerant mass flow rate $\dot{m}_k'$ to the inlet of the compressor. This mass flow rate, modeled as

$$\dot{m}_k' = \beta_{k1} P_q - \beta_{k2} P_c + \beta_{k3} \omega_k - \beta_{k4} T_{rq}, \tag{7.4}$$

is a function of the junction and condenser pressures, P_q and P_c, the compressor speed ω_k (a control input to the system), and the refrigerant temperature of the junction T_{rq}, which is a dynamic state of the thermal model presented in the next section. The parameters β_{k1}, β_{k2}, β_{k3}, and β_{k4} are positive linearization coefficients.

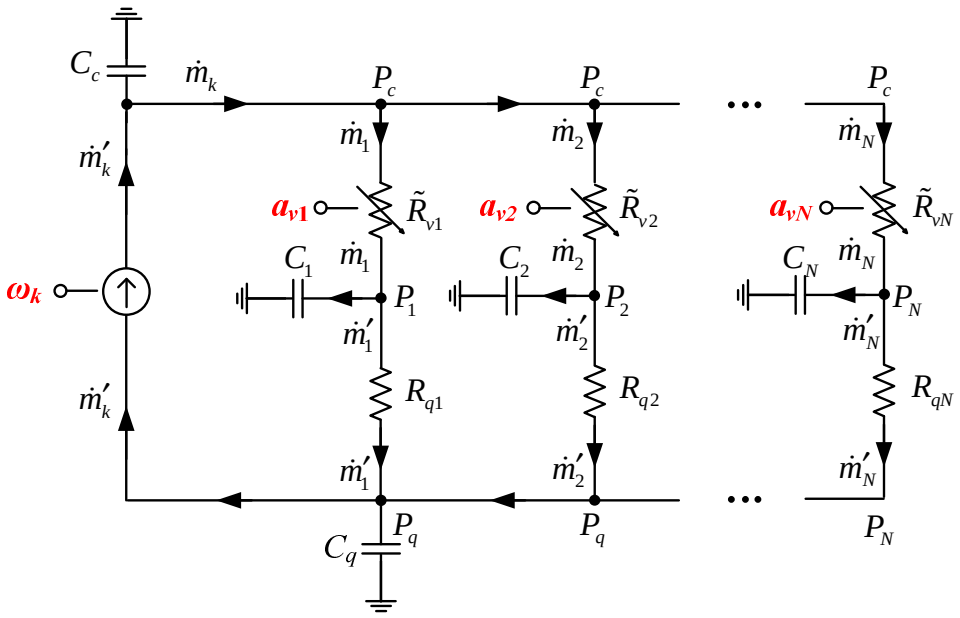

Figure 7.3 Electrical circuit diagram of the fluid dynamics of a VRF system. The symbol ○— indicates the presence of a control input.

Table 7.1 Electrical Circuit Analogues for Fluid Dynamics of VRF Systems

VRF System	**Electrical Circuit Analogue**
refrigerant mass flow rate	electric current
pressure differential	voltage potential
fluid resistance	resistor
compressor	current source
EEV	variable resistor
condenser/evaporators	capacitor
fluid junction	capacitor

Each of the N EEVs is modeled as a variable resistor whose resistance $\tilde{R}_{vi}$ is a function of the EEV aperture a_{vi} (a control input to the system). The pressure drop across the EEV is

$$P_c - P_i = \tilde{R}_{vi}\left(a_{vi}\right)\dot{m}_i \;\forall\, i \in \mathcal{N}. \tag{7.5}$$

This equation is linearized as

$$P_c - P_i = R_{vi}\dot{m}_i - K_{vi}a_{vi} \;\; \forall \; i \in \mathcal{N}, \tag{7.6}$$

so that the pressure drop across the EEV is a linear combination of the refrigerant flow rate through the valve and the valve aperture. Positive constants R_{vi} and K_{vi} arise from the linearization and R_{vi} can be thought of as the nominal resistance of the valve.

With each evaporator potentially operating at a distinct pressure, there is a pressure drop after the evaporator prior to the joining of refrigerant flows at the junction near the inlet of the compressor, resulting in a common pressure P_q. The pressure drop downstream of each evaporator is modeled as a resistance, resulting in

$$P_i - P_q = R_{qi}\dot{m}_i' + K_{q1i}\omega_k + K_{q2i}\omega_{fi} - K_{q3i}T_{ri} \;\; \forall \; i \in \mathcal{N}. \tag{7.7}$$

Note that several terms in addition to a fixed resistance R_{qi} are needed in order to accurately model this pressure drop. The pressure drop is also a function of the compressor speed ω_k and the evaporator fan speed ω_{fi}, which are inputs to the system, as well as the temperature of the refrigerant in the evaporator T_{ri}, which comes from the thermal model in the following section.

Finally, based on Kirchoff's current law, there is a mass flow conservation equation corresponding to the refrigerant flow split following the condenser which is given by

$$\dot{m}_k = \sum_{i=1}^{N} \dot{m}_i. \tag{7.8}$$

7.2.2 Thermal Dynamics

Fig. 7.4 shows the electrical circuit schematic used to represent the thermal dynamics for the VRF system. Table 7.2 lists the corresponding electrical analogue for each component of the thermal system. For the N evaporator system there are $2N+2$ differential equations used to represent the thermal dynamics of the system. First, the dynamics of the lumped wall temperatures for the condenser and each evaporator are described by

$$C_{wc}\dot{T}_{wc} = \dot{Q}_{rwc} - \dot{Q}_{wac}, \tag{7.9}$$

$$C_{wi}\dot{T}_{wi} = \dot{Q}_{awi} - \dot{Q}_{wri} \;\; \forall \; i \in \mathcal{N}. \tag{7.10}$$

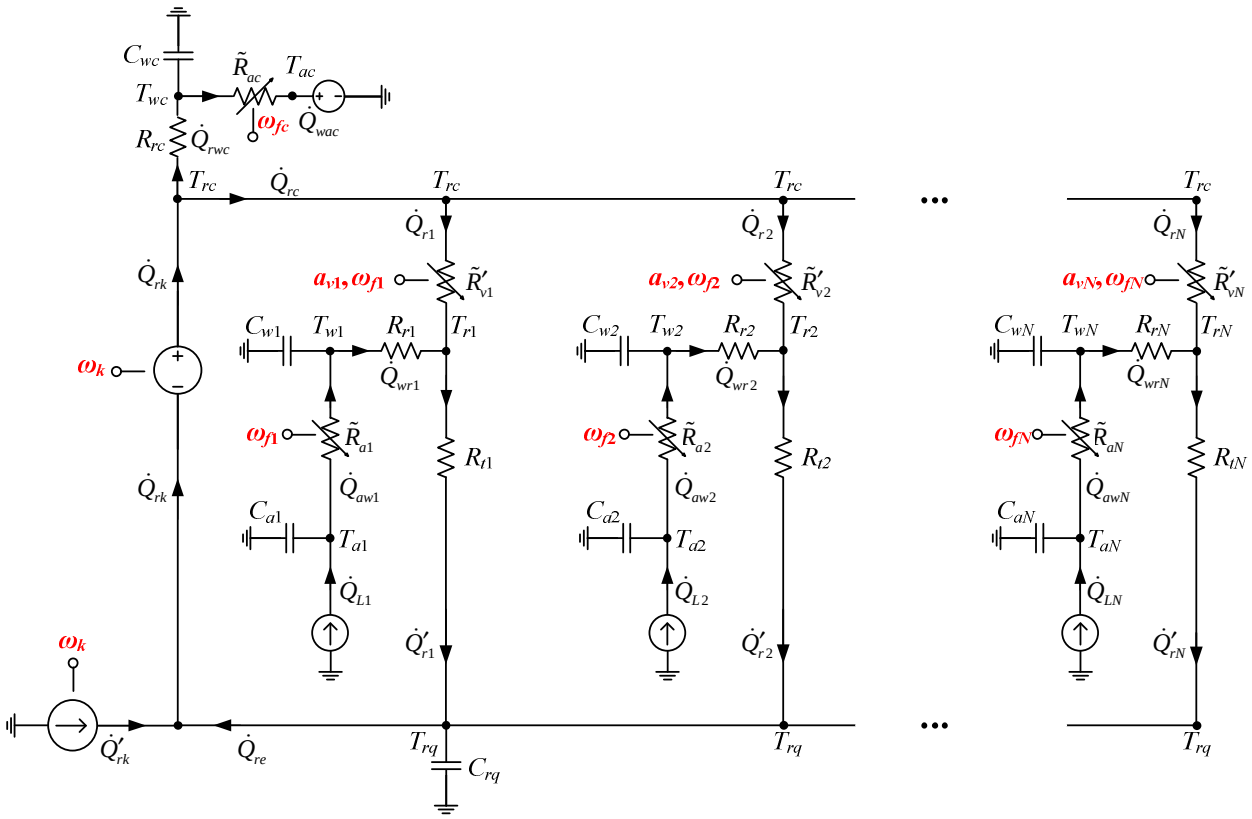

Figure 7.4 Electrical circuit diagram of the thermal dynamics of a VRF system. The symbol ○— indicates the presence of a control input.

Table 7.2 Electrical Circuit Analogues for Thermal Dynamics of VRF Systems

VRF System	Electrical Circuit Analogue
heat transfer rate	electric current
temperature differential	voltage potential
thermal resistance	resistor
compressor	voltage source and current source
EEV	variable resistor
condenser/evaporator tube wall	thermal capacitor
thermal junction	thermal capacitor

Once again, it is important to remember the $\dot{T}$ is a derivative state and should not be confused with the heat transfer rate $\dot{Q}$, which is an algebraic quantity. The temperature T_{rq} of the

refrigerant in the junction downstream of the evaporators is also a dynamic state of the system and is given by

$$C_{rq}\dot{T}_{rq} = \sum_{i}^{n} \dot{Q}'_{ri} - \dot{Q}_{re} - \kappa_{rq}\sum_{i}^{n} \dot{m}'_{i} + \kappa_{rq}\dot{m}'_{k}. \tag{7.11}$$

The constant C_{rq} represents the thermal capacitance of the refrigerant inside the junction and κ_{rq} is a positive linearization coefficient. The remaining N differential equations capture the dynamics of the air temperatures T_{ai} inside each of the N rooms cooled by the VRF system and are written as

$$C_{ai}\dot{T}_{ai} = \dot{Q}_{Li} - \dot{Q}_{awi} \;\; \forall \; i \in \mathcal{N}, \tag{7.12}$$

where $\dot{Q}_{Li}$ is an unknown thermal load for each room and acts as a disturbance to the system. For this study, it is assumed that there is no heat transfer between the rooms cooled by the VRF system and the effects of thermal coupling between rooms is left for future work.

Each heat exchanger has an air-side thermal resistance and a refrigerant-side thermal resistance. Starting with the air-side resistance, the temperature difference between the air and wall of the heat exchangers varies as a function of the heat exchanger fan speed and is written as

$$T_{wc} - T_{ac} = \dot{Q}_{wac}\tilde{R}_{fc}\left(\omega_{fc}\right), \tag{7.13}$$

$$T_{ai} - T_{wi} = \dot{Q}_{awi}\tilde{R}_{fi}\left(\omega_{fi}\right) \;\; \forall \; i \in \mathcal{N}. \tag{7.14}$$

These equations are linearized such that the temperature difference is a linear function of the heat transfer rate and the heat exchanger fan speed (a control input to the system), shown as

$$T_{wc} - T_{ac} = R_{ac}\dot{Q}_{wac} - K_{fc}\omega_{fc}, \tag{7.15}$$

$$T_{ai} - T_{wi} = R_{ai}\dot{Q}_{awi} - K_{fi}\omega_{fi} \;\; \forall \; i \in \mathcal{N}. \tag{7.16}$$

Both R_{ai} and K_{fi} are positive coefficients arising from the linearization and R_{ai} can be thought of as the nominal thermal resistance between the air and heat exchanger wall. The same is true for R_{ac} and K_{fc} of the condenser.

For the refrigerant side, the thermal resistance is modeled as a function of the refrigerant mass flow rate:

$$T_{rc} - T_{wc} = \dot{Q}_{rwc}\tilde{R}_{rc}\left(\dot{m}_k'\right), \tag{7.17}$$

$$T_{wi} - T_{ri} = \dot{Q}_{wri}\tilde{R}_{ri}\left(\dot{m}_i\right) \ \forall \ i \in \mathcal{N}. \tag{7.18}$$

Once again, these equations are linearized so that the temperature difference is a linear function of the heat transfer rate and the refrigerant mass flow rate through the heat exchanger. From the parameter identification process presented in Section 7.4, it was found that the temperature difference for the condenser was also a strong function of the condenser pressure and, therefore, the equations are

$$T_{rc} - T_{wc} = R_{rc}\dot{Q}_{rwc} - \lambda_{rc1}\dot{m}_k' + \lambda_{rc2}P_c, \tag{7.19}$$

$$T_{wi} - T_{ri} = R_{ri}\dot{Q}_{wri} - \lambda_{ri}\dot{m}_i \ \forall \ i \in \mathcal{N}. \tag{7.20}$$

Again, the coefficients are all positive and arise from the linearization.

The lumped temperature of the refrigerant in each heat exchanger is approximated as

$$T_{rc} = \eta_{rc}P_c, \tag{7.21}$$

$$T_{ri} = \eta_{r1i}P_i + \eta_{r2i}\omega_{fi} - \eta_{r3i}\dot{m}_i \ \forall \ i \in \mathcal{N} \tag{7.22}$$

The refrigerant temperatures T_{rc} and T_{ri} are not made to be dynamic states of the system due to their strong dependence on the pressures P_c and P_i, which are dynamic states of the fluid system.

The compressor is modeled as both a current source and a voltage source. The compressor not only adds thermal energy to the refrigerant, it also increases the refrigerant temperature through the compression process, thus the need to model the compressor as a current and voltage source. The additional heat transfer rate from the compressor is given by

$$\dot{Q}_{rk}' = \alpha_{k1}T_{rc} - \alpha_{k2}T_{rq} + \alpha_{k3}\omega_k, \tag{7.23}$$

which is a function of the compressor speed as well as the refrigerant temperatures at the inlet and outlet of the compressor. The total heat transfer rate of the refrigerant at the exit of the compressor is

$$\dot{Q}_{rk} = \dot{Q}_{re} + \dot{Q}'_{rk} = \gamma_{k1} T_{rc} - \gamma_{k2} T_{rq} + \gamma_{k3} \omega_k + \gamma_{k4} P_q - \gamma_{k5} P_c. \tag{7.24}$$

Each EEV is modeled as a variable resistor with a thermal resistance $\tilde{R}'_{vi}$ which is a function of the valve aperture and the evaporator fan speed, both of which are inputs to the system. The temperature drop across the valve is given by

$$T_{rc} - T_{ri} = \tilde{R}'_{vi}\left(a_{vi}, \omega_{fi}\right)\dot{Q}_{ri} \;\; \forall\, i \in \mathcal{N}, \tag{7.25}$$

and is linearized, resulting in

$$T_{rc} - T_{ri} = R'_{vi}\dot{Q}_{ri} - K'_{vi}a_{vi} - K'_{ki}\omega_{fi} \;\; \forall\, i \in \mathcal{N}, \tag{7.26}$$

where R'_{vi}, K'_{vi}, and K'_{ki} are positive coefficients from the linearization and R'_{vi} can be thought of as the nominal thermal resistance of the EEV.

The temperature change downstream of the evaporators is modeled as

$$T_{ri} - T_{rq} = \mu_{m1i}\dot{Q}'_{ri} - \mu_{m2i}\dot{m}'_i - \mu_{m3i}\dot{Q}_{re} + \mu_{m4i}\dot{m}'_k + \mu_{m5i}\omega_{fi} \;\; \forall\, i \in \mathcal{N}, \tag{7.27}$$

and it is important to note that T_{rq} may be higher or lower than T_{ri} depending on the operating conditions of the system.

Finally, two energy conservation equations can be written for the VRF system; one for the refrigerant flow split downstream of the condenser and the other for the compressor, which are given by

$$\dot{Q}_{rc} = \sum_{i=1}^{N} \dot{Q}_{ri}, \tag{7.28}$$

$$\dot{Q}_{rk} = \dot{Q}'_{rk} + \dot{Q}_{re}. \tag{7.29}$$

7.3 Complete System Model

With the dynamic and algebraic equations developed in Section 7.2, the N evaporator VRF system can be represented in state-space form. First, a subsystem representation is used to divide the VRF system into $N+1$ subsystems. The i^{th} EEV and i^{th} evaporator make up the i^{th} subsystem $\mathbf{S}_i$ and the compressor and condenser combine to make up an additional subsystem $\mathbf{S}_0$. Combining the fluid and thermal dynamics for the EEV and evaporator, the state, input, and disturbance vectors for $\mathbf{S}_i$ are $x_i = \begin{bmatrix} P_i & T_{wi} & T_{ai} \end{bmatrix}^T$, $u_i = \begin{bmatrix} a_{vi} & \omega_{fi} \end{bmatrix}^T$, and $d_i = \begin{bmatrix} \dot{Q}_{Li} \end{bmatrix}$. The state, inputs, and disturbance vectors for $\mathbf{S}_0$ are $x_0 = \begin{bmatrix} P_c & P_q & T_{wc} & T_{rq} \end{bmatrix}^T$, $u_0 = \begin{bmatrix} \omega_k & \omega_{fc} \end{bmatrix}$, and $d_0 = \begin{bmatrix} T_{ac} \end{bmatrix}$.

Before developing the subsystem representations, it is important to note that the fluid dynamics are represented by a *closed* system. Here closed refers to the fact that the refrigerant mass flow rate $\dot{m}$ is conserved throughout the system and cannot enter or exit the system. This is contrary to the example system presented in the previous chapters, where current can enter the system through the voltage source in $\mathbf{S}_0$ and exit through the current sinks in $\mathbf{S}_i$. This refrigerant mass flow rate conservation imposes an algebraic relationship between the dynamic states of the system. Using (7.1), (7.2), (7.3), (7.8), (7.9), and (7.10), it is clear that $\dot{P}_q$ can be written as

$$C_q \dot{P}_q = \sum_{i=1}^{n} \left[-C_i \dot{P}_i + \kappa_i C_{wi} \dot{T}_{wi} \right] - C_c \dot{P}_c + \kappa_c C_{wc} \dot{T}_{wc}, \tag{7.30}$$

which results in

$$C_q P_q = \sum_{i=1}^{n} \left[-C_i P_i + \kappa_i C_{wi} T_{wi} \right] - C_c P_c + \kappa_c C_{wc} T_{wc} + \gamma, \tag{7.31}$$

where γ is a constant of integration and depends on the initial state of the system, assumed here to equal 0. As discussed in Section 2.3, relationships between parameters in the A and B matrices of a system can create a situation where a system is not controllable despite being structurally controllable. The algebraic relationship from (7.31) creates such a situation, where if

a state-space representation were developed containing P_q, P_i, T_{wi}, P_c, and T_{wc} as states, the system would not be controllable. Therefore, (7.31) is used to rewrite the T_{wc} state as a function of the other states in the system, thus, with a slight abuse of notation, the state vector for $\mathbf{S}_0$ is reduced to $x_0 = \begin{bmatrix} P_c & P_q & T_{rq} \end{bmatrix}^T$.

The subsystem representation for $\mathbf{S}_i$ is written as

$$\mathbf{S}_i : \dot{x}_i = A_{ii}x_i + A_{i0}x_0 + B_{ii}u_i + B_{i0}u_0 + V_{ii}d_i \;\; \forall i \in \mathcal{N}, \tag{7.32}$$

where the matrices A_{ii}, A_{i0}, B_{ii}, B_{i0}, and V_{ii} are

$$A_{ii} = \begin{bmatrix} a_{ii\_11} & a_{ii\_12} & a_{ii\_13} \\ a_{ii\_21} & a_{ii\_22} & a_{ii\_23} \\ 0 & a_{ii\_32} & a_{ii\_33} \end{bmatrix}, \;\; A_{i0} = \begin{bmatrix} a_{i0\_11} & a_{i0\_12} & 0 \\ a_{i0\_21} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{7.33}$$

$$B_{ii} = \begin{bmatrix} b_{ii\_11} & b_{ii\_12} \\ b_{ii\_21} & b_{ii\_22} \\ 0 & b_{ii\_32} \end{bmatrix}, \;\; B_{i0} = \begin{bmatrix} b_{i0\_11} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{7.34}$$

$$V_{ii} = \begin{bmatrix} 0 \\ 0 \\ v_{ii\_3} \end{bmatrix}. \tag{7.35}$$

Due to the complexity of the terms in each of these matrices, Table 7.3 is used to present these terms in a single location. Table 7.4 includes additional terms which are too large to fit in Table 7.3. The subsystem representation for $\mathbf{S}_0$ is written as

$$\mathbf{S}_0 : \dot{x}_0 = A_{00}x_0 + \sum_{i=1}^{N} A_{0i}x_i + B_{00}u_0 + \sum_{i=1}^{N} B_{0i}u_i + V_{00}d_0, \tag{7.36}$$

where the matrices A_{00}, A_{0i}, B_{00}, B_{0i}, and V_{00} are

$$A_{00} = \begin{bmatrix} a_{00\_11} & a_{00\_12} & a_{00\_13} \\ a_{00\_21} & a_{00\_22} & a_{00\_23} \\ a_{00\_31} & a_{00\_32} & a_{00\_33} \end{bmatrix}, \;\; A_{0i} = \begin{bmatrix} a_{0i\_11} & a_{0i\_12} & 0 \\ a_{0i\_21} & 0 & 0 \\ a_{0i\_31} & 0 & 0 \end{bmatrix}, \tag{7.37}$$

$$B_{00} = \begin{bmatrix} b_{00\_11} & b_{00\_12} \\ b_{00\_21} & 0 \\ b_{00\_31} & 0 \end{bmatrix},\ B_{0i} = \begin{bmatrix} b_{0i\_11} & 0 \\ b_{0i\_21} & b_{0i\_22} \\ b_{0i\_31} & b_{0i\_32} \end{bmatrix}, \tag{7.38}$$

$$V_{00} = \begin{bmatrix} v_{00\_1} \\ 0 \\ 0 \end{bmatrix}. \tag{7.39}$$

Table 7.3 VRF System Matrix Elements

a_{ii_11}	See Table 7.4	a_{00_11}	See Table 7.4
a_{ii_12}	$-\frac{\kappa_i}{C_i}\left(\frac{1}{R_{ai}}+\frac{1}{R_{ri}}\right)$	a_{00_12}	$\frac{\beta_{k1}}{C_c}\left(1+\frac{\kappa_c\lambda_{rc1}}{R_{rc}}\right)-\frac{\kappa_c}{C_c}\left(\frac{1}{R_{rc}}+\frac{1}{R_{ac}}\right)\frac{C_q}{\kappa_c C_{wc}}$
a_{ii_13}	$\frac{\kappa_i}{C_i R_{ai}}$	a_{00_13}	$-\frac{\beta_{k4}}{C_c}\left(1+\frac{\kappa_c\lambda_{rc1}}{R_{rc}}\right)$
a_{ii_21}	$\frac{1}{C_{wi}R_{ri}}\left(\eta_{r1i}+\frac{\lambda_{ri}+\eta_{r3i}}{R_{vi}}\right)$	a_{00_21}	$\frac{1}{C_q}\left(\beta_{k2}-\sum_{i=1}^{N}\frac{K_{q3i}\eta_{r3i}}{R_{qi}R_{vi}}\right)$
a_{ii_22}	$-\frac{1}{C_{wi}}\left(\frac{1}{R_{ai}}+\frac{1}{R_{ri}}\right)$	a_{00_22}	$-\frac{1}{C_q}\left(\beta_{k1}+\sum_{i=1}^{N}\left(\frac{1}{R_{qi}}\right)\right)$
a_{ii_23}	$\frac{1}{C_{wi}R_{ai}}$	a_{00_23}	$\frac{\beta_{k4}}{C_q}$
a_{ii_32}	$\frac{1}{C_{ai}R_{ai}}$	a_{00_31}	See Table 7.4
a_{ii_33}	$-\frac{1}{C_{ai}R_{ai}}$	a_{00_32}	See Table 7.4
a_{i0_11}	$\frac{1}{C_i R_{vi}}\left(1+\frac{K_{q3i}\eta_{r3i}}{R_{qi}}-\frac{\kappa_i}{R_{ri}}(\eta_{r3i}+\lambda_{ri})\right)$	a_{00_33}	See Table 7.4
a_{i0_12}	$\frac{1}{C_i R_{qi}}$	a_{0i_11}	$\frac{1}{C_c R_{vi}}-\frac{\kappa_c}{C_c}\left(\frac{1}{R_{rc}}+\frac{1}{R_{ac}}\right)\frac{C_i}{\kappa_c C_{wc}}$
a_{i0_21}	$-\frac{\lambda_{ri}+\eta_{r3i}}{C_{wi}R_{ri}R_{vi}}$	a_{0i_12}	$\frac{\kappa_c}{C_c}\left(\frac{1}{R_{rc}}+\frac{1}{R_{ac}}\right)\frac{\kappa_i C_{wi}}{\kappa_c C_{wc}}$
b_{ii_11}	$\frac{K_{vi}}{C_i R_{vi}}\left(1+\frac{K_{q3i}\eta_{r3i}}{R_{qi}}-\frac{\kappa_i}{R_{ri}}(\lambda_{ri}+\eta_{r3i})\right)$	a_{0i_21}	$\frac{1}{C_q R_{qi}}\left(1+K_{q3i}\eta_{r1i}+\frac{K_{q3i}\eta_{r3i}}{R_{vi}}\right)$
b_{ii_12}	$\frac{1}{C_i}\left(\frac{K_{q2i}}{R_{qi}}-\frac{K_{q3i}\eta_{r2i}}{R_{qi}}+\frac{\kappa_i K_{fi}}{R_{ai}}+\frac{\kappa_i\eta_{r2i}}{R_{ri}}\right)$	a_{0i_31}	See Table 7.4

b_{ii_21}	$-\frac{K_{vi}\left(\lambda_{ri}+\eta_{r3i}\right)}{C_{wi}R_{ri}R_{vi}}$	b_{00_11}	$\frac{\beta_{k3}}{C_c}\left(1+\frac{\kappa_c\lambda_{rc1}}{R_{rc}}\right)$
b_{ii_22}	$\frac{1}{C_{wi}}\left(\frac{K_{fi}}{R_{ai}}+\frac{\eta_{r2i}}{R_{ri}}\right)$	b_{00_12}	$-\frac{\kappa_c K_{fc}}{C_c R_{ac}}$
b_{ii_32}	$-\frac{K_{fi}}{C_{ai}R_{ai}}$	b_{00_21}	$-\frac{1}{C_q}\left(\beta_{k3}+\sum_{i=1}^{N}\left(\frac{K_{q1i}}{R_{qi}}\right)\right)$
b_{i0_11}	$\frac{K_{q1i}}{C_i R_{qi}}$	b_{00_31}	See Table 7.4
v_{ii_3}	$\frac{1}{C_{ai}}$	b_{0i_11}	$-\frac{K_{vi}}{C_c R_{vi}}$
		b_{0i_21}	$-\frac{K_{q3i}\eta_{r3i}K_{vi}}{C_q R_{qi}R_{vi}}$
		b_{0i_22}	$-\frac{1}{C_q R_{qi}}\left(K_{q2i}-K_{q3i}\eta_{r2i}\right)$
		b_{0i_31}	See Table 7.4
		b_{0i_32}	See Table 7.4
		v_{00_1}	$\frac{\kappa_c}{C_c R_{ac}}$

Table 7.4 VRF System Matrix Elements Continued

a_{ii_11}	$-\frac{1}{C_i}\left(\frac{1}{R_{vi}}\left(1+\frac{K_{q3i}\eta_{r3i}}{R_{qi}}\right)+\frac{1}{R_{qi}}\left(1+K_{q3i}\eta_{r1i}\right)+\frac{\kappa_i}{R_{ri}}\left(-\eta_{r1i}+\frac{\eta_{r3i}}{R_{vi}}-\frac{\lambda_{ri}}{R_{vi}}\right)\right)$
a_{00_11}	$-\frac{1}{C_c}\left(\beta_{k2}+\sum_{i=1}^{N}\left(\frac{1}{R_{vi}}\right)+\frac{\kappa_c}{R_{rc}}\left(\lambda_{rc1}\beta_{k2}-\eta_{rc}+\lambda_{rc2}\right)\right)-\frac{\kappa_c}{C_c}\left(\frac{1}{R_{rc}}+\frac{1}{R_{ac}}\right)\frac{C_c}{\kappa_c C_{wc}}$
a_{00_31}	$\frac{1}{C_{rq}}\left[\begin{array}{l}\sum_{i=1}^{N}\left[\frac{1}{\mu_{m1i}}\left[\left(1+\frac{\mu_{m2i}K_{q3i}}{R_{qi}}\right)\left(-\frac{\eta_{r3i}}{R_{vi}}\right)+\mu_{m3i}\left(\gamma_{k1}\eta_{rc}-\gamma_{k5}-\alpha_{k1}\eta_{rc}\right)+\mu_{m4i}\beta_{k2}\right]+\frac{\kappa_{rq}K_{q3i}\eta_{r3i}}{R_{qi}R_{vi}}\right] \\ -\left(\gamma_{k1}\eta_{rc}-\gamma_{k5}-\alpha_{k1}\eta_{rc}\right)-\kappa_{rq}\beta_{k2}\end{array}\right]$
a_{00_32}	$\frac{1}{C_{rq}}\left[\sum_{i=1}^{N}\left[\frac{1}{\mu_{m1i}}\left[-\frac{\mu_{m2i}}{R_{qi}}+\mu_{m3i}\gamma_{k4}-\mu_{m4i}\beta_{k1}\right]+\frac{\kappa_{rq}}{R_{qi}}\right]-\gamma_{k4}+\kappa_{rq}\beta_{k1}\right]$

a_{00_33}	$\frac{1}{C_{rq}}\left[\sum_{i=1}^{N}\frac{1}{\mu_{m1i}}\left[-1-\mu_{m3i}\left(\gamma_{k2}-\alpha_{k2}\right)+\mu_{m4i}\beta_{k4}\right]+\left(\gamma_{k2}-\alpha_{k2}\right)-\kappa_{rq}\beta_{k4}\right]$
a_{0i_31}	$\frac{1}{C_{rq}}\left[\frac{1}{\mu_{m1i}}\left[\left(1+\frac{\mu_{m2i}K_{q3i}}{R_{qi}}\right)\left(\eta_{r1i}+\frac{\eta_{r3i}}{R_{vi}}\right)+\frac{\mu_{m2i}}{R_{qi}}\right]-\kappa_{rq}\left(\frac{1}{R_{qi}}+\frac{K_{q3i}}{R_{qi}}\left(\eta_{r1i}+\frac{\eta_{r3i}}{R_{vi}}\right)\right)\right]$
b_{00_31}	$\frac{1}{C_{rq}}\left[\sum_{i=1}^{N}\left[\frac{1}{\mu_{m1i}}\left[-\frac{\mu_{m2i}K_{q1i}}{R_{qi}}+\mu_{m3i}\left(\gamma_{k3}-\alpha_{k3}\right)-\mu_{m4i}\beta_{k3}\right]+\frac{\kappa_{rq}K_{q1i}}{R_{qi}}\right]+\left(\alpha_{k3}-\gamma_{k3}\right)+\kappa_{rq}\beta_{k3}\right]$
b_{0i_31}	$\frac{1}{C_{rq}}\left[\frac{1}{\mu_{m1i}}\left[\left(1+\frac{\mu_{m2i}K_{q3i}}{R_{qi}}\right)\left(-\frac{\eta_{r3i}K_{vi}}{R_{vi}}\right)\right]+\frac{\kappa_{rq}K_{q3i}\eta_{r3i}K_{vi}}{R_{qi}R_{vi}}\right]$
b_{0i_32}	$\frac{1}{C_{rq}}\left[\frac{1}{\mu_{m1i}}\left[\left(1+\frac{\mu_{m2i}K_{q3i}}{R_{qi}}\right)\eta_{r2i}-\frac{\mu_{m2i}K_{q2i}}{R_{qi}}-\mu_{m5i}\right]-\frac{\kappa_{rq}}{R_{qi}}\left(K_{q3i}\eta_{r2i}-K_{q2i}\right)\right]$

With the subsystem representation from (7.32) and (7.36), the complete N evaporator system can be represented as

$$\mathbf{S}: \dot{x} = Ax + Bu + Vd, \tag{7.40}$$

where $x = \begin{bmatrix} x_1 & x_2 & \dots & x_N & x_0 \end{bmatrix}^T$, $u = \begin{bmatrix} u_1 & u_2 & \dots & u_N & u_0 \end{bmatrix}^T$, and $d = \begin{bmatrix} d_1 & d_2 & \dots & d_N & d_0 \end{bmatrix}^T$ are the state, control, and disturbance vectors for the entire VRF system. We have $x \in \mathbb{R}^{3(N+1)}$, $u \in \mathbb{R}^{2(N+1)}$, and $d \in \mathbb{R}^{N+1}$ with the A, B, and V matrices of the appropriate sizes written as

$$A = \left[\begin{array}{cccc|c} A_{11} & 0 & \cdots & 0 & A_{10} \\ 0 & A_{22} & \ddots & \vdots & A_{20} \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & A_{NN} & A_{N0} \\ \hline A_{01} & A_{02} & \cdots & A_{0N} & A_{00} \end{array}\right], \quad B = \left[\begin{array}{cccc|c} B_{11} & 0 & \cdots & 0 & B_{10} \\ 0 & B_{22} & \ddots & \vdots & B_{20} \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & B_{NN} & B_{N0} \\ \hline B_{01} & B_{02} & \cdots & B_{0N} & B_{00} \end{array}\right], \tag{7.41}$$

$$V = \left[\begin{array}{cccc|c} V_{11} & 0 & \cdots & 0 & 0 \\ 0 & V_{22} & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & V_{NN} & 0 \\ \hline 0 & 0 & \cdots & 0 & V_{00} \end{array}\right]. \tag{7.42}$$

7.4 Model Validation

Up until this point, the gray-box modeling approach presented in the previous sections can be used to develop a linear model of any N evaporator of the form shown in Fig. 7.2. In order to validate this modeling approach, a 5-evaporator system, developed in ATTMO, is used as a representative system. Fig. 7.5 shows the 5-evaporator system modeled using ATTMO in the Simulink® environment. It is assumed that all 5 evaporators, EEVs, and evaporator fans are identical to one another. Note that future work will consider when this is not the case. Each of the heat exchangers are of the tube-and-fin configuration and the evaporators and EEVs are modeled after the components of a physical dual-evaporator system presented in [38], while the compressor and condenser are scaled appropriately. Fig. 7.5 does not show the rooms being cooled by each evaporator. The air temperature of the inlet air to each evaporator is varied according to the model of the room dynamics from (7.12). Each of the 5 rooms is identical.

To identify the parameters used throughout Tables 7.3 and 7.4, pseudo-random binary signals (PRBS) were sent to each of the actuators of the entire system. Fig. 7.6 shows the input signals for the compressor and condenser fan and Fig. 7.7 shows the inputs signals for EEV 1 and evaporator 1. Similar signals were also sent to the other 4 EEVs and evaporators. The step inputs of the PRBSs theoretically contain infinitely many frequencies and are used to adequately excite the system in order to accurately identify the various system parameters. A least-squares based approach was used to identify each of the parameters using the input and corresponding output signals. The identified parameters are shown in Table 7.5. Note that standard SI units are used: pressure (kPa), mass flow rate ($kg \cdot s^{-1}$), temperature (^oC), and heat transfer rate (kW).

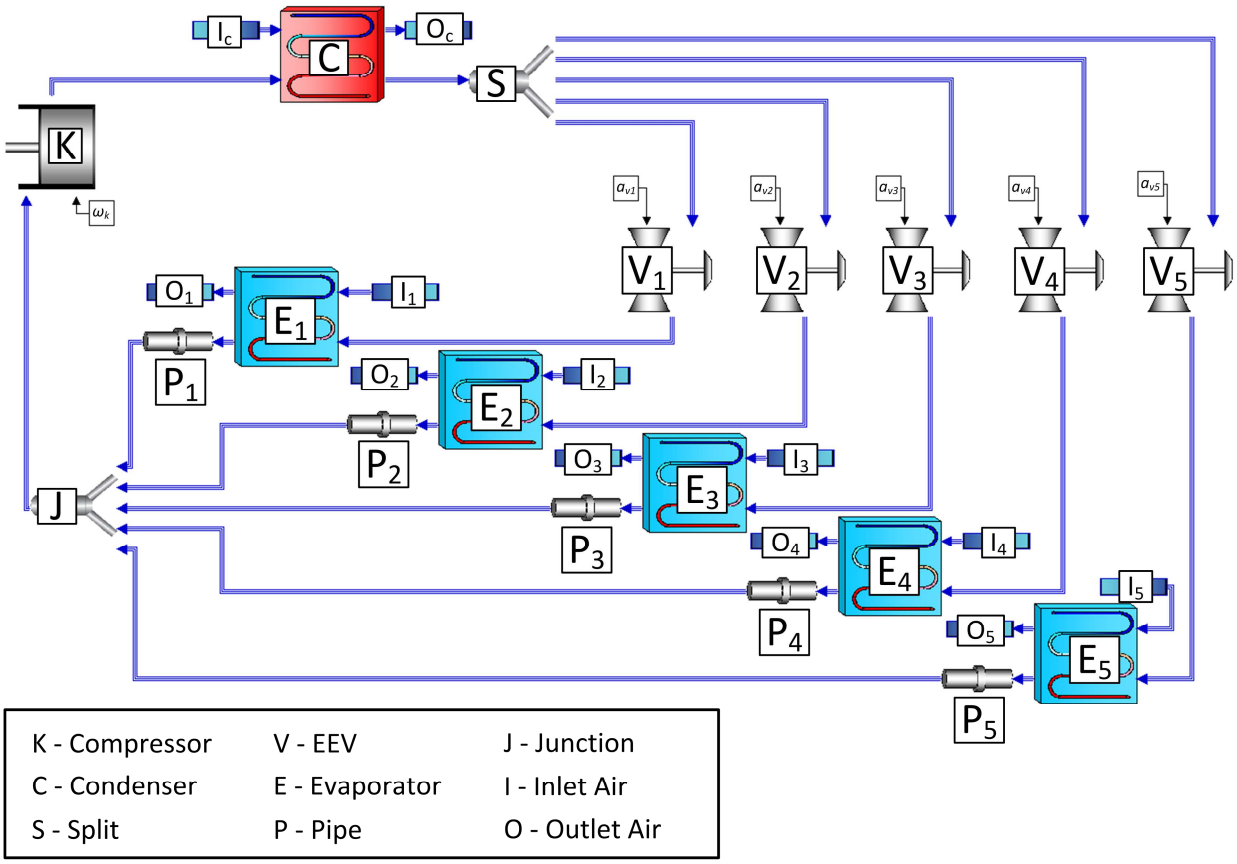

Figure 7.5 ATTMO model of 5-evaporator VRF system.

Table 7.5 Identified Fluid and Thermal Parameters

Component	Fluid		Thermal	
Compressor	β_{k1}	1.039×10^{-4}	α_{k1}	1.990×10^{-2}
	β_{k2}	2.422×10^{-7}	α_{k2}	1.510×10^{-4}
	β_{k3}	1.277×10^{-5}	α_{k3}	3.656×10^{-4}
	β_{k4}	1.163×10^{-4}	γ_{k1}	8.140×10^{-2}
			γ_{k2}	6.600×10^{-3}
			γ_{k3}	3.800×10^{-3}
			γ_{k4}	2.600×10^{-2}
			γ_{k5}	1.500×10^{-3}
EEV	R_{vi}	5.098×10^{5}	R'_{vi}	5.401×10^{1}
	K_{vi}	6.593×10^{1}	K'_{vi}	5.487×10^{-1}
			K'_{ki}	3.700×10^{-3}

Condenser	C_c	9.932×10^{-4}	C_{wc}	5.570×10^{0}
	κ_c	7.400×10^{-3}	R_{ac}	4.404×10^{0}
			R_{rc}	1.501×10^{0}
			η_{rc}	2.530×10^{-2}
			K_{fc}	1.890×10^{-2}
			λ_{rc1}	2.667×10^{2}
			λ_{rc2}	1.300×10^{-3}
Evaporator	C_i	7.888×10^{-5}	C_{wi}	1.211×10^{0}
	κ_i	7.991×10^{-4}	R_{ai}	9.250×10^{0}
			R_{ri}	1.276×10^{1}
			η_{r1i}	3.980×10^{-2}
			η_{r2i}	3.900×10^{-3}
			η_{r3i}	5.593×10^{2}
			K_{fi}	4.700×10^{-3}
			λ_{ri}	1.602×10^{3}
Pipe	R_{qi}	5.930×10^{3}	μ_{m1i}	4.151×10^{1}
	K_{q1i}	8.900×10^{-3}	μ_{m2i}	1.195×10^{4}
	K_{q2i}	3.400×10^{-3}	μ_{m3i}	3.762×10^{1}
	K_{q3i}	6.428×10^{-1}	μ_{m4i}	1.050×10^{4}
			μ_{m5i}	2.400×10^{-3}
Junction	C_q	4.720×10^{-4}	C_{rq}	6.510×10^{-2}
			κ_{rq}	2.691×10^{2}
Room			C_{ai}	6.500×10^{1}

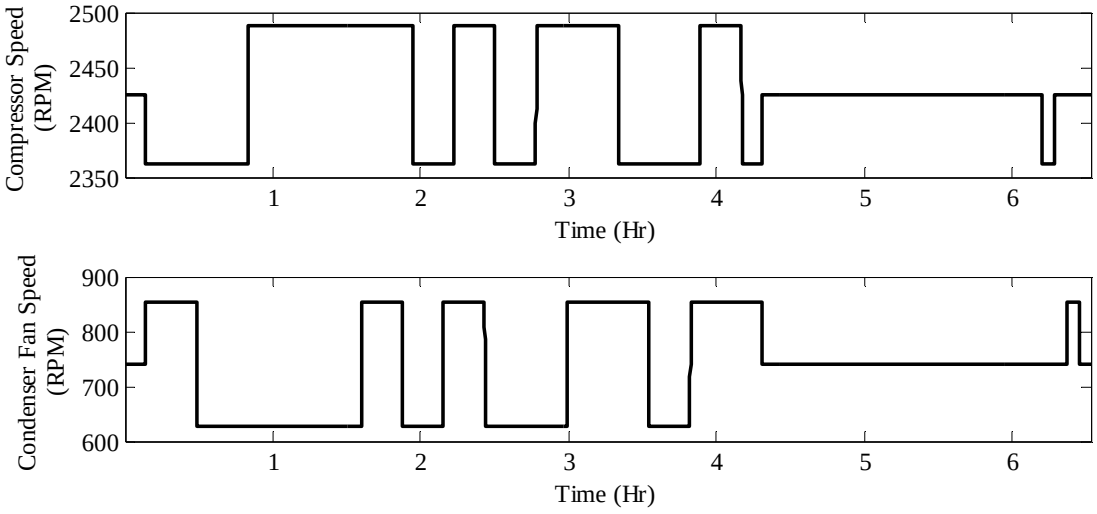

Figure 7.6 Compressor and condenser fan input signal for parameter identification.

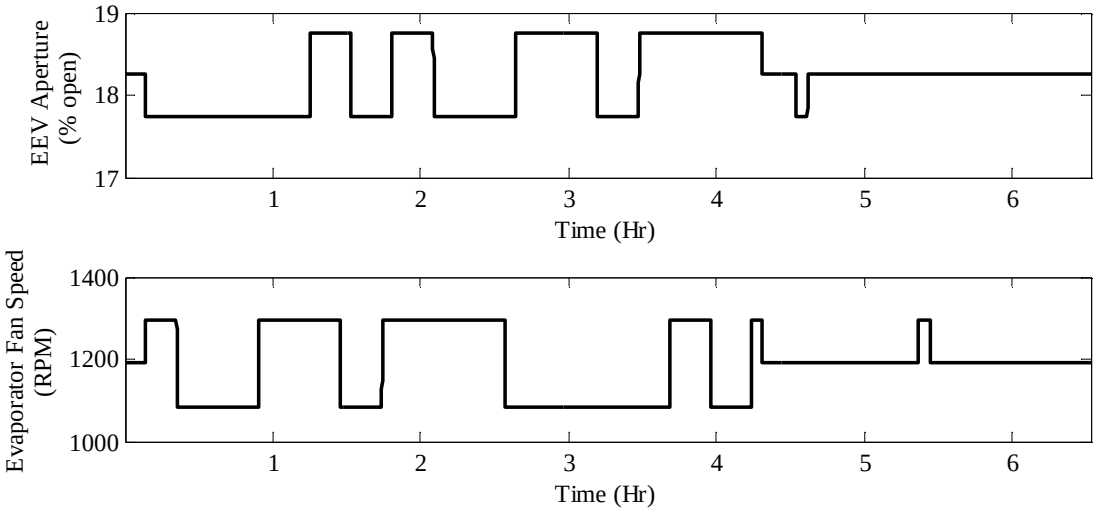

Figure 7.7 EEV and evaporator fan input signal for parameter identification.

Using the parameters from Table 7.5, the matrix elements from Tables 7.3 and 7.4 are evaluated and the complete system is modeled using A and B matrices from (7.41) and (7.42), with $N = 5$ and

$$A_{ii} = \begin{bmatrix} -2.186\times10^{0} & -1.890\times10^{0} & 1.095\times10^{0} \\ 2.848\times10^{-3} & -1.540\times10^{-1} & 8.925\times10^{-2} \\ 0 & 1.663\times10^{-3} & -1.663\times10^{-3} \end{bmatrix}, \quad A_{i0} = \begin{bmatrix} 2.301\times10^{-2} & 2.138\times10^{0} & 0 \\ -2.743\times10^{-4} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad (7.43)$$

$$B_{ii} = \begin{bmatrix} 1.517\times10^{0} & 1.009\times10^{-2} \\ -1.809\times10^{-2} & 6.762\times10^{-4} \\ 0 & -7.844\times10^{-6} \end{bmatrix}, \quad B_{i0} = \begin{bmatrix} 1.896\times10^{-2} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{7.44}$$

$$V_{ii} = \begin{bmatrix} 0 \\ 0 \\ 1.539\times10^{-2} \end{bmatrix}, \tag{7.45}$$

$$A_{00} = \begin{bmatrix} -5.140\times10^{-2} & 1.664\times10^{-1} & -2.715\times10^{-1} \\ -7.466\times10^{-4} & -2.007\times10^{0} & 2.464\times10^{-1} \\ 3.832\times10^{-3} & -4.206\times10^{-1} & -4.202\times10^{-1} \end{bmatrix}, \quad A_{0i} = \begin{bmatrix} -1.076\times10^{-2} & 1.563\times10^{-1} & 0 \\ 3.667\times10^{-1} & 0 & 0 \\ 6.498\times10^{-2} & 0 & 0 \end{bmatrix}, \tag{7.46}$$

$$B_{00} = \begin{bmatrix} 2.978\times10^{-2} & -3.204\times10^{-2} \\ -4.289\times10^{-2} & 0 \\ -1.321\times10^{-2} & 0 \end{bmatrix}, \quad B_{0i} = \begin{bmatrix} -1.302\times10^{-1} & 0 \\ -1.661\times10^{-2} & -3.003\times10^{-4} \\ -2.901\times10^{-2} & 5.380\times10^{-4} \end{bmatrix}, \tag{7.47}$$

$$V_{00} = \begin{bmatrix} 1.697\times10^{0} \\ 0 \\ 0 \end{bmatrix}. \tag{7.48}$$

Note that since all of the evaporators and EEVs are identical, the matrices for the i^{th} subsystem are applied to all 5 subsystems.

By applying the same input signals used to identify the parameters in Table 7.5, the outputs of the identified linear model are compared to the outputs from the ATTMO model in order to validate the linear model. Fig. 7.8 shows the validation results for the pressure P_1 wall temperature T_{w1}, and room air temperature T_{a1} for evaporator 1. Both the steady-state and transient responses for the evaporator pressure are accurately captured by the linear model. While the transient responses for the evaporator wall temperature are very accurate, there is a slight steady-state offset which results in a slight disagreement between the room air temperatures. Fig. 7.9 shows the validation results for the condenser pressure P_c, junction pressure P_q, and junction temperature T_{rq}. Once again, the transient responses for these outputs are accurately captured by the linear model, while there is some slight disagreement in the steady-state values. As will be shown in Section 7.6, these steady-state discrepancies can be easily overcome when controlling the system using integral action. In fact, the simulation results

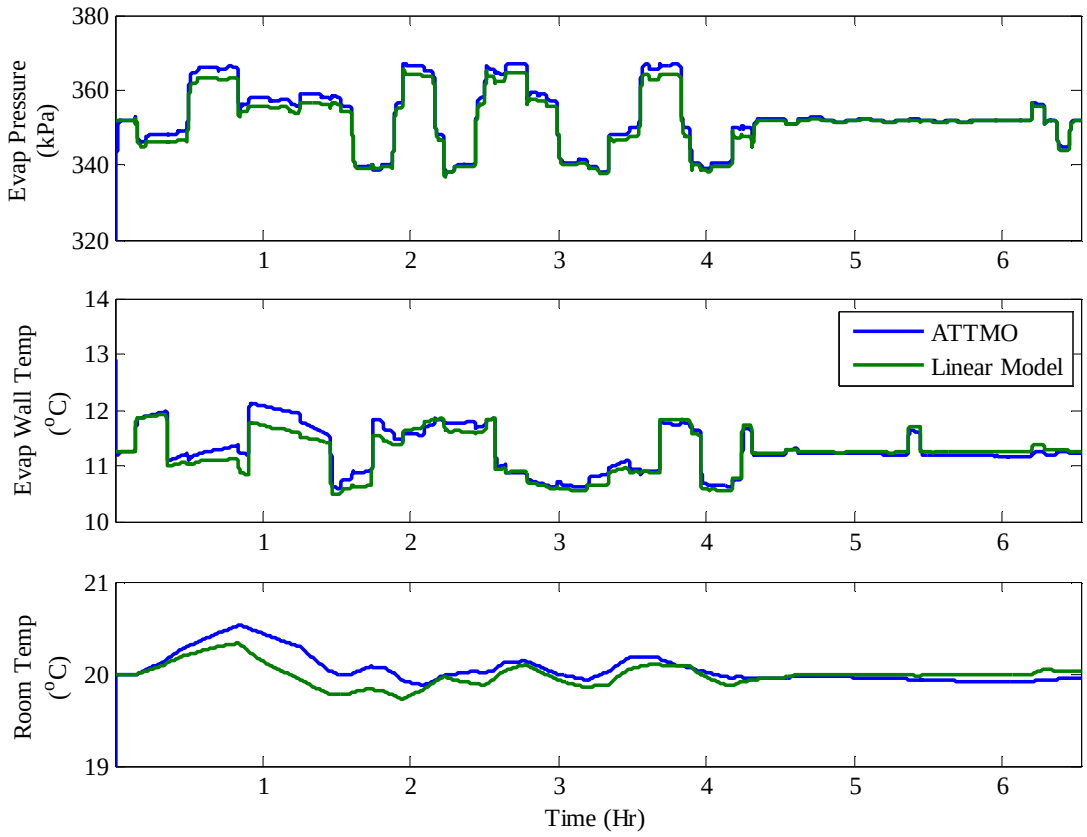

Figure 7.8 Validation of linear model for pressure, wall temperature, and room air temperature for evaporator 1.

in Section 7.6 show that the linear model using the parameters from Table 7.5 can be used to control the system well outside of the range of operating conditions for which the parameters were identified. For example, during the identification process the EEV apertures were only varied by ±0.5% open. However, from the scenarios in Section 7.6, the EEV apertures change by over 10% open from the nominal condition. Clearly the identified linear model can be used successfully outside of the range for which the parameters were identified. Unfortunately, additional simulation studies have shown that a single linear model cannot be used for all operating conditions. Examples of such conditions are very low heat exchanger fan speed and very low evaporator superheats (which is defined in Section 7.5.2), for which the system becomes very sensitive. If a model identified for operating conditions where the system is less sensitive is used under these more sensitive conditions, the system may oscillate significantly and even go unstable. A systematic analysis of the range in which a linear model can be used is

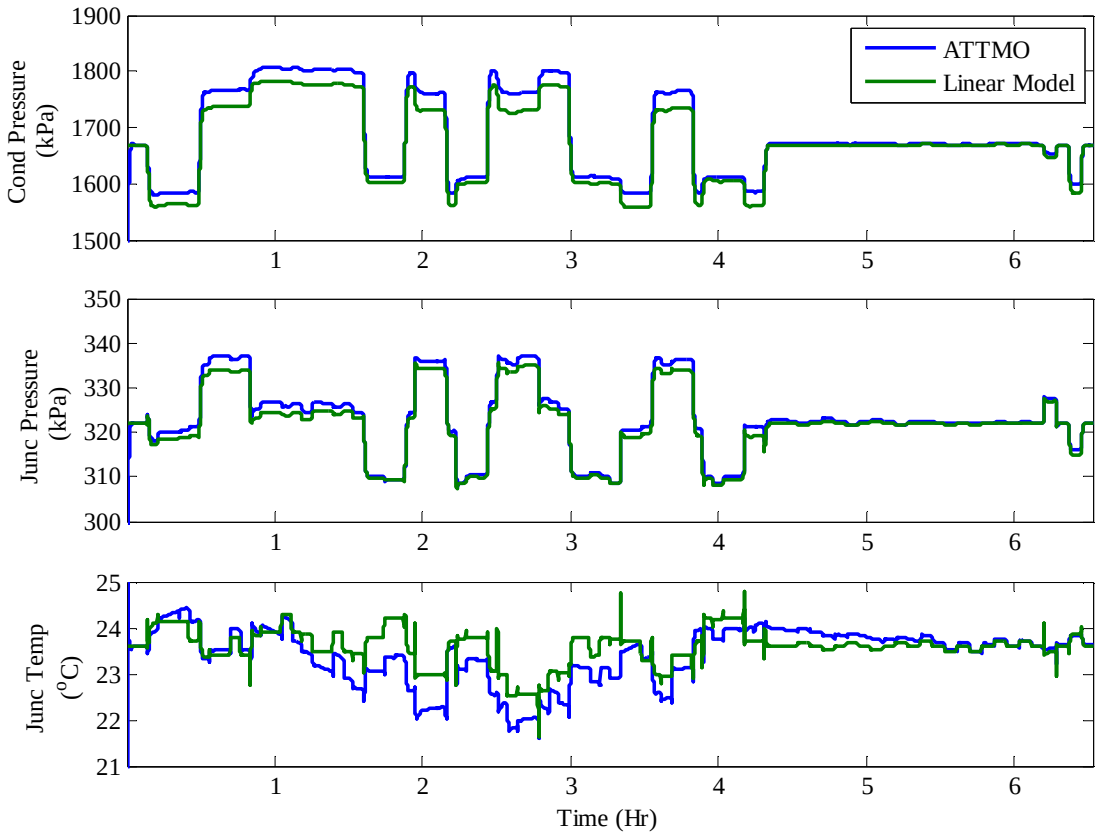

Figure 7.9 Validation of linear model for condenser pressure, junction pressure, and junction temperature.

left for future work. Additionally, while these figures only show model validation results about a single nominal operating condition, it is found that this gray-box modeling approach is able to accurately identify linear models for a wide range of operating conditions. Therefore, this same approach can be used to identify multiple linear models for different operating conditions and a gain scheduling approach, such as the one developed in [38], can be used to control the system over a very large range of operating conditions. It is expected that this same approach can also be used for a wide variety of systems with different components and configuration, however, validation of this claim is left for future work.

7.5 Control Design and Analysis

7.5.1 Controllability

Prior to developing a controller based on the linear model of the 5-evaporator VRF system, the controllability of the system must be verified. Using the test presented in Chapter 2, the structural controllability of a generic N-evaporator system can be easily verified using the linear system representation developed in Section 7.3. From Chapter 2, in order for a system to be structurally controllable, the system must both be input reachable and satisfy

$$\tilde{\rho}\left(\begin{bmatrix}\tilde{A} & \tilde{B}\end{bmatrix}\right) = n_s, \tag{7.49}$$

where $n_s = 3N$ is the number of states of the system. As stated in Chapter 2, for this work it is assumed that the decoupled subsystems $\mathbf{S}_i$ and $\mathbf{S}_0$ are structurally controllable, which guarantees that the entire system $\mathbf{S}$ is structurally controllable. To check the structural controllability of $\mathbf{S}_i$, the binary matrices $\bar{A}_{ii}$ and $\bar{B}_{ii}$ are

$$\bar{A}_{ii} = \begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 0 & 1 & 1\end{bmatrix},\ \bar{B}_{ii} = \begin{bmatrix}1 & 1\\ 1 & 1\\ 0 & 1\end{bmatrix}. \tag{7.50}$$

Since $\bar{B}_{ii}$ does not have any zero rows, the G matrix of the reachability matrix R will not have any zero rows and, therefore, the $\mathbf{S}_i$ subsystem is input reachable. Additionally, $\tilde{\rho}\left(\begin{bmatrix}\tilde{A}_{ii} & \tilde{B}_{ii}\end{bmatrix}\right) = 3$ and, therefore, $\mathbf{S}_i$ has full generic rank. Thus, the subsystem $\mathbf{S}_i$ is structurally controllable. The same is shown for the $\mathbf{S}_0$ subsystem, where the binary matrices $\bar{A}_{00}$ and $\bar{B}_{00}$ are

$$\bar{A}_{ii} = \begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix},\ \bar{B}_{ii} = \begin{bmatrix}1 & 1\\ 1 & 0\\ 1 & 0\end{bmatrix}. \tag{7.51}$$

Once again, $\bar{B}_{00}$ does not have any zero rows, and thus, $\mathbf{S}_0$ is input reachable. The subsystem $\mathbf{S}_0$ also has full generic rank with $\tilde{\rho}\left(\begin{bmatrix}\tilde{A}_{00} & \tilde{B}_{00}\end{bmatrix}\right)=3$. Therefore, the subsystem $\mathbf{S}_0$ is also structurally controllable. As mentioned in Section 2.3, structural controllability does not always mean that the system is controllable. In fact, if the state T_{wc} had not been removed from the $\mathbf{S}_0$ subsystem, the N-evaporator system would not be controllable despite being structurally controllable. As previously mentioned, this is due to a relationship between the parameters in the A and B matrices based on an algebraic relationship between several states of the system shown in (7.31). Therefore, when using gray-box modeling techniques, it is very important to look for these types of algebraic constraints and use them to reduce the number of states in the system prior to developing a control strategy for the system.

7.5.2 Control Objectives

As with the example system, the VRF system has several performance and efficiency control objectives. The primary purpose of a VRF system is to provide a cooling capacity $\dot{Q}_{awi}$ to offset the thermal load $\dot{Q}_{Li}$ in each room or zone being cooled by the system. In doing so, the VRF system can regulate the air temperature in each room to a desired value. Thus, the local performance objective $J_{p,i}$ for each $\mathbf{S}_i$ subsystem is to regulate T_{ai} to a desired value r_{ai}, which is expressed as

$$J_{p,i}=\left(T_{ai}-r_{ai}\right)^2. \tag{7.52}$$

In practice, a popular alternative to this performance objective may be to constrain the room air temperature between upper and lower bounds, as is done in [39]. This provides additional flexibility in the control of the system and will be explored in future work.

An additional performance objective is typically placed on the VRF system pertaining to the superheat of the system. In a single-evaporator vapor compression system, superheat, denoted here as T_{SH}, is the difference between the temperature of the refrigerant exiting the evaporator $T_{e,out}$ and the evaporation (or saturation) temperature of the refrigerant in the

evaporator $T_{e,\text{sat}}$, which is based on the evaporator pressure P_e. Thus, superheat for a single evaporator system is defined as

$$T_{SH} = T_{e,out} - T_{e,sat}\Big|_P . \tag{7.53}$$

A positive superheat means that the refrigerant exiting the evaporator is completely vaporized and some of the heat absorbed by the evaporator has been used to raise the temperature of this vapor above the evaporation temperature. Zero superheat means that the temperature of the refrigerant exiting the evaporator is the same as the evaporation temperature. With zero superheat, there is the potential that all of the liquid which entered the evaporator has not turned to vapor and, thus, some liquid may exit the evaporator. This liquid can cause damage to the compressor and, therefore, it is desirable to run the system with a non-zero value of superheat. However, running the system with a very high superheat is very inefficient. Therefore, an additional control objective for the system is to regulate evaporator superheat to a desired value. Alternatively, model predictive control provides the flexibility of simply constraining the values of superheat where a lower bound is used to prevent liquid from entering the compressor and an upper bound is used to prevent the system from running inefficiently. This approach has been used in a number of research efforts including VRF systems [36].

While controlling evaporator superheat is very effective for single evaporator systems, in [7] it is found that directly controlling the superheat of each evaporator for a multi-evaporator system may be challenging, especially if a decentralized control approach is used. Additionally, there may not even be a need to directly control the superheat of each evaporator. The true purpose of controlling superheat is to prevent damage to the compressor. Therefore, the state of the refrigerant entering the compressor is of interest, not the state of the refrigerant exiting the evaporator. For single evaporator systems the state of the refrigerant is typically very similar at these two locations, but for a VRF system this may not be the case due to the mixing of refrigerant in the junction downstream of the evaporators. The superheat of the refrigerant exiting the junction $T_{SH,q}$ defined as

$$T_{SH,q} = T_{rq} - T_{rq,sat}\Big|_{P_q} \tag{7.54}$$

where T_{rq} is the junction temperature and $T_{rq,sat}$ is the junction saturation temperature, which is based on the junction pressure P_q. By controlling $T_{SH,q}$, the evaporators have additional freedom. Now it is possible for an evaporator to lose superheat but due to the superheat of the other evaporators, the junction superheat can remain non-zero. This additional freedom helps prevent the fighting behavior seen in [7]. Thus, in addition to the air temperature regulation performance objectives, the control of junction superheat is a performance objective for the $\mathbf{S}_0$ subsystem and is written as

$$J_{p,0} = \left(T_{SH,q} - r_{SH}\right)^2. \tag{7.55}$$

Similar to the air temperature objective, this superheat objective could be replaced by upper and lower constraints as is done in [7] and [36] to provide greater flexibly in the operation of the system.

Unfortunately, $T_{SH,q}$ is not a state of the linear system $\mathbf{S}$ (7.40) identified in Section 7.3. However, as seen in (7.54), $T_{SH,q}$ is a function of T_{rq} and P_q which are both states of $\mathbf{S}_0$. Therefore, $T_{SH,q}$ can be written as a function of states, if the relationship between $T_{rq,sat}$ and P_q is identified. It is well known, that the saturation temperature for a refrigerant is a nonlinear function of pressure. Fig. 7.10 shows this relationship for R-134a over a wide range of pressures. Fig. 7.10 also shows that a linear approximation can be used to calculate the saturation temperature based on pressure. This linear approximation matches the true saturation temperature value within $\pm 2^{o}$C between 186.7 and 452.3 kPa. This range easily covers the range of expected operating conditions for the low pressure side of most R-134a systems and, therefore, the linear approximation can be used. Thus, $T_{SH,q}$ is written as a function of the states T_{rq} and P_q as

$$T_{SH,\mathrm{q}} = T_{rq} - \alpha P_q, \tag{7.56}$$

where $\alpha = 0.092$. Note that since all of the states are actually deviations from a nominal operating condition, the constant term from the linear model can be neglected.

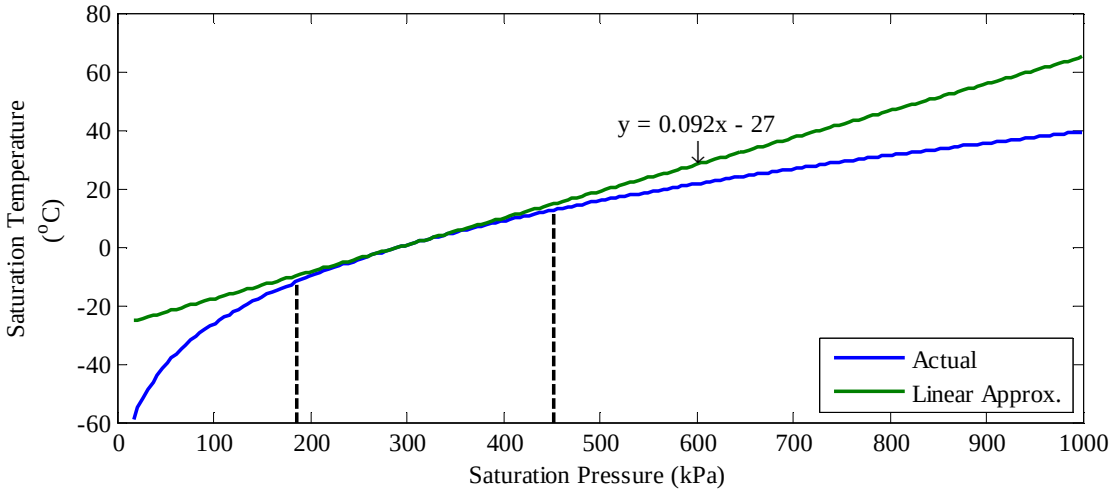

Figure 7.10 Linear approximation of the relationship between saturation pressure and saturation temperature for R-134a.

In order to have $T_{SH,q}$ be a state of the linear system, a transformation similar to the one done in (3.20) can be used, where

$$C = \left[\begin{array}{c|ccc} I_{15\times 15} & & 0_{15\times 3} & \\ \hline & 0 & 0 & 0 \\ 0_{3\times 15} & 0 & 0 & 0 \\ & 0 & -\alpha & 1 \end{array}\right]. \tag{7.57}$$

The state vector for the $\mathbf{S}_i$ subsystems remains $x_i = \begin{bmatrix} P_i & T_{wi} & T_{ai} \end{bmatrix}^T$, but now the state vector for the $\mathbf{S}_0$ subsystem is $x_0 = \begin{bmatrix} P_c & P_q & T_{SH,q} \end{bmatrix}^T$.

In addition to the performance objective, there is a global efficiency objective. The compressor, condenser fan, and the N evaporator fans all consume a significant amount of power. Therefore, the efficiency objective for the system is to minimize the power consumed by the entire system. Fig. 7.11 shows the relationship between fan speed and power consumption for the condenser and evaporator fans. This data is based on the fans used for the dual-evaporator experimental system in [38] and was scaled appropriately for the 5-evaporator system. A quadratic function is used to approximate the relationship between fan speed and

power consumption. The resulting power consumption for the condenser and evaporator fans is approximated as

$$\kappa_c = a_c \omega_{fc}^{\ 2} + b_c \omega_{fc} + c_c, \tag{7.58}$$

$$\kappa_i = a_i \omega_{fi}^{\ 2} + b_i \omega_{fi} + c_i. \tag{7.59}$$

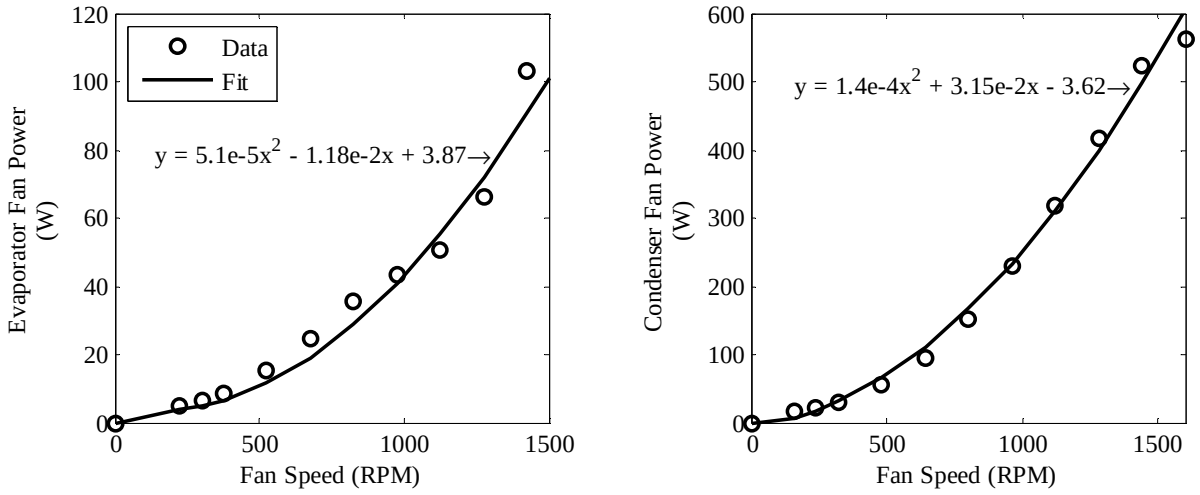

Figure 7.11 Power consumption for heat exchanger fans.

The power consumption for the compressor is not as straightforward. Fig. 7.12 shows the power consumed by the compressor for the experimental system in [38] over a range of compressor speeds and EEV apertures, shown in Fig. 7.13. The figure also shows two approximations based on the quadratic functions

$$\kappa_k = a_k \omega_k^{\ 2} + b_k \omega_k + c_k, \tag{7.60}$$

$$\kappa_k = a_k \omega_k^{\ 2} + b_k \omega_k + a_{Pc} P_c^2 + b_{Pc} P_c + a_{Pq} P_q^2 + b_{Pq} P_q + c_k. \tag{7.61}$$

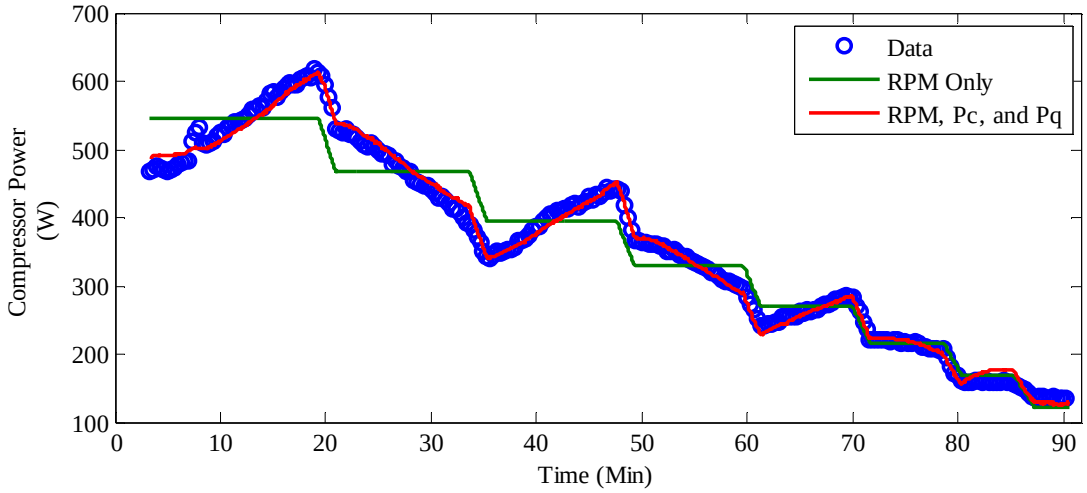

Figure 7.12 Power consumption for compressor.

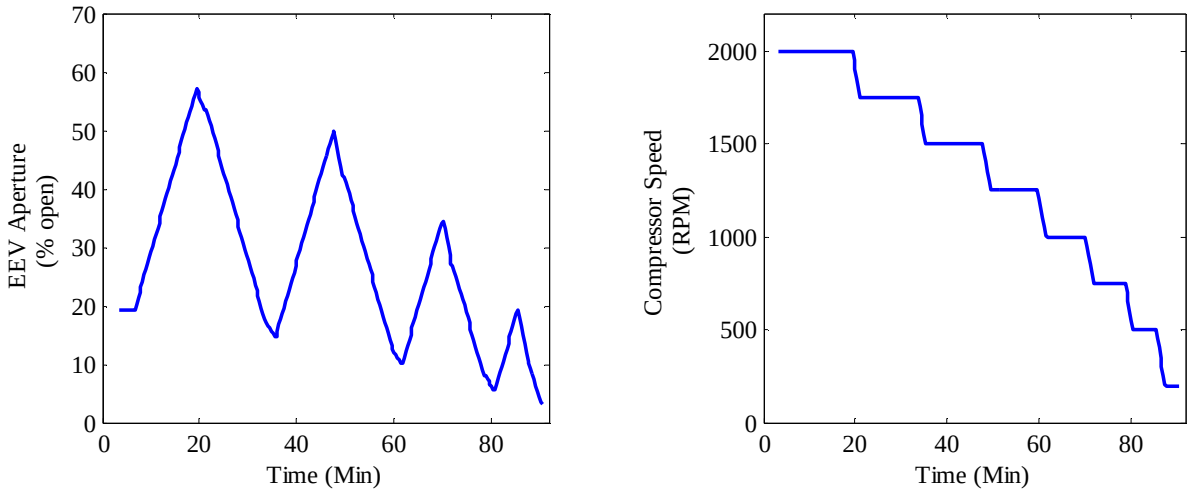

Figure 7.13 Compressor speed and EEV inputs used to develop compressor power map.

From Fig. 7.12, it is clear that unlike the heat exchanger fans, the compressor power is not just a function of the compressor speed. The power is also significantly affected by the condenser and junction pressures. Fortunately, these pressures are states in the identified linear model. With the addition of the dependence of state variables, the actuator cost function from (3.32) can be written as

$$K = U^T Q_a U + q_b^T U + X^T Q_{a,x} X + q_{b,x}^T X, \tag{7.62}$$

where $Q_{a,x}$ and $q_{b,x}$ contain the a_{Pc}, b_{Pc}, a_{Pq}, and b_{Pq} terms from (7.61). This slightly modifies the MPC cost function

$$J = \Delta U^T H \Delta U + F^T \Delta U, \tag{7.63}$$

where now

$$\begin{aligned} H &= \bar{S}^T \bar{P}^T Q_1 \bar{P} \bar{S} + Q_2 + \bar{N}^T Q_a \bar{N} + \bar{S}^T Q_{a,x} \bar{S}, \\ F &= 2\bar{S}^T \bar{P}^T Q_1 \bar{P} \bar{T} \bar{x}(k) + 2\bar{S}^T \bar{P}^T Q_1 \bar{P} \bar{R} d + 2\bar{N}^T Q_a \bar{n} u(k) + \bar{N}^T q_b \\ &\quad + 2\bar{S}^T Q_{a,x} \bar{T} \bar{x}(k) + 2\bar{S}^T Q_{a,x} \bar{R} d + \bar{S}^T q_{b,x}. \end{aligned} \tag{7.64}$$

7.5.3 MPC Formulation

The centralized and BAS control architectures developed in Chapter 5 are used to control the 5-evaporators VRF system. The decentralized control architecture is not used for reasons explained in the following section. The same procedure demonstrated with the example system is used to develop these control architecture and individual MPC controllers with a few slight differences. For the centralized controller, we now have $x \in \mathbb{R}^{18}$, $u \in \mathbb{R}^{12}$, $d \in \mathbb{R}^6$, $y \in \mathbb{R}^{18}$, and C is from (7.57). With 6 reference values, $r \in \mathbb{R}^6$, the matrix M used to select which outputs have desired reference values is,

$$M = \text{blkdiag}\left(\begin{bmatrix}0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 0 & 1\end{bmatrix}\right). \tag{7.65}$$

Now, the exact same procedure is used to augment the system into error states e, integral states z, and changes in the control input Δu. The system is then written in lifted form, and the cost function is written in terms of the lifted output and input vectors. The fact that the compressor power consumption depends on the states has already been addressed in the previous section, resulting in the cost function (7.63) with augmented terms in (7.64). All of the actuators in the VRF system have upper and lower bounds which are presented for the 5-evaporator system in Table 7.6. These actuator constraints are written in a lifted form and converted to be a function of ΔU and the actuator input $u(k)$ at sample time k.

The same procedure is used to develop the BAS controllers, which are based on the BAS subsystem representation

Table 7.6 Actuator Constraints for VRF System

Actuator	Lower Bound	Upper Bound
Compressor Speed (RPM)	0	3500
Condenser Fan Speed (RPM)	0	1600
EEV Aperture (% open)	0	100
Evaporator Fan Speed (RPM)	0	1500

$$\mathbf{S}_i' : \begin{bmatrix} \dot{x}_i \\ \dot{\tilde{x}}_0 \end{bmatrix} = \begin{bmatrix} A_{ii} & A_{i0} \\ 5A_{0i} & A_{00} \end{bmatrix} \begin{bmatrix} x_i \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} B_{ii} & B_{i0} \\ 5B_{0i} & B_{00} \end{bmatrix} \begin{bmatrix} u_i \\ \tilde{u}_0 \end{bmatrix} + \begin{bmatrix} V_{ii} & 0 \\ 0 & V_{00} \end{bmatrix} \begin{bmatrix} d_i \\ d_0 \end{bmatrix}. \tag{7.66}$$

Note that the A_{0i} and B_{0i} terms are multiplied by the number of evaporators ($N = 5$). Since each $\mathbf{S}_i$ subsystem is identical, multiplying A_{0i} and B_{0i} by N can significantly improve the accuracy of the BAS models. In fact, if all the states x_i and input u_i are the same, then this BAS model has the same dynamics as the centralized model. For the BAS controllers, we have $x \in \mathbb{R}^6$, $u \in \mathbb{R}^4$, $d \in \mathbb{R}^2$, $y \in \mathbb{R}^6$, and C is

$$C_{ii}' = \left[\begin{array}{c|ccc} I_{3\times3} & \multicolumn{3}{c}{0_{3\times3}} \\ \hline & 0 & 0 & 0 \\ 0_{3\times3} & 0 & 0 & 0 \\ & 0 & -\alpha & 1 \end{array}\right]. \tag{7.67}$$

With 2 reference values, $r \in \mathbb{R}^2$, the matrix M_i' used to select which outputs have desired reference values is,

$$M_i' = \text{blkdiag}\left(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\right). \tag{7.68}$$

The remainder of the steps used for the centralized controller can be directly applied to develop the BAS controllers.

For the centralized and BAS MPC controllers, a sample time of $\Delta t = 10$ seconds is used with a control horizon of $N_u = 15$ steps and a prediction horizon of $N_p = 60$ steps. These parameters were chosen based on the dynamics of the VRF system. It is important to choose these parameters such that the MPC controller predicts far enough into the future to capture to dynamics of the slowest dynamic of the system. The combination of the prediction horizon and

sample time allows the controllers to look 600 seconds into the future, which is long enough to capture the dynamics of the room air temperatures. Additionally, the sample time needs to be small enough to capture the fastest dynamics of the system. It is found that 10 seconds was the largest sample time which still adequately captured the dynamics of the junction superheat. Finally, the control horizon is chosen as a balance of control performance and computational costs. The control horizon directly affects the size of the optimization vector U. The smaller the control horizon the lower the computation costs, however, the control performance can be significantly reduced by not considering enough future control decisions. A control horizon of 15 steps is found to be an appropriate compromise for the VRF system.

7.5.4 Decentralized Control

It is well known that some systems cannot be controlled in a decentralized manor due to the high degree of coupling between subsystems. Typically, if this high degree of coupling is ignored, the controlled actuators begin to "fight" with each other, causing the system to oscillate and even go unstable [40]. From the linear model (7.40) identified in Section 7.3, it is found that the high degree of coupling between the $\mathbf{S}_i$ and $\mathbf{S}_0$ subsystems prevents the decentralized control approach developed in Chapter 5 from being effective. The decentralized model for the $\mathbf{S}_0$ subsystem

$$\mathbf{S}_0 : \dot{x}_0 = A_{00}x_0 + B_{00}u_0 + V_{00}d_0, \tag{7.69}$$

does not accurately capture the relationship between the inputs u_0 (compressor speed and condenser fan speed) and the junction superheat $T_{SH,q}$. Fig. 7.14 shows the superheat response for step changes in compressor speed and condenser fan speed (Fig. 7.15) for the centralized, decentralized, and BAS linear system models along with the response from the ATTMO model. Note that the centralized and BAS models predict the same responses and are represented by the same trace in the figure. For the step decrease in compressor speed at 5 minutes, the superheat for the ATTMO model decreases and is accurately captured by the centralized and BAS models. However, the decentralized model predicts an increase in superheat. Similarly, for the increase in condenser fan speed at 15 minutes the superheat increases, however, the decentralized model predicts a decrease in superheat.

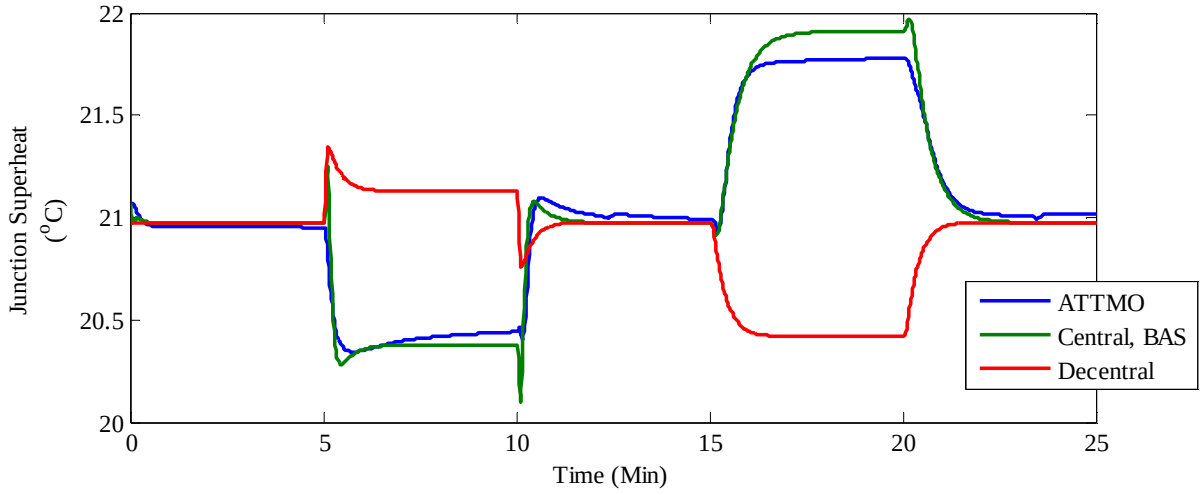

Figure 7.14 Junction superheat response for changes in compressor speed and condenser fan speed from ATTMO model and linear models.

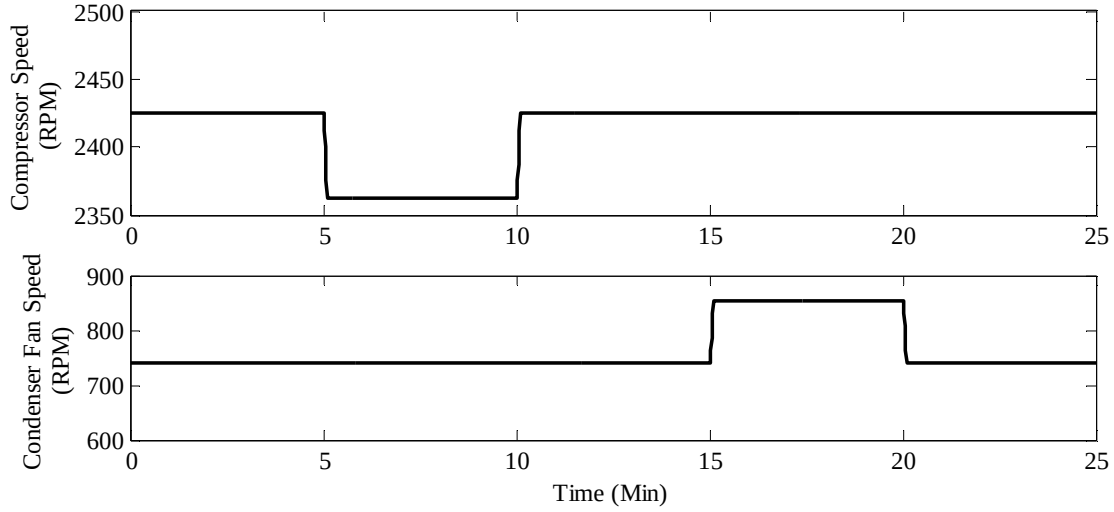

Figure 7.15 Changes in compressor speed and condenser fan speed for superheat analysis.

The fact that both of the relationships between compressor speed and superheat and condenser fan speed and superheat have the wrong sign means that integral action cannot simply be used to overcome the model mismatch. In fact, if integral action is used, the system goes unstable. While the current model cannot be used for decentralized control, this is not to say that decentralized control can never be used for VRF systems or even that a different operating condition would produce these same results. Other control approaches [7] have shown that decentralized approaches can be effective when using different modeling frameworks. However,

these modeling frameworks often use black-box model identification techniques and do not provide the benefits of a gray-box modeling approach identified in Section 7.2. Additionally, unlike the current approach, these alternative techniques may not be able to consider system-level efficiency objectives and do not easily scale to N evaporator systems.

7.6 Simulation Results

Three different scenarios are used to demonstrate the performance of the centralized and BAS control strategies. First, a baseline scenario is used to compare the centralized and BAS strategies under step changes in thermal load Q_{Li} and ambient temperature T_{ac}. While these disturbances may not be realistic, this scenario uses these simple disturbances to demonstrate the similarities and differences between the two control strategies. The second scenario shows how ESC can be used to modify the BAS control architecture in order to provide additional improvements in system efficiency. The final scenario tests the BAS control architecture under a more realistic set of disturbances.

7.6.1 Baseline Scenario

Fig. 7.16 shows the disturbances Q_{Li} and T_{ac} over the 80 minute simulation for the baseline scenario. Note that this 80 minute scenario took only 2.5 minutes in real-time to

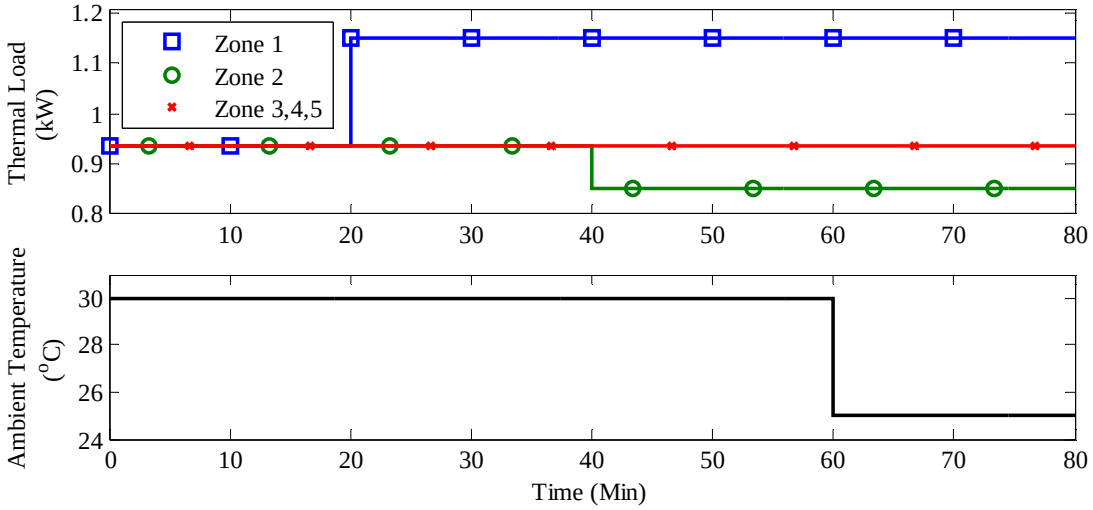

Figure 7.16 System disturbances Q_{Li} and T_{ac} for baseline scenario.

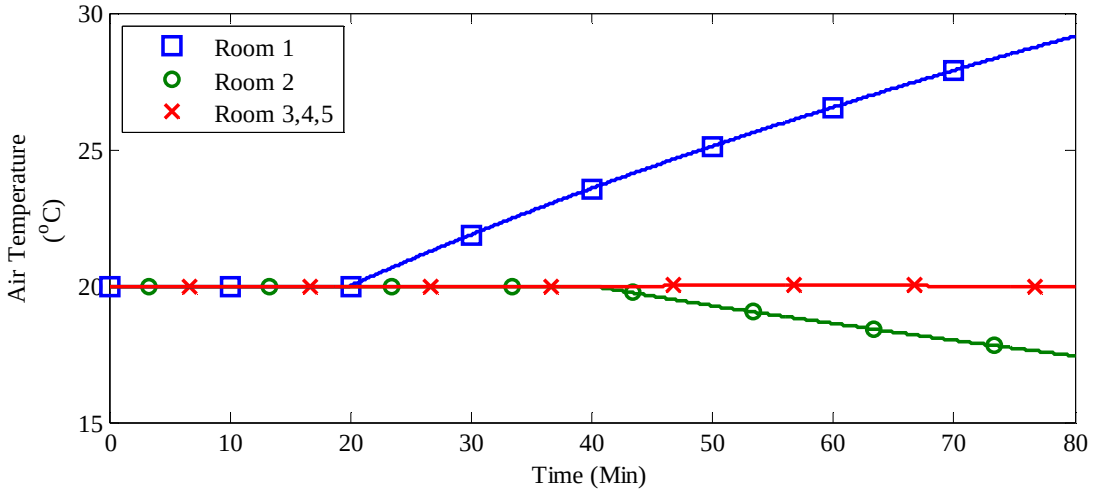

Figure 7.17 Open-loop response for T_{ai} due to disturbances for baseline scenario.

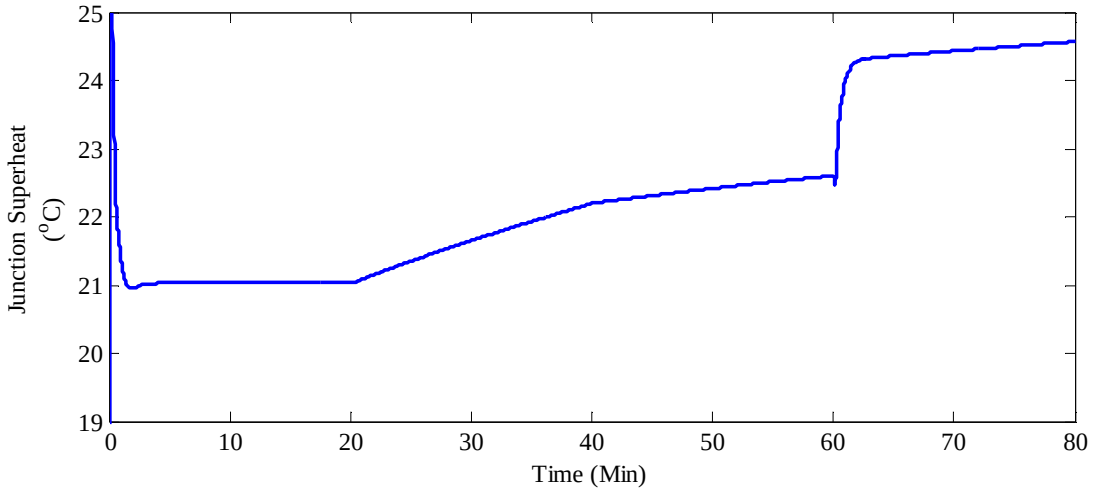

Figure 7.18 Open-loop response for $T_{SH,q}$ due to disturbances for baseline scenario.

simulate, including the computation time for the MPC controllers, thus highlighting one of the many benefits of the ATTMO modeling framework. These disturbances are roughly 10 – 20% deviations from the nominal conditions and were designed to sufficiently test the performance of the control architectures. These disturbances to the system are unknown to the MPC controllers since the thermal load in a room is typically unknown. Note that for simplicity subsystems $\mathbf{S}_3$, $\mathbf{S}_4$, and $\mathbf{S}_5$ have the same disturbances and, thus, the results for these subsystems will be presented together. For comparison purposes, Fig. 7.17 and 7.18 show the open-loop response of

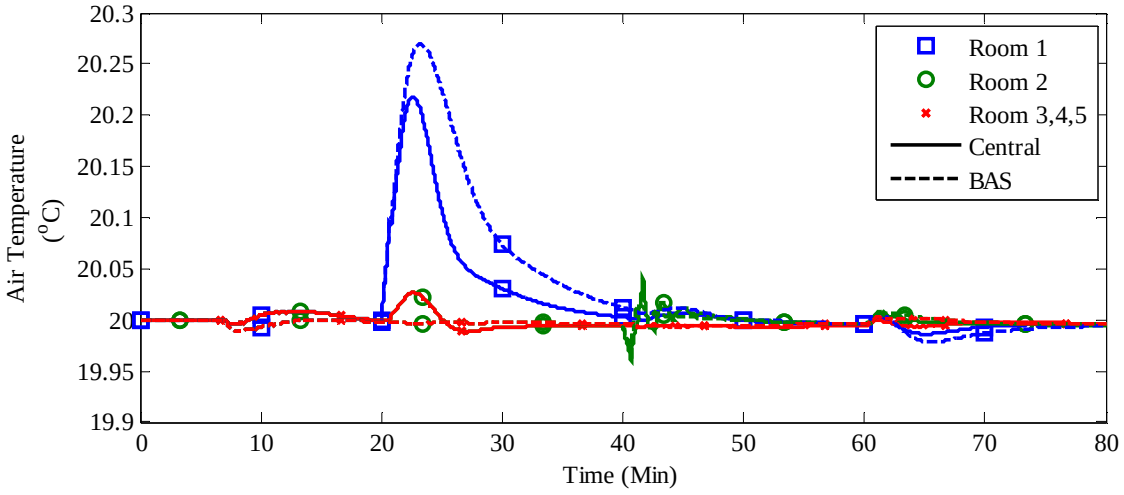

Figure 7.19 Ability of each control architecture to meet local performance objectives by tracking the desired value for T_{ai} for baseline scenario.

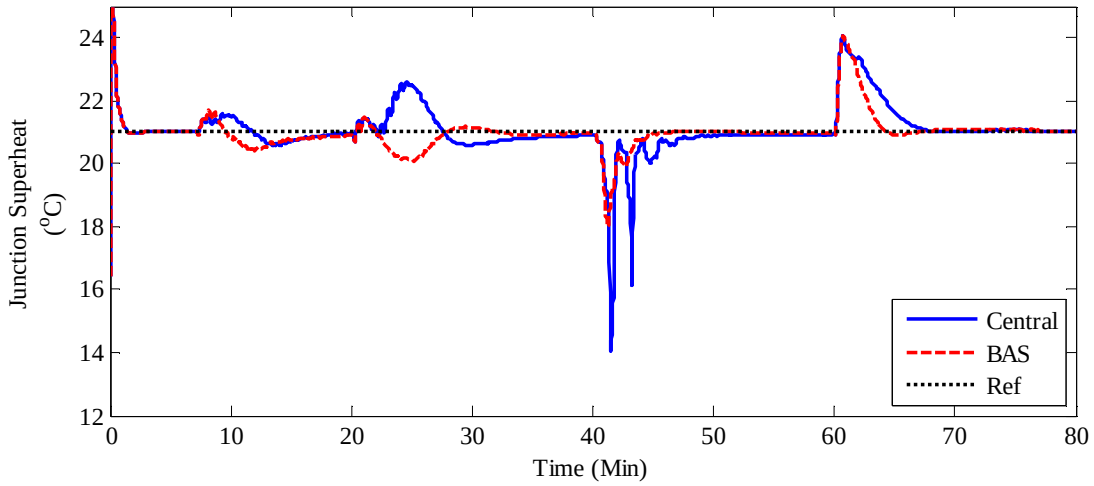

Figure 7.20 Ability of each control architecture to track the desire junction superheat for baseline scenario.

the system to the disturbances from Fig. 7.16. With all of the inputs held constant at the nominal input conditions, the room 1 air temperature deviates nearly 10°C from the nominal condition and the junction superheat deviates nearly 5°C.

Figs. 7.19 and 7.20 show the responses in air temperature T_{ai} and junction superheat $T_{SH,q}$, which represent the performance objectives for both the centralized and BAS control

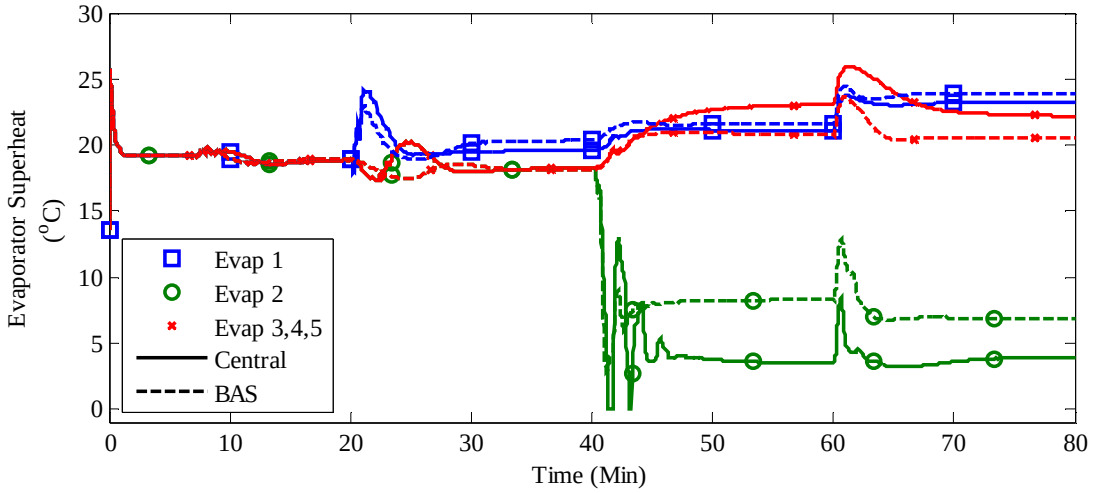

Figure 7.21 Different refrigerant superheat for each evaporator for baseline scenario.

architectures. Both architectures are able to track the desired room temperatures very accurately. The BAS approach performs slightly better than the centralized at tracking the desired superheat. This is because the centralized solution drives the superheat for evaporator 2 to a lower value and superheat is lost during the transient, causing large oscillations, as seen in Fig. 7.21. Fig. 7.21 also demonstrates the additional flexibility of the proposed superheat control strategy. The superheat in each evaporator can differ significantly depending on the operating conditions as long as the junction superheat maintains the desired value.

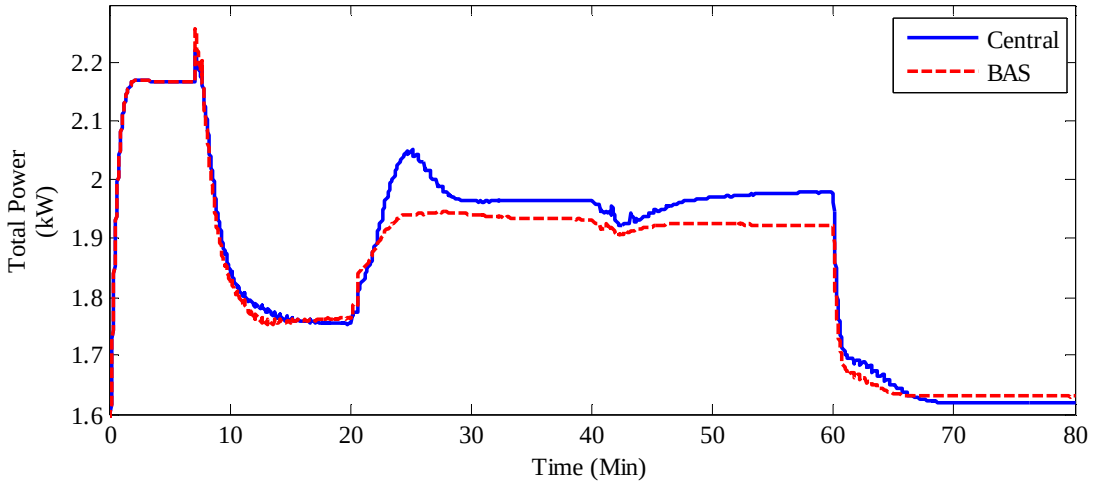

Figure 7.22 Ability of each control architecture to meet the global efficiency objective by minimizing the total power consumption by the actuators for baseline scenario.

As seen in Fig. 7.22, the centralized and BAS strategies result in very similar system efficiencies. In fact, the BAS strategy is able to meet the performance objectives using less power than the centralize approach for a large portion of the simulation. It is expected that this is due to the unknown disturbances as well as the nonlinearity in the system which is not captured by the linear models used for MPC. The decentralized nature of the BAS approach may be more robust to these disturbances and nonlinearity than the centralized approach. More importantly though is the fact that both control approaches provide significant improvement in the efficiency of the system. When the controllers are activated 4 minutes into the simulation, the total system power decreases 23% prior to the first disturbance at 17 minutes. Each control architecture is able to find a better combination of actuator inputs which reduces the total system power, while still meeting the performance objectives. Figs. 7.23 and 7.24 show that the two approaches result in very similar control decisions for the EEV aperture a_{vi} and evaporator fan speeds ω_{fi}. The majority of the difference between the two approaches comes from the control of the compressor speed ω_k and the condenser fan ω_{fc} as seen in Figs. 7.25 and 7.26.

Overall, this scenario shows that the BAS control architecture is a very effective approach for controlling VRF systems. The fact that all of the evaporators are identical allows the BAS model from (7.66) to very accurately capture the dynamics of the entire system allowing for performance comparable to the centralized approach while remaining scalable to larger systems with more evaporators.

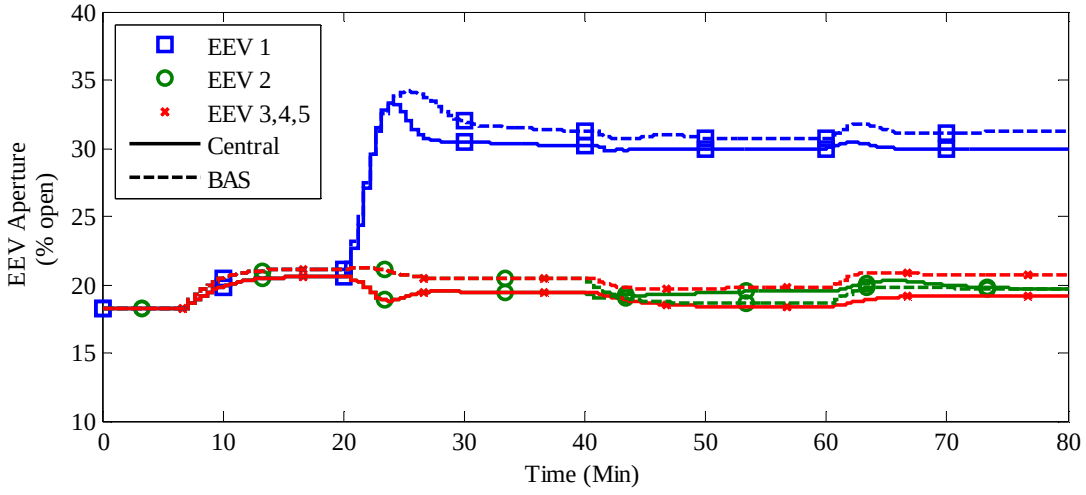

Figure 7.23 Actuator inputs a_{vi} from each control architecture for baseline scenario.

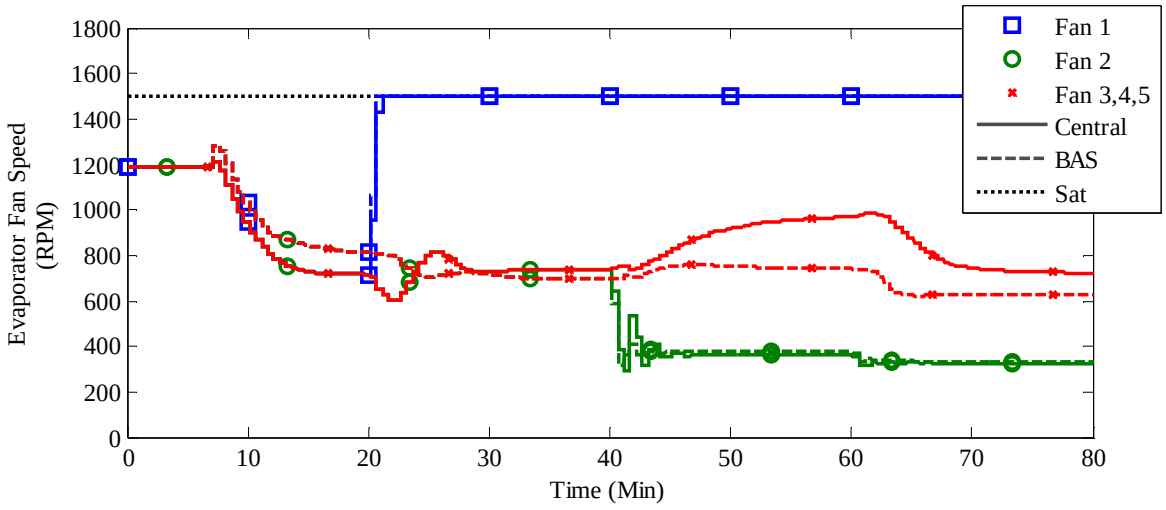

Figure 7.24 Actuator inputs ω_{fi} from each control architecture for baseline scenario.

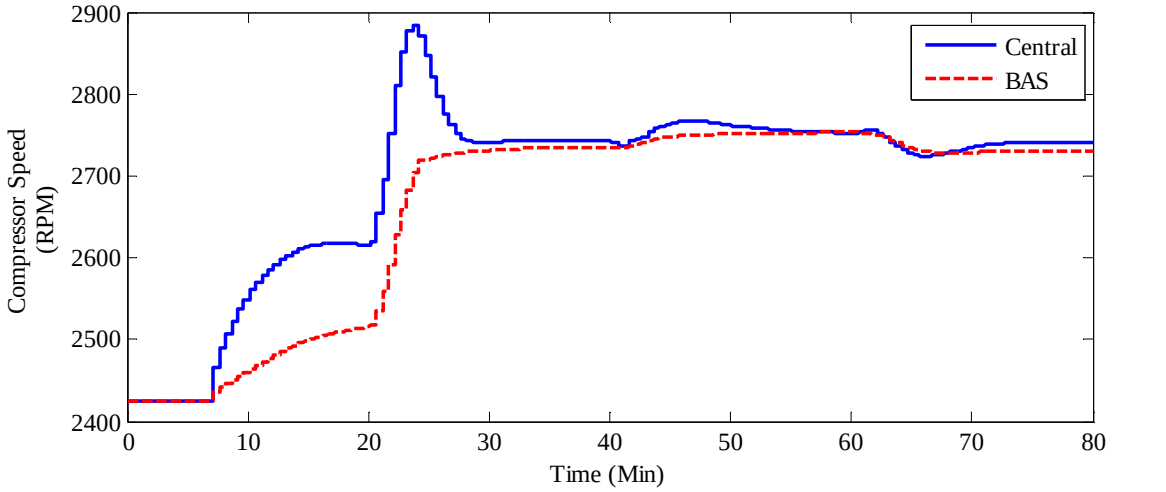

Figure 7.25 Actuator inputs ω_k from each control architecture for baseline scenario.

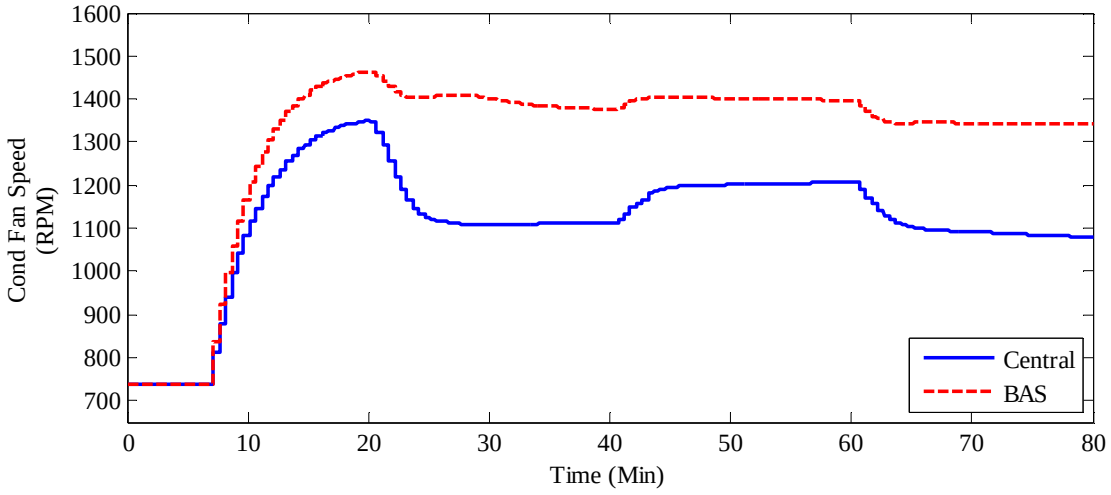

Figure 7.26 Actuator inputs ω_{fc} from each control architecture for baseline scenario.

7.6.2 ESC Scenario

In the presence of unknown disturbances and nonlinearity, modifying the BAS control approach with ESC may further improve the efficiency of the system. For VRF systems, the condenser fan speed is a natural choice of input signal to modify with ESC. As mentioned in Chapter 5, ESC should be used to modify one of the $\mathbf{S}_0$ subsystem inputs. Therefore, either the compressor speed or the condenser fan speed should be used. Additionally, as seen in the B_{00} matrix from (7.47), the compressor directly influences the P_q and T_{rq} states and, therefore, directly influences $T_{SH,q}$, which is a performance objective for the system. Alternatively, the condenser fan only directly affects the condenser pressure. It is interesting to note that increasing the condenser fan speed, which requires an increase in power, will decrease the condenser pressure. In turn, this decrease in condenser pressure decreases the compressor power. Therefore, there is a trade-off between condenser fan power and compressor power. Using ESC to modify the condenser fan speed, the combination of condenser fan speed and compressor speed which minimizes total system power can be found even in the presence of unknown disturbances and system nonlinearity. The following set of figures is used to show the potential improvements ESC can provide when used to augment the BAS control architecture.

The un-augmented BAS architecture is used for comparison. Note that the centralized control approach could also be augmented with ESC to further improve system efficiency.

The same disturbances from the previous scenario are applied to the system, however, these disturbances are stretched in time by a factor of 5 since ESC is relatively slow to adapt to changes in operating condition. Table 7.7 shows the ESC parameters used for the VRF system. Fig. 7.27 shows that the addition of ESC has virtually zero effect on the control of the room air temperatures while Fig. 7.28 shows that the sinusoidal oscillations from ESC are not completely removed when controlling the junction superheat but the desired value is still tracked very closely.

Table 7.7 ESC Parameters for VRF System

Parameter	Value
ω	$\pi/200$ rad.
a	20 RPM
k	5ω
ϕ	$50\pi/180$ rad.
ω_h	10ω
ω_l	0.1ω

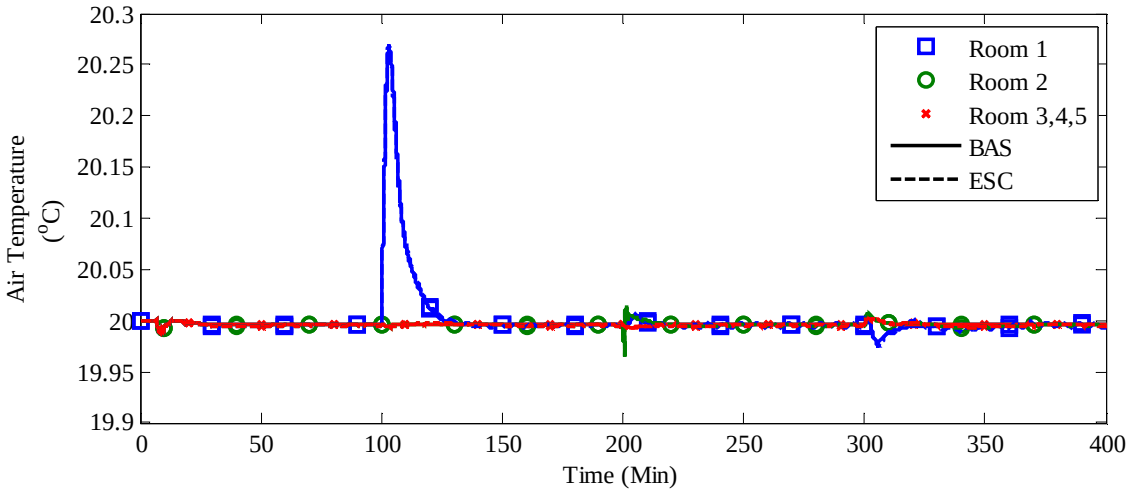

Figure 7.27 Ability of each control architecture to meet local performance objectives by tracking the desired value for T_{ai} for ESC scenario.

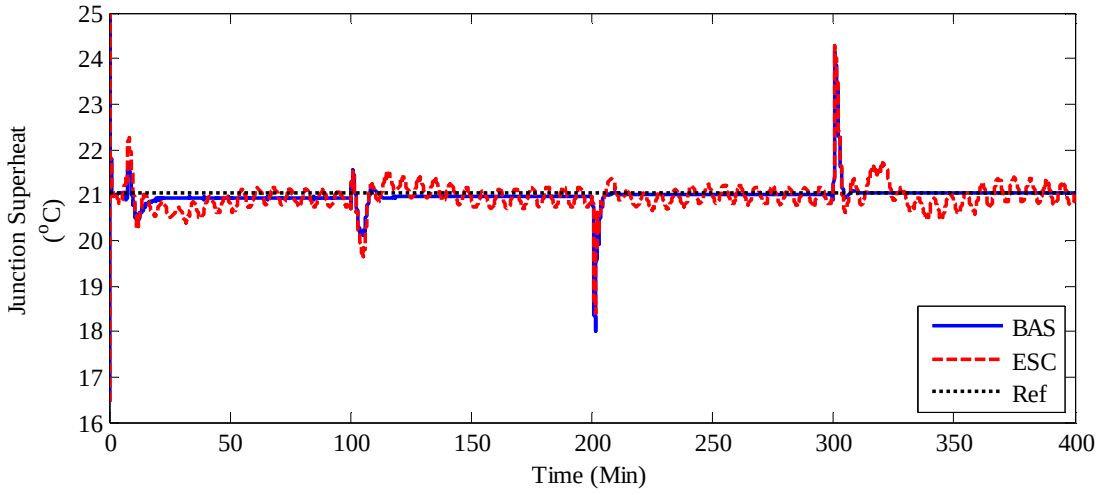

Figure 7.28 Ability of each control architecture to track the desired junction superheat for baseline scenario.

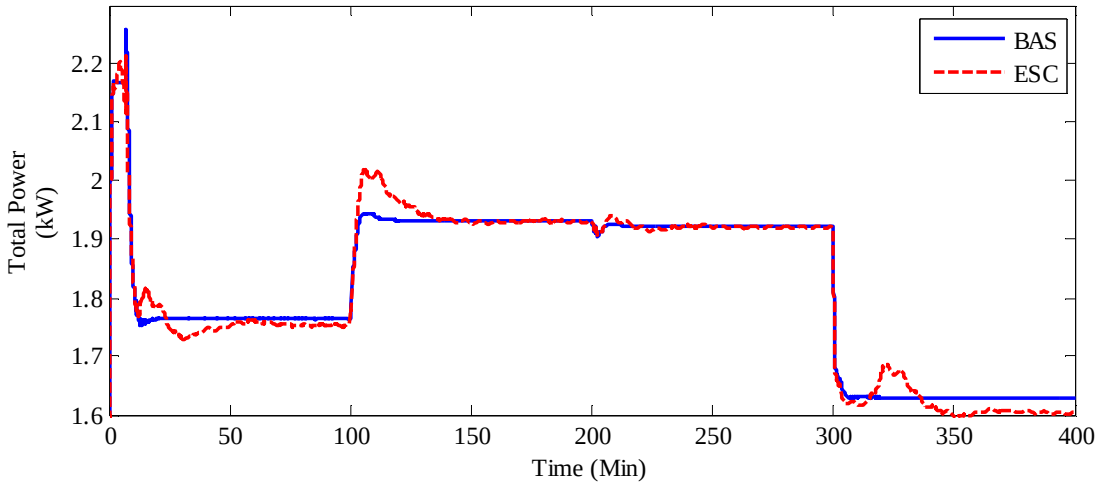

Figure 7.29 Ability of each control architecture to meet the global efficiency objective by minimizing the total power consumption by the actuators for ESC scenario.

Fig. 7.29 shows that the ESC approach is only able to provide slight improvements in system efficiency at steady-state while performing slightly worse during the transients. The major difference between the ESC and un-augmented BAS approaches is seen following the change in ambient temperature at 300 minutes into the simulation. The ESC approach is able to meet the performance objectives while consuming approximately 2% less energy. This shows that the un-augmented BAS approach is able to operate the system very close to the optimal

conditions despite the unknown disturbances and system nonlinearity. It is expected that if the disturbances were larger, the BAS approach would be less optimal and the ESC augmentation would provide greater improvements in efficiency. The actuator input signals are shown in Figs. 7.30-7.33. From Figs. 7.32 and 7.33 it is clear that the ESC approach found a slightly different combination of compressor speed and condenser fan speed which resulted in slightly lower total system power.

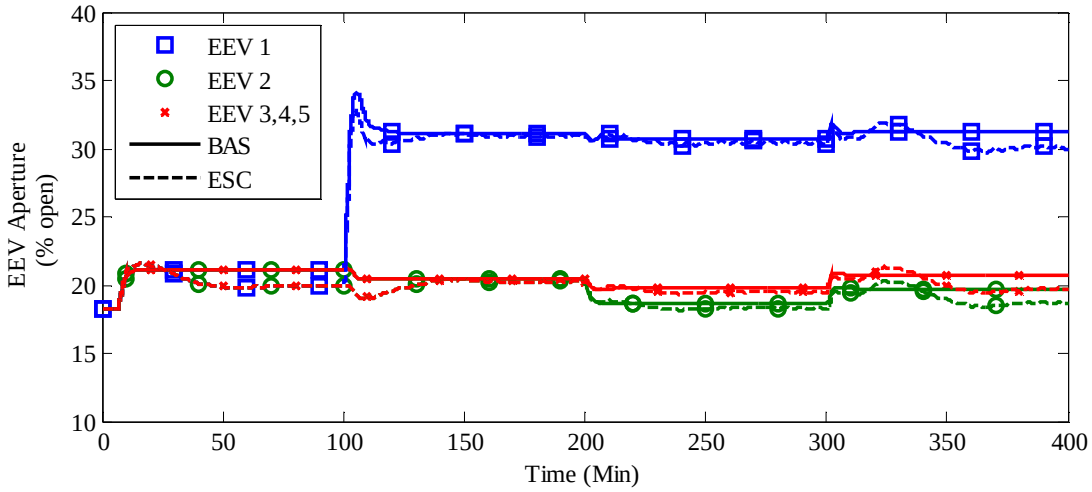

Figure 7.30 Actuator inputs a_{vi} from each control architecture for ESC scenario.

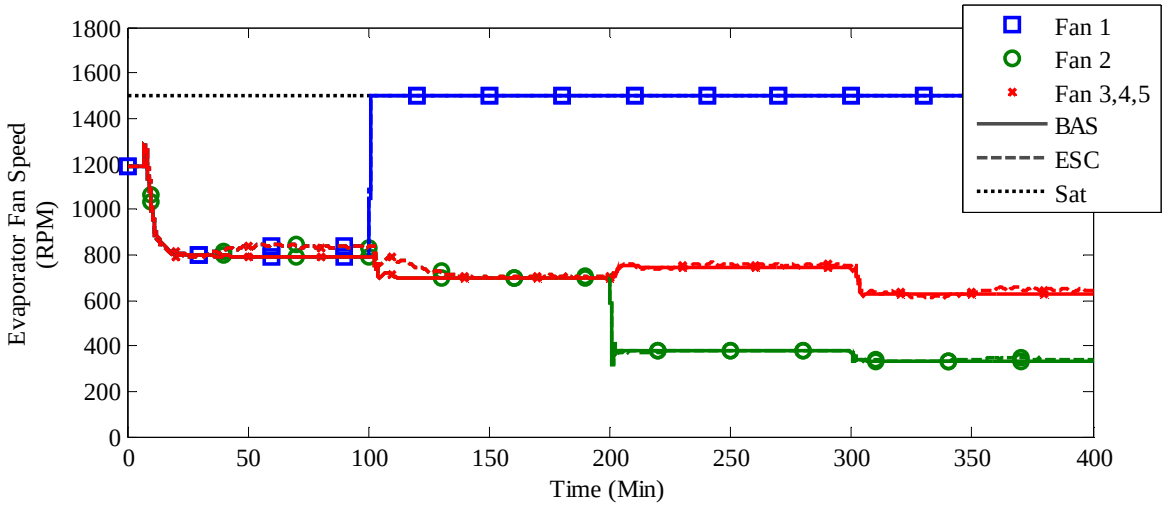

Figure 7.31 Actuator inputs ω_{fi} from each control architecture for ESC scenario.

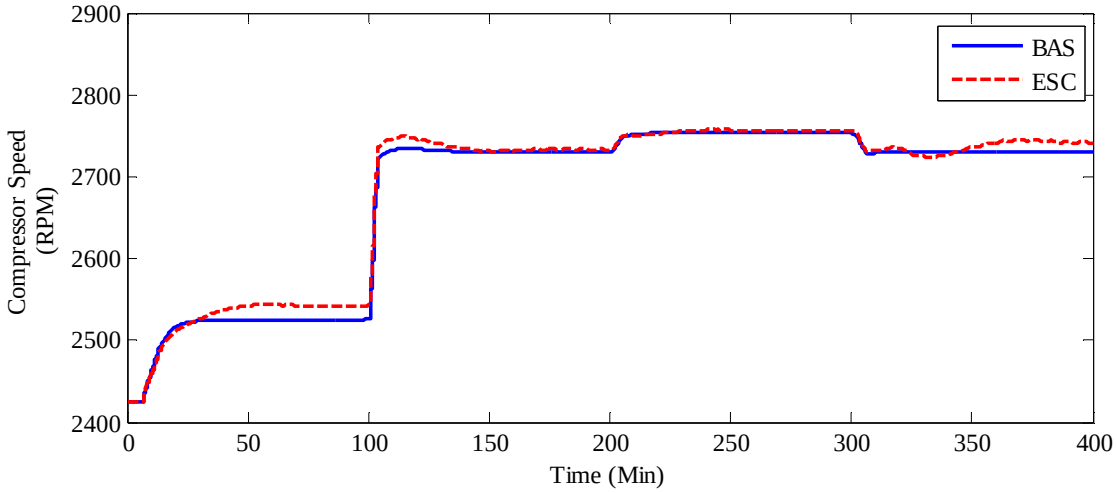

Figure 7.32 Actuator inputs ω_k from each control architecture for ESC scenario.

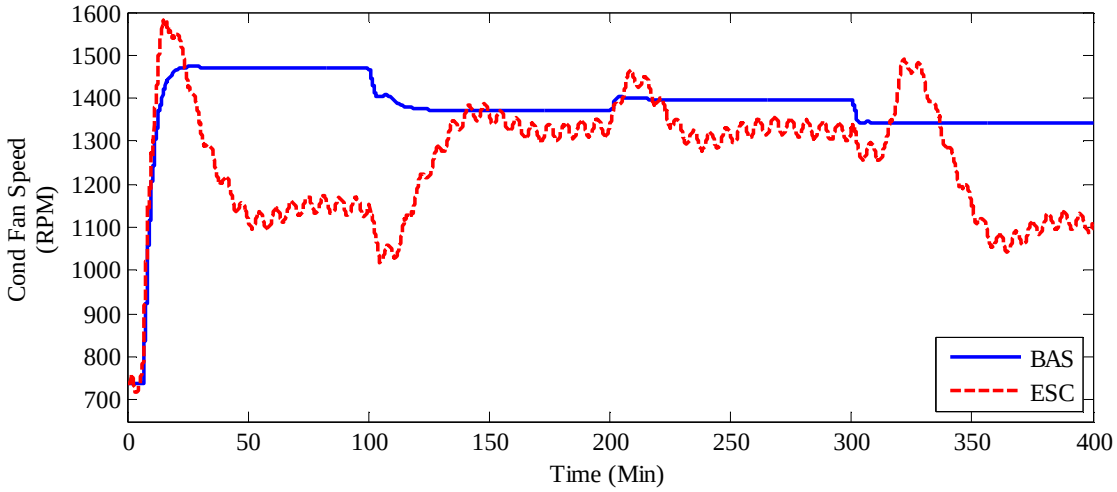

Figure 7.33 Actuator inputs ω_{fc} from each control architecture for ESC scenario.

7.6.3 Realistic Scenario

While the previous scenarios are useful when comparing the performance of various control architectures, the thermal loads and ambient temperature are not very realistic for a typical VRF system. This scenario is used to demonstrate the capabilities of the BAS control architecture using more realistic and taxing operating conditions. The changes in thermal loads and ambient temperature are shown in Fig. 7.34 for the 8 hour simulation.

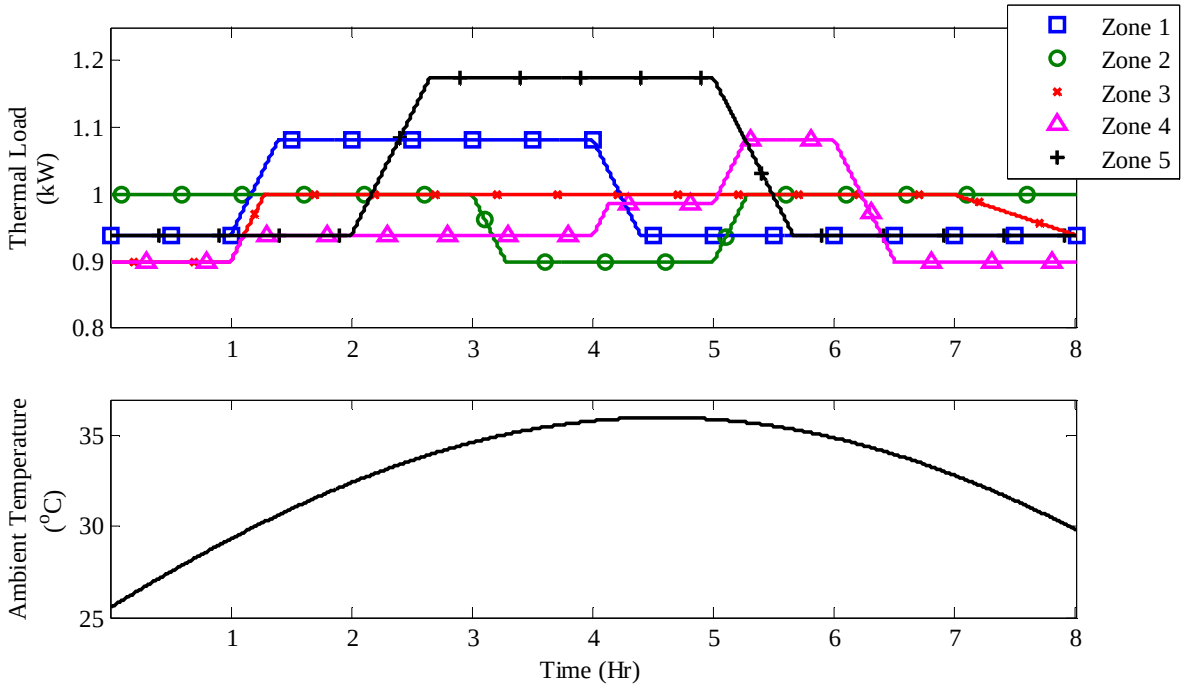

Figure 7.34 System disturbances Q_{Li} and T_{ac} for realistic scenario.

Figs. 7.35 and 7.36 show the air temperature and superheat tracking performance. Both the desired air temperatures and desired junction superheat are tracked very closely. Fig. 7.37 shows the superheat for each of the five evaporators. Interestingly, the superheat for each evaporator can change drastically depending on the thermal load for each room. For example the superheat for evaporator 2 ranges from greater than 20°C to less than 5°C. Despite the large changes in individual evaporator superheat, the junction superheat is maintained within ±0.5°C of the desired value. Also note that the system was linearized about a nominal junction superheat of 21°C, which was used for the desired value in the previous scenarios. However, 21°C superheat is rather large and lower superheat values are typically used in industry. The desired superheat for this scenario has been reduced to 15°C in order to be more realistic. Despite operating fairly far away from the nominal operating condition for which the linear model was identified, the BAS controller is still able to track the desired superheat value.

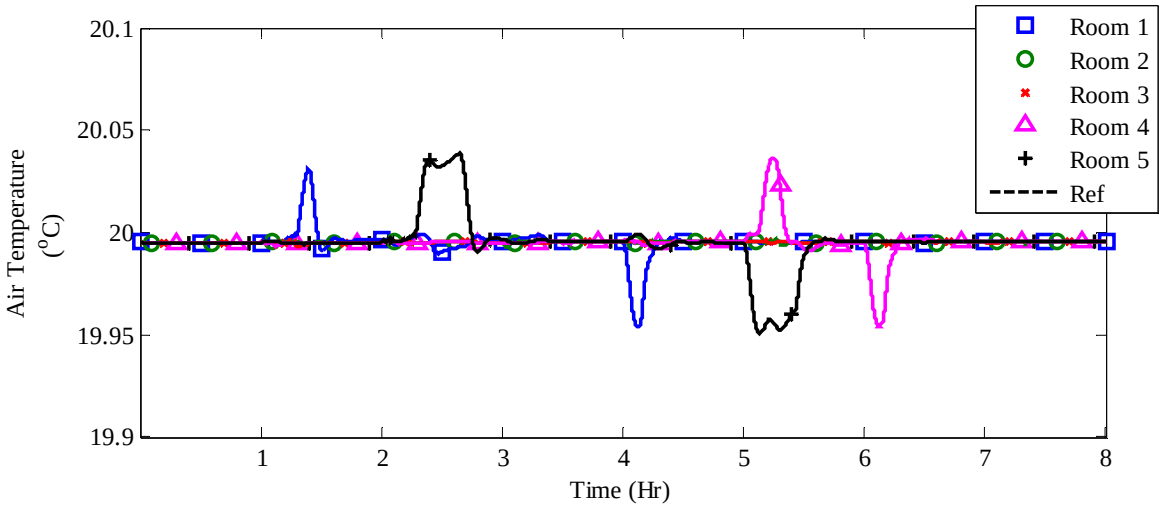

Figure 7.35 Ability of BAS control architecture to meet local performance objectives by tracking the desired value for T_{ai} for realistic scenario.

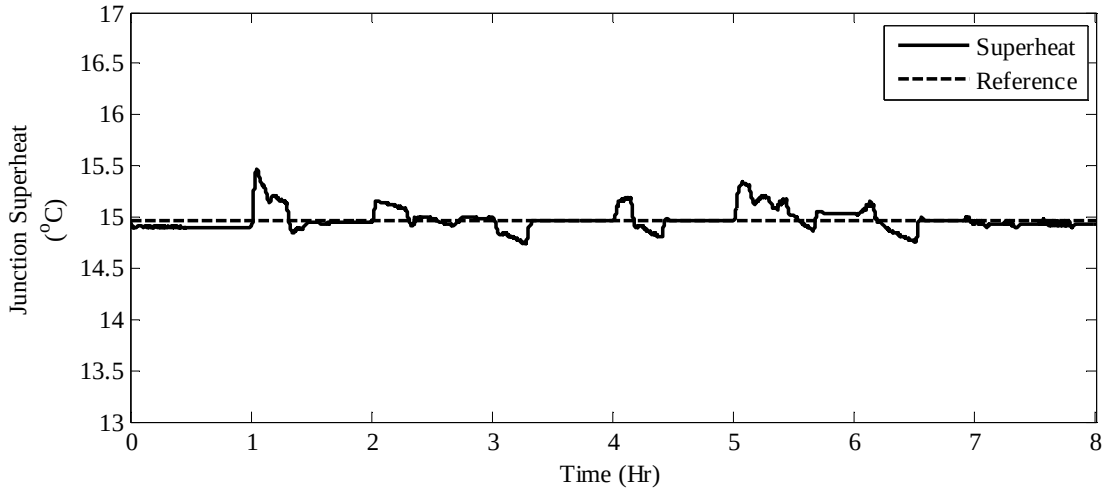

Figure 7.36 Ability of BAS control architecture to track the desire junction superheat for realistic scenario.

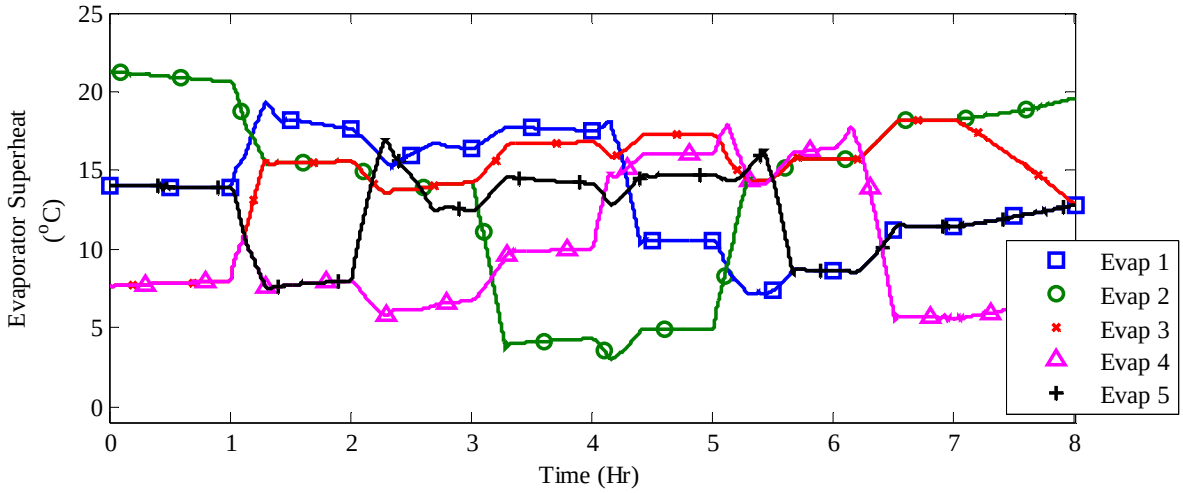

Figure 7.37 Different refrigerant superheat for each evaporator for realistic scenario.

The total power consumption for the system is shown in Fig. 7.38 and it is clear that the power consumption closely follows the change in ambient temperature. Figs. 7.39-7.42 display the actuator input signals throughout the simulation. As seen in Figs. 7.39 and 7.40, the actuator inputs for the EEVs and evaporator fans can be significantly different for each subsystem due to different thermal loads. Despite the differences in actuator inputs, the BAS approach is very successful in controlling the system to meet both the performance and efficiency objectives.

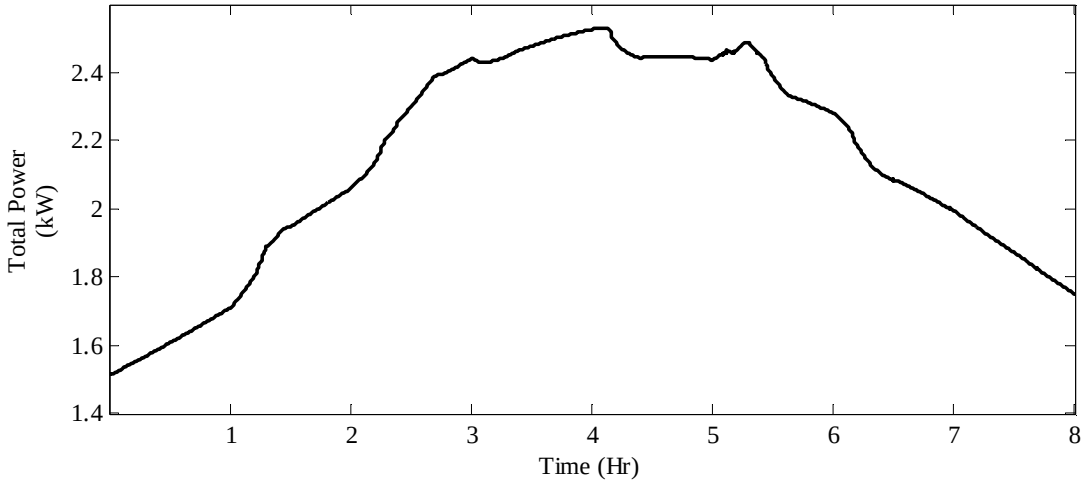

Figure 7.38 Ability of BAS control architecture to meet the global efficiency objective by minimizing the total power consumption by the actuators for realistic scenario.

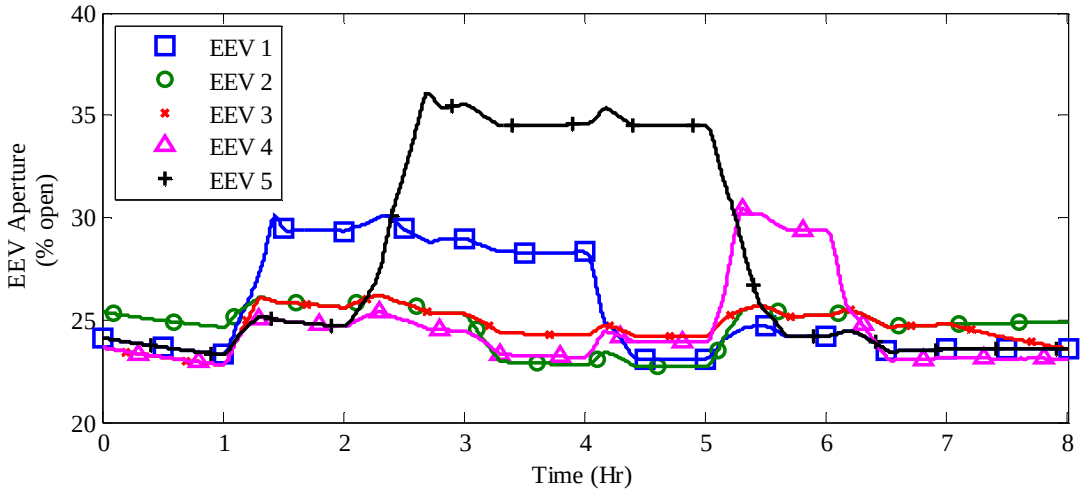

Figure 7.39 Actuator inputs a_{vi} from BAS control architecture for realistic scenario.

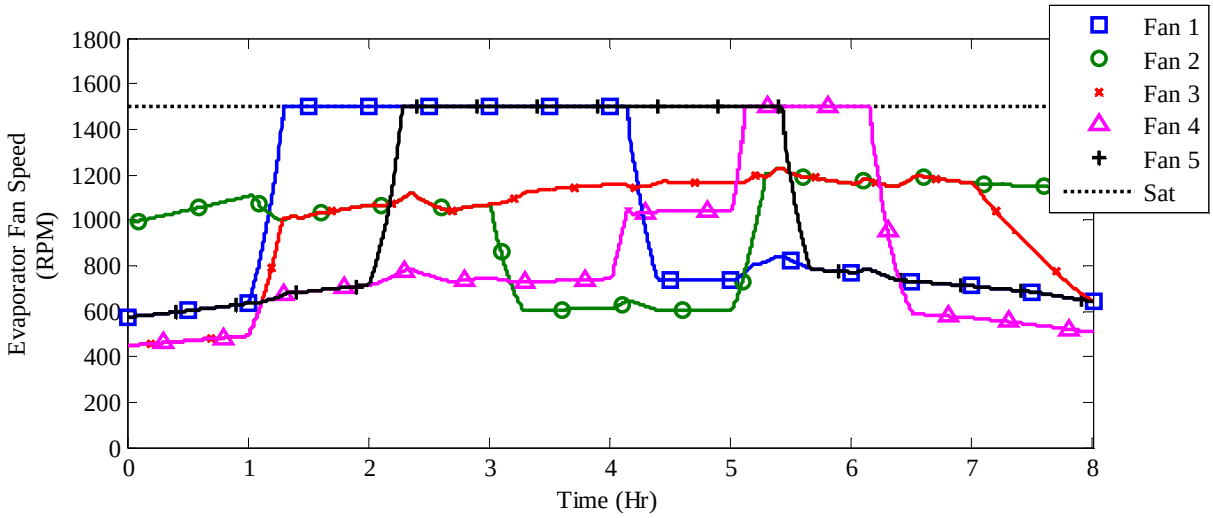

Figure 7.40 Actuator inputs ω_{fi} from BAS control architecture for realistic scenario.

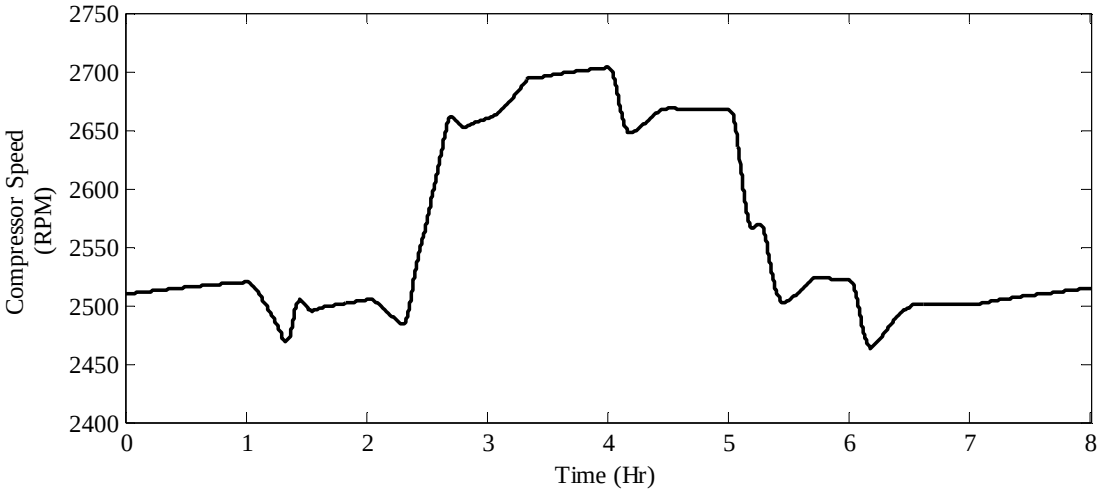

Figure 7.41 Actuator inputs ω_k from BAS control architecture for realistic scenario.

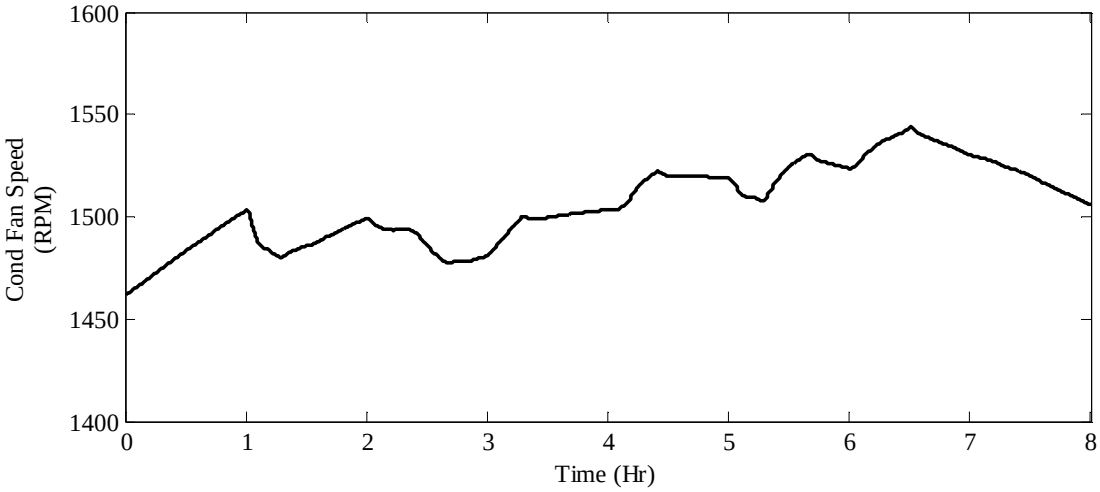

Figure 7.42 Actuator inputs ω_{fc} from BAS control architecture for realistic scenario.

Chapter 8
Conclusion

8.1 Summary of Research Contributions

This thesis develops and analyzes a control architecture for a class of large-scale systems with a Block Arrow Structure (BAS). The large number of states and actuators of large-scale systems often prevents the system from being analyzed and controlled as a whole. Often these systems are divided into multiple subsystems which have interacting dynamics. A typical decentralized control architecture controls each subsystem without explicit knowledge of the interactions between subsystems. If the degree of interaction between subsystems is too large, the performance of a decentralized control approach can be significantly degraded when compared to a centralized approach and may even cause an open-loop stable system to go unstable. The BAS control architecture proposed in this work takes advantage of the structure of BAS systems in order to combine the benefits of both centralized and decentralized control approaches. The BAS approach remains decentralized in the fact that there are multiple controllers that do not cooperate in making control decisions. However, the BAS approach is able to use direct knowledge of the coupling between subsystems, providing a significant improvement in control performance, which can be comparable to the performance of a centralized approach.

In order to evaluate the BAS control approach, centralized, decentralized, and BAS control architectures are developed which use Model Predicative Control (MPC). Using a linear model of the system, MPC makes control decisions by predicting how the system will respond to different control inputs. The controllers implemented in this work take advantage of the ability to directly consider actuator saturation provided by MPC as well as the flexibility when

designing the control cost function. Using a linear example system, the BAS control architecture was found to perform, in terms of meeting both performance and efficiency objectives, significantly better than the decentralized approach while maintaining scalability.

In addition to the BAS control architecture, it was found that Extremum Seeking Control (ESC) can be used to provide even greater system efficiencies. In this work, ESC is used to augment the BAS control architecture by adjusting one of the control inputs to the system with the objective of maximizing system efficiency. Exploiting the BAS structure of the system, the ESC algorithm is used to modify the control decisions for one of the $\mathbf{S}_0$ subsystem inputs such that the effects of the ESC algorithm are distributed to the entire system. In this way, ESC can be used to maximize the efficiency of the entire system and drive the BAS control solution closer to the centralized control solution. The model-free nature of ESC also allows the controller to achieve greater efficiencies in the presence of unknown disturbances and system nonlinearity, which may cause the model-based MPC control approaches to operate away from the optimal conditions.

Finally, the BAS control approach was developed to control a Variable-Refrigerant-Flow (VRF) system. These systems are becoming widely used to meet the cooling demands for many applications including large buildings. For buildings, VRF systems may be used to directly cool over 30 rooms using a single system. With such large systems, decentralized control approaches have been the focus of many research efforts. While most of the previous work has focused on the modeling and control of dual- and triple-evaporator systems, this work develops an approach applicable to an N evaporator system which readily scales to systems where the number of evaporators is large. Through a gray-box modeling approach, it was found that VRF systems are naturally BAS systems and, therefore, benefit from the BAS control approach developed in this thesis. Through a series of simulations, the BAS control architecture was found to be very effective in meeting the performance and efficiency objectives for a 5-evaporator VRF system. While this thesis has shown that there are significant advantages that come from exploiting the BAS structure when making control decisions, there are still several aspects that require additional attention as detailed in the following section.

8.2 Future Work

Future work will continue to develop the BAS control approach through advancements in theory and application.

8.2.1 Theory

There are aspects of the BAS control approach which warrant additional attention from a theoretical perspective. Decentralized control is often more robust to disturbances and system faults than centralized control. Future work will analyze the robustness of the BAS control architecture. In [9] it is mentioned that the BAS control architecture provides the flexibility to add and remove subsystems without significantly changing the controllers. This is in contrast to a centralized control approach which requires a completely new model every time the system is changed. Future work will investigate this functionality and the robustness of the BAS approach to changes in system architecture.

Additionally, the current BAS approach only considers a two-level hierarchy with a common subsystem at the higher level and decoupled subsystems at the lower level. However, a larger class of systems may have multiple levels resulting in a nested BAS structure as described in [4]. It is expected that the BAS framework and associated control benefits can be generalized to the class of system with a nested BAS structure and a control framework for these systems will be developed and tested in future work.

8.2.2 Application

Future work will also utilize the additional control features provided by the MPC framework. By replacing the state tracking and regulation control objectives with upper and lower constraints on system states and outputs, the system can be operated with greater flexibility, which may provide greater system efficiency. MPC also provides the ability to use information about upcoming disturbances to make preemptive control decisions to better react to these disturbances. Future work will explore how information of future disturbances can be used to achieve even better control performance.

While the BAS control architecture was able to control a 5-evaporator VRF system in simulation, a natural extension is to implement the control approach on an experimental system.

As with any experimental platform, the presence of signal noise, unmodeled dynamics and disturbances, and restrictions on sensor location and computational power provides additional challenges when developing and implementing a control strategy. Therefore, the BAS approach needs to be implemented on a physical system in order to determine its practicality.

Another avenue of future work is the application of the BAS control architecture to systems other than VRF systems. This will include systems such as hydraulic and electrical system which also exhibit the block arrow structure. In addition to exploring the potential benefits of applying the BAS approach to these systems, it will be interesting to evaluate the generality of the approach and see if the BAS approach needs to be modified to handle these applications.

References

[1] M. S. Andersen, J. Dahl, and L. Vandenberghe, "Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones," *Mathematical Programming Computation*, vol. 2, pp. 167–201, Aug. 2010.

[2] E. Mizutani and J. W. Demmel, "On separable nonlinear least squares algorithms for neuro-fuzzy modular network learning," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, pp. 2399–2404, 2002.

[3] A. P. Leros and P. P. Groumpos, "Time-invariant BAS-decentralized large-scale linear regulator problem," *International Journal of Control*, vol. 46, no. 1, pp. 129–152, 1987.

[4] A. Zecevic and D. D. Siljak, "A Decomposition-based Control Strategy for Large, Sparse Dynamic Systems," *Mathematical Problems in Engineering*, pp. 1–16, 2005.

[5] M. Morari and J. H. Lee, "Model Predictive Control: Past, Present and Future," in *Joint 6th International Symposium on Process Systems Engineering and 30th European Symposium on Computer Aided Process Engineering*, 1997.

[6] W. Goetzler, "Variable Refrigerant Flow Systems," *ASHRAE Journal*, no. April, pp. 24–31, 2007.

[7] J. P. Koeln and A. G. Alleyne, "Decentralized Controller Analysis and Design for Multi-Evaporator Vapor Compression Systems," *Proceedings of the 2013 American Control Conference*, pp. 437–442, 2013.

[8] "Annual Energy Review 2011."

[9] P. P. Groumpos, "Structural Modelling and Optimisation of Large Scale Systems," in *IEE Proceedings Control Theory and Applications*, 1994.

[10] P. P. Groumpos and A. V. Pagalos, "A two-level structural model for large scale systems," *Computers in Industry*, vol. 36, pp. 147–154, Apr. 1998.

[11] A. V Pagalos, G. S. Stavropoulos, and P. P. Groumpos, “A control strategy for two-level systems with optimization in the feedback loop,” in *Proceedings of 3rd IEEE Conference on Control Applications*, 1994.

[12] D. D. Siljak and A. Zecevic, “Control of large-scale systems: Beyond decentralized feedback,” *Annual Reviews in Control*, vol. 29, pp. 169–179, Jan. 2005.

[13] R. E. Kalman, “Mathematical Description of Linear Dynamical Systems,” *SIAM Journal on Control*, vol. 1, pp. 152–192, 1963.

[14] D. D. Siljak, *Decentralized Control of Complex Systems*. Academic Press, Inc., 1991.

[15] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and Multidisciplinary Optimization*, vol. 26, pp. 369–395, Apr. 2004.

[16] M. A. Henson, “Nonlinear model predictive control: current status and future directions,” *Computers & Chemical Engineering*, vol. 23, pp. 187–202, Dec. 1998.

[17] A. Bemporad, “Model Predictive Control Design: New Trends and Tools,” *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6678–6683, 2006.

[18] “Toolbox, MATLAB Optimization.” The Mathworks Inc., Natick, MA, 2002.

[19] M. Krstic and H. Wang, “Stability of extremum seeking feedback for general nonlinear dynamic systems,” *Automatica*, vol. 36, pp. 595–601, 2000.

[20] W. H. Moase, C. Manzie, and M. J. Brear, “Newton-Like Extremum-Seeking for the Control of Thermoacoustic Instability,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2094–2105, Sep. 2010.

[21] J. Creaby, Y. Li, and J. E. Seem, “Maximizing Wind Turbine Energy Capture using Multivariable Extremum Seeking Control,” *Wind Engineering*, vol. 33, no. 4, pp. 361–387, Jun. 2009.

[22] N. J. Killingsworth and M. Krstic, “PID Tuning Using Extremum Seeking,” *IEEE Control Systems Magazine*, no. February 2006, pp. 70–79, 2006.

[23] J. P. Koeln and A. G. Alleyne, “Optimal Subcooling in Vapor Compression Systems via Extremum Seeking Control,” *Proceedings of ASME Dynamic Systems and Control Conference*, 2013.

[24] M. Krstic, “Towards Faster Adaptation in Extremum Seeking Control,” *Proceedings of 38th Conference on Decision and Control*, pp. 4766–4771, 1999.

[25] E. Elong, M. Krstic, and K. B. Ariyur, “A Case Study of Performance Improvement in Extremum Seeking Control,” *Proceedings of the 2000 American Control Conference*, pp. 428–432, 2000.

[26] G. Gelbert, J. P. Moeck, C. O. Paschereit, and R. King, “Advanced algorithms for gradient estimation in one- and two-parameter extremum seeking controllers,” *Journal of Process Control*, vol. 22, pp. 700–709, Apr. 2012.

[27] C. Zhang and R. Ordonez, “Non-gradient Extremum Seeking Control of Feedback Linearizable Systems with Application to ABS Design,” *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6666–6671, 2006.

[28] K. J. Astrom and L. Rundqwist, “Integrator Windup and How to Avoid It,” *Proceedings of the 1989 American Control Conference*, pp. 1693–1698, 1989.

[29] Y. Tan, D. Nešić, and I. Mareels, “On the choice of dither in extremum seeking systems: A case study,” *Automatica*, vol. 44, pp. 1446–1450, May 2008.

[30] B. P. Rasmussen, “Dynamic modeling for vapor compression systems — Part I : Literature review,” *HVAC&R Research*, vol. 18, no. 5, pp. 934–955, 2012.

[31] Mitsubishi Electric Corporation, *Variable Refrigerant Flow Zoning Systems: 3-30 Tons*. 2011.

[32] N. Jain, S. Sundaram, and A. G. Alleyne, “Stability analysis for decentralized control of multi-evaporator vapor-compression cycle systems,” *Proceedings of the 51st IEEE Conference on Decision and Control*, pp. 7589–7595, Dec. 2012.

[33] Q. Zhang, S. Ye, Y. Li, and X. Wang, “An Enhanced LMI Approach for Mixed H2/H∞ Flight Tracking Control,” *Chinese Journal of Aeronautics*, vol. 24, no. 3, pp. 324–328, Jun. 2011.

[34] J.-L. Lin and T.-J. Yeh, “Identification and control of multi-evaporator air-conditioning systems,” *International Journal of Refrigeration*, vol. 30, pp. 1374–1385, Dec. 2007.

[35] R. Shah, “Dynamic Modeling and Control of Single and Multi-evaporator Subcritical Vapor Compression Systems,” Masters Thesis, Dept. Mech. Eng., Univ. Illinois Urbana-Champaign, Urbana, IL, 2003.

[36] M. S. Elliott, “Decentralized Model Predictive Control of a Multiple Evaporator HVAC System,” Masters Thesis, Dept. Mech. Eng., Texas A&M University, College Station, TX, 2008.

[37] M. Kania, J. Koeln, and A. Alleyne, “A Dynamic Modeling Toolbox for Air Vehicle Vapor Cycle Systems,” in *SAE Power Systems International*, 2012.

[38] B. P. Rasmussen, “Dynamic Modeling and Advanced Control of Air Conditioning and Refrigeration Systems,” Ph.D. Dissertation, Dept. Mech. Eng., Univ. Illinois Urbana-Champaign, Urbana, IL, 2005.

[39] L. F. S. Larsen, “Model Based Control of Refrigeration Systems,” Ph.D. Thesis, Danfoss A/S, Nordborg, Denmark, 2005.

[40] M. Keir, “Dynamic Modeling, Control, and Fault Detection in Vapor Compression Systems,” Masters Thesis, Dept. Mech. Eng., Univ. Illinois Urbana-Champaign, Urbana, IL, 2005.

Appendix A

Example System MATLAB Code

A.1 Parameters and System Development

The following MATLAB code is used to develop the centralized, decentralized, BAS, and ESC controllers for the example system from Chapter 2.

```
% Parameters
% Define parameters for each subsystem
% Subsystem S0
Ce      = 1e-2;
Cg      = 5e-2;
Rg      = 5;
Kg      = 0.2;
alpha_1 = 0.002;
alpha_2 = 0.002;
alpha_3 = 0.008;
% Subsystem S1
C1      = 6e-2;
Ca1     = 2;
R1      = 15;
K1      = 0.3;
Ra1     = 10;
Ka1     = 0.18;
Re1     = 300;
% Subsystem S2
C2      = 3e-2;
Ca2     = 4;
R2      = 22;
K2      = 0.172;
Ra2     = 45;
Ka2     = 0.28;
Re2     = 400;
% Subsystem S3
C3      = 1e-2;
Ca3     = 5;
R3      = 20;
K3      = 0.12;
```

```
Ra3     = 20;
Ka3     = 0.46;
Re3     = 50;
% Define simplifying notation
R_bar1  = (1/R1 + 1/Ra1 + 1/Re1);
R_bar2  = (1/R2 + 1/Ra2 + 1/Re2);
R_bar3  = (1/R3 + 1/Ra3 + 1/Re3);
% Nominal actuator inputs (all nominal inputs are 50)
u0      = 50;
% Actuator costs (actuators for S1,S2,S3 have the same costs)
ai      = 1/100;  % (3/100 for Modified BAS)
bi      = 2*ai*u0 + 0;
ci      = 0;
aai     = 1/100;  % (3/100 for Modified BAS)
bai     = 2*aai*u0 + 0;
cai     = 0;
at      = 1/100;
bt      = 2*at*u0 + 0;
ct      = 0;
ag      = 2/100;
bg      = 2*ag*u0 + 0;
cg      = 0;
% Initial Conditions for Simulation
Vg0     = 80;
Ve0     = 20;
V10     = 50;
Va10    = 30;
V20     = 60;
Va20    = 20;
V30     = 40;
Va30    = 25;
%% Matrices
% Build matrices A,B,V,C,D for system S
% Matrix A
a1      = -R_bar1/C1;
b1      = 1/(C1*Ra1);
c1      = 1/(Ca1*Ra1);
d1      = -1/(Ca1*Ra1);
block_a11   = [a1 b1; c1 d1];
a2      = -R_bar2/C2;
b2      = 1/(C2*Ra2);
c2      = 1/(Ca2*Ra2);
d2      = -1/(Ca2*Ra2);
block_a22   = [a2 b2; c2 d2];
a3      = -R_bar3/C3;
b3      = 1/(C3*Ra3);
c3      = 1/(Ca3*Ra3);
d3      = -1/(Ca3*Ra3);
block_a33   = [a3 b3; c3 d3];
A11     = blkdiag(block_a11,block_a22,block_a33);
block_a10   = [1/(C1*R1) 1/(C1*Re1); 0 0];
block_a20   = [1/(C2*R2) 1/(C2*Re2); 0 0];
block_a30   = [1/(C3*R3) 1/(C3*Re3); 0 0];
A10     = [block_a10; block_a20; block_a30];
block_a01   = [1/(Cg*R1) 0; 1/(Ce*Re1) 0];
```

```
block_a02   = [1/(Cg*R2) 0; 1/(Ce*Re2) 0];
block_a03   = [1/(Cg*R3) 0; 1/(Ce*Re3) 0];
A01     = [block_a01 block_a02 block_a03];
A00     = [-1/Cg*(1/Rg+(1/R1+1/R2+1/R3)-alpha_1) alpha_2/Cg;...
           -alpha_1/Ce -1/Ce*((1/Re1+1/Re2+1/Re3)+alpha_2)];
% Matrix B
e1      = K1/(C1*R1);
f1      = -Ka1/(C1*Ra1);
g1      = 0;
h1      = Ka1/(Ca1*Ra1);
block_b11   = [e1 f1; g1 h1];
e2      = K2/(C2*R2);
f2      = -Ka2/(C2*Ra2);
g2      = 0;
h2      = Ka2/(Ca2*Ra2);
block_b22   = [e2 f2; g2 h2];
e3      = K3/(C3*R3);
f3      = -Ka3/(C3*Ra3);
g3      = 0;
h3      = Ka3/(Ca3*Ra3);
block_b33   = [e3 f3; g3 h3];
B11     = blkdiag(block_b11,block_b22,block_b33);
B10     = zeros(6,2);
block_b01   = [-K1/(Cg*R1) 0; 0 0];
block_b02   = [-K2/(Cg*R2) 0; 0 0];
block_b03   = [-K3/(Cg*R3) 0; 0 0];
B01     = [block_b01 block_b02 block_b03];
B00     = [alpha_3/Cg Kg/(Cg*Rg); -alpha_3/Ce 0];
% Matrix V
v11     = [0; -1/Ca1];
v22     = [0; -1/Ca2];
v33     = [0; -1/Ca3];
V11     = blkdiag(v11,v22,v33);
V12     = zeros(6,1);
V21     = zeros(2,3);
V22     = [1/(Cg*Rg); 0];
% Matrix C
C   = [eye(7) zeros(7,1);...
       1/Re1 0 1/Re2 0 1/Re3 0 0 -(1/Re1+1/Re2+1/Re3)];
%% Common Information
% Sample time
Sys.dt      = 10;
% Control Horizon
Sys.Nu      = 15;
% Prediction Horizon
Sys.Np      = 30;
% Objectives vs delta U
Sys.gamma_a     = 0.98;
% Performance vs efficiency
Sys.gamma_b     = 0.01;
% Error vs integral
Sys.gamma_c     = 0.9;
% Va tracking performance objective
q_Va        = 1;
% Ie tracking performance objective
```

```
q_Ie        = 50;
% Actuator ui
q_ui        = 1;
% Actuator uai
q_uai       = 1;
% Actuator ut
q_ut        = 10;
% Actuator ug
q_ug        = 1;
% Constraints
Min_ui  = 0-u0;
Min_uai = 0-u0;
Min_ut  = 0-u0;
Min_ug  = 0-u0;
Max_ui  = 100-u0;
Max_uai = 100-u0;
Max_ut  = 100-u0;
Max_ug  = 100-u0;
%% Centralized System Information
% Model
Sys.A   = [A11 A10; A01 A00];
Sys.B   = [B11 B10; B01 B00];
Sys.V   = [V11 V12; V21 V22];
Sys.C   = C;
% Number of States
Sys.ns      = 8;
% Number of Inputs
Sys.nu      = 8;
% Number of Disturbances
Sys.nd      = 4;
% Number of References
Sys.nr      = 4;
% States with desired values
Sys.M       = blkdiag([0 1],[0 1],[0 1],[0 1]);
% Weightings
Sys.qe  = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Va q_Va q_Va q_Ie];
Sys.qi  = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Va q_Va q_Va q_Ie];
Sys.q2  = [q_ui q_uai q_ui q_uai q_ui q_uai q_ut q_ug];
Sys.qa  = [ai aai ai aai ai aai at ag];
Sys.qb1 = [bi bai bi bai bi bai bt bg];
% Min and Max inputs
Sys.Mins = [Min_ui;Min_uai;Min_ui;Min_uai;Min_ui;Min_uai;Min_ut;Min_ug];
Sys.Maxs = [Max_ui;Max_uai;Max_ui;Max_uai;Max_ui;Max_uai;Max_ut;Max_ug];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
T_bar = Outputs.T_bar;
S_bar = Outputs.S_bar;
R_bar = Outputs.R_bar;
P_bar = Outputs.P_bar;
N_bar = Outputs.N_bar;
n_bar = Outputs.n_bar;
Q1 = Outputs.Q1;
Q2 = Outputs.Q2;
Qa = Outputs.Qa;
qb = Outputs.qb;
```

```
U_min = Outputs.U_min;
U_max = Outputs.U_max;
%% BAS1 System Information
% Model
Sys.A   = [block_a11 block_a10; block_a01 A00];
Sys.B   = [block_b11 zeros(2,2); block_b01 B00];
Sys.V   = [v11 zeros(2,1);zeros(2,1) V22];
Sys.C   = [eye(3) zeros(3,1); 1/Re1 0 0 -(1/Re1+1/Re2+1/Re3)];
% Number of States
Sys.ns      = 4;
% Number of Inputs
Sys.nu      = 4;
% Number of Disturbances
Sys.nd      = 2;
% Number of References
Sys.nr      = 2;
% States with desired values
Sys.M       = blkdiag([0 1],[0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Va q_Ie];
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Va q_Ie];
Sys.q2 = [q_ui q_uai q_ut q_ug];
Sys.qa = [ai aai at ag];
Sys.qb1 = [bi bai bt bg];
% Min and Max inputs
Sys.Mins = [Min_ui;Min_uai;Min_ut;Min_ug];
Sys.Maxs = [Max_ui;Max_uai;Max_ut;Max_ug];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
BAS1_T_bar = Outputs.T_bar;
BAS1_S_bar = Outputs.S_bar;
BAS1_R_bar = Outputs.R_bar;
BAS1_P_bar = Outputs.P_bar;
BAS1_N_bar = Outputs.N_bar;
BAS1_n_bar = Outputs.n_bar;
BAS1_Q1 = Outputs.Q1;
BAS1_Q2 = Outputs.Q2;
BAS1_Qa = Outputs.Qa;
BAS1_qb = Outputs.qb;
BAS1_U_min = Outputs.U_min;
BAS1_U_max = Outputs.U_max;
%% BAS2 System Information
% Model
Sys.A   = [block_a22 block_a20; block_a02 A00];
Sys.B   = [block_b22 zeros(2,2); block_b02 B00];
Sys.V   = [v22 zeros(2,1);zeros(2,1) V22];
Sys.C   = [eye(3) zeros(3,1); 1/Re2 0 0 -(1/Re1+1/Re2+1/Re3)];
% Number of States
Sys.ns      = 4;
% Number of Inputs
Sys.nu      = 4;
% Number of Disturbances
Sys.nd      = 2;
% Number of References
Sys.nr      = 2;
```

```
% States with desired values
Sys.M       = blkdiag([0 1],[0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Va q_Ie];
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Va q_Ie];
Sys.q2 = [q_ui q_uai q_ut q_ug];
Sys.qa = [ai aai at ag];
Sys.qb1 = [bi bai bt bg];
% Min and Max inputs
Sys.Mins = [Min_ui;Min_uai;Min_ut;Min_ug];
Sys.Maxs = [Max_ui;Max_uai;Max_ut;Max_ug];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
BAS2_T_bar = Outputs.T_bar;
BAS2_S_bar = Outputs.S_bar;
BAS2_R_bar = Outputs.R_bar;
BAS2_P_bar = Outputs.P_bar;
BAS2_N_bar = Outputs.N_bar;
BAS2_n_bar = Outputs.n_bar;
BAS2_Q1 = Outputs.Q1;
BAS2_Q2 = Outputs.Q2;
BAS2_Qa = Outputs.Qa;
BAS2_qb = Outputs.qb;
BAS2_U_min = Outputs.U_min;
BAS2_U_max = Outputs.U_max;
%% BAS3 System Information
% Model
Sys.A   = [block_a33 block_a30; block_a03 A00];
Sys.B   = [block_b33 zeros(2,2); block_b03 B00];
Sys.V   = [v33 zeros(2,1);zeros(2,1) V22];
Sys.C   = [eye(3) zeros(3,1); 1/Re3 0 0 -(1/Re1+1/Re2+1/Re3)];
% Number of States
Sys.ns      = 4;
% Number of Inputs
Sys.nu      = 4;
% Number of Disturbances
Sys.nd      = 2;
% Number of References
Sys.nr      = 2;
% States with desired values
Sys.M       = blkdiag([0 1],[0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Va q_Ie];
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Va q_Ie];
Sys.q2 = [q_ui q_uai q_ut q_ug];
Sys.qa = [ai aai at ag];
Sys.qb1 = [bi bai bt bg];
% Min and Max inputs
Sys.Mins = [Min_ui;Min_uai;Min_ut;Min_ug];
Sys.Maxs = [Max_ui;Max_uai;Max_ut;Max_ug];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
BAS3_T_bar = Outputs.T_bar;
BAS3_S_bar = Outputs.S_bar;
BAS3_R_bar = Outputs.R_bar;
```

```
BAS3_P_bar = Outputs.P_bar;
BAS3_N_bar = Outputs.N_bar;
BAS3_n_bar = Outputs.n_bar;
BAS3_Q1 = Outputs.Q1;
BAS3_Q2 = Outputs.Q2;
BAS3_Qa = Outputs.Qa;
BAS3_qb = Outputs.qb;
BAS3_U_min = Outputs.U_min;
BAS3_U_max = Outputs.U_max;
%% Decentral 1 System Information
% Model
Sys.A   = block_a11;
Sys.B   = block_b11;
Sys.V   = v11;
Sys.C   = eye(2);
% Number of States
Sys.ns      = 2;
% Number of Inputs
Sys.nu      = 2;
% Number of Disturbances
Sys.nd      = 1;
% Number of References
Sys.nr      = 1;
% States with desired values
Sys.M       = blkdiag([0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*q_Va;
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*q_Va;
Sys.q2 = [q_ui q_uai];
Sys.qa = [ai aai];
Sys.qb1 = [bi bai];
% Min and Max inputs
Sys.Mins = [Min_ui;Min_uai];
Sys.Maxs = [Max_ui;Max_uai];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
D1_T_bar = Outputs.T_bar;
D1_S_bar = Outputs.S_bar;
D1_R_bar = Outputs.R_bar;
D1_P_bar = Outputs.P_bar;
D1_N_bar = Outputs.N_bar;
D1_n_bar = Outputs.n_bar;
D1_Q1 = Outputs.Q1;
D1_Q2 = Outputs.Q2;
D1_Qa = Outputs.Qa;
D1_qb = Outputs.qb;
D1_U_min = Outputs.U_min;
D1_U_max = Outputs.U_max;
%% Decentral 2 System Information
% Model
Sys.A   = block_a22;
Sys.B   = block_b22;
Sys.V   = v22;
Sys.C   = eye(2);
% Number of States
```

```
Sys.ns      = 2;
% Number of Inputs
Sys.nu      = 2;
% Number of Disturbances
Sys.nd      = 1;
% Number of References
Sys.nr      = 1;
% States with desired values
Sys.M       = blkdiag([0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*q_Va;
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*q_Va;
Sys.q2 = [q_ui q_uai];
Sys.qa = [ai aai];
Sys.qb1 = [bi bai];
% Min and Max inputs
Sys.Mins = [Min_ui;Min_uai];
Sys.Maxs = [Max_ui;Max_uai];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
D2_T_bar = Outputs.T_bar;
D2_S_bar = Outputs.S_bar;
D2_R_bar = Outputs.R_bar;
D2_P_bar = Outputs.P_bar;
D2_N_bar = Outputs.N_bar;
D2_n_bar = Outputs.n_bar;
D2_Q1 = Outputs.Q1;
D2_Q2 = Outputs.Q2;
D2_Qa = Outputs.Qa;
D2_qb = Outputs.qb;
D2_U_min = Outputs.U_min;
D2_U_max = Outputs.U_max;
%% Decentral 3 System Information
% Model
Sys.A   = block_a33;
Sys.B   = block_b33;
Sys.V   = v33;
Sys.C   = eye(2);
% Number of States
Sys.ns      = 2;
% Number of Inputs
Sys.nu      = 2;
% Number of Disturbances
Sys.nd      = 1;
% Number of References
Sys.nr      = 1;
% States with desired values
Sys.M       = blkdiag([0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*q_Va;
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*q_Va;
Sys.q2 = [q_ui q_uai];
Sys.qa = [ai aai];
Sys.qb1 = [bi bai];
% Min and Max inputs
```

```
Sys.Mins = [Min_ui;Min_uai];
Sys.Maxs = [Max_ui;Max_uai];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
D3_T_bar = Outputs.T_bar;
D3_S_bar = Outputs.S_bar;
D3_R_bar = Outputs.R_bar;
D3_P_bar = Outputs.P_bar;
D3_N_bar = Outputs.N_bar;
D3_n_bar = Outputs.n_bar;
D3_Q1 = Outputs.Q1;
D3_Q2 = Outputs.Q2;
D3_Qa = Outputs.Qa;
D3_qb = Outputs.qb;
D3_U_min = Outputs.U_min;
D3_U_max = Outputs.U_max;
%% Decentral 0 System Information
% Model
Sys.A   = A00;
Sys.B   = B00;
Sys.V   = V22;
Sys.C   = [1 0; 0 -(1/Re1+1/Re2+1/Re3)];
% Number of States
Sys.ns      = 2;
% Number of Inputs
Sys.nu      = 2;
% Number of Disturbances
Sys.nd      = 1;
% Number of References
Sys.nr      = 1;
% States with desired values
Sys.M       = blkdiag([0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Ie];
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Ie];
Sys.q2 = [q_ut q_ug];
Sys.qa = [at ag];
Sys.qb1 = [bt bg];
% Min and Max inputs
Sys.Mins = [Min_ut;Min_ug];
Sys.Maxs = [Max_ut;Max_ug];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
D0_T_bar = Outputs.T_bar;
D0_S_bar = Outputs.S_bar;
D0_R_bar = Outputs.R_bar;
D0_P_bar = Outputs.P_bar;
D0_N_bar = Outputs.N_bar;
D0_n_bar = Outputs.n_bar;
D0_Q1 = Outputs.Q1;
D0_Q2 = Outputs.Q2;
D0_Qa = Outputs.Qa;
D0_qb = Outputs.qb;
D0_U_min = Outputs.U_min;
D0_U_max = Outputs.U_max;
```

```
%% ESC Parameters
a0      = 5;
ug0     = 0;
delta   = 1;
omega   = 2*pi/200;
phi     = 15*pi/180;
k_prime = 3e0;
omega_l_prime = 0.1;
omega_h_prime = 10;
k       = omega*delta*k_prime;
omega_l = omega*delta*omega_l_prime;
omega_h = omega*delta*omega_h_prime;
```

A.2 MPC Formulation

The following MATLAB code contains the function *MPC_Formulation.m* which is called in the code above to generate the variables used for the MPC controllers.

```
function [Outputs] = MPC_Formulation(Sys);
A = Sys.A;
B = Sys.B;
V = Sys.V;
C = Sys.C;
ns = Sys.ns;
nu = Sys.nu;
nd = Sys.nd;
nr = Sys.nr;
M = Sys.M;
dt = Sys.dt;
Nu = Sys.Nu;
Np = Sys.Np;
gamma_a = Sys.gamma_a;
gamma_b = Sys.gamma_b;
gamma_c = Sys.gamma_c;
qe = Sys.qe;
qi = Sys.qi;
q2 = Sys.q2;
qa = Sys.qa;
qb1 = Sys.qb1;
Mins = Sys.Mins;
Maxs = Sys.Maxs;
%% Augment system into error system S_hat
A_hat   = C*A/C;
B_hat   = C*B;
V_hat   = [C*V C*A/C*pinv(M)];
C_hat   = eye(ns);
D_hat   = zeros(ns,nu);
%% Discretize System to get S_d
% Centralized System
sys_c   = ss(A_hat,[B_hat V_hat],C_hat,[D_hat zeros(size(V_hat))]);
sys_d_prime = c2d(sys_c,dt);
A_d     = sys_d_prime.a;
```

```
B_d     = sys_d_prime.b(:,1:nu);
V_d     = sys_d_prime.b(:,nu+1:nu+nd+nr);
C_d     = sys_d_prime.c;
D_d     = sys_d_prime.d(:,1:nu);
%% Add Integrator and x_u states to get S_d_bar
A_bar   = [A_d          zeros(ns,nr)    B_d;...
           M*dt         eye(nr)         zeros(nr,nu);...
           zeros(nu,ns) zeros(nu,nr)    eye(nu)];
B_bar   = [B_d; zeros(nr,nu); eye(nu)];
V_bar   = [V_d; zeros(nr,nd+nr); zeros(nu,nd+nr)];
C_bar   = [M            zeros(nr)   zeros(nr,nu);...
           zeros(nr,ns) eye(nr)     zeros(nr,nu)];
%% Generate lifted system matrices
% T_bar
T_bar = zeros((ns+nr+nu)*Np,ns+nr+nu);
for i = 1:Np
    T_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),:) = A_bar^i;
end
% S_bar
S_bar = zeros((ns+nr+nu)*Np,nu*Nu);
for i = 1:Np
    for j = 1:Nu
        if i-j < 0
            S_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),(j-1)*nu+1:j*nu) =
zeros((ns+nr+nu),nu);
        elseif i-j == 0
            S_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),(j-1)*nu+1:j*nu) = B_bar;
        elseif i-j > 0
            S_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),(j-1)*nu+1:j*nu) =
A_bar^(i-j)*B_bar;
        end
    end
end
% R_bar
R_prev = zeros(size(V_bar));
R_bar = zeros((ns+nr+nu)*Np,nd+nr);
for i = 1:Np
    R_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),:) = A_bar^(i-1)*V_bar+R_prev;
    R_prev = A_bar^(i-1)*V_bar+R_prev;
end
% P_bar
P_bar = zeros(2*nr*Np,(ns+nr+nu)*Np);
for i = 1:Np
    P_bar((i-1)*2*nr+1:i*2*nr,(i-1)*(ns+nr+nu)+1:i*(ns+nr+nu)) = C_bar;
end
% N_bar
N_bar = zeros(nu*Np,nu*Nu);
for i = 1:Np
    for j = 1:Nu
        if i >= j
            N_bar((i-1)*nu+1:i*nu,(j-1)*nu+1:j*nu) = eye(nu);
        end
    end
end
% n_bar
```

```
n_bar = zeros(nu*Np,nu);
n_bar(1:nu,1:nu) = eye(nu);
%% Weighting Matrices
% Q1
q1 = [qe qi];
for i = 1:Np
    q1_all((i-1)*2*nr+1:i*2*nr) = q1;
end
Q1 = diag(q1_all);
% Q2
for i = 1:Nu
    q2_all((i-1)*nu+1:i*nu) = q2;
end
Q2 = (1-gamma_a)*diag(q2_all);
% Qa
for i = 1:Np
    qa_all((i-1)*nu+1:i*nu) = qa;
end
Qa = gamma_a*(1-gamma_b)*diag(qa_all);
% qb
for i = 1:Np
    qb_all((i-1)*nu+1:i*nu) = qb1;
end
qb = gamma_a*(1-gamma_b)*qb_all;
%% Constraints
% Lifted Constraints
for i = 1:Np
    U_min((i-1)*nu+1:i*nu,1) = Mins;
end
for i = 1:Np
    U_max((i-1)*nu+1:i*nu,1) = Maxs;
end
%% Outputs
Outputs.T_bar = T_bar;
Outputs.S_bar = S_bar;
Outputs.R_bar = R_bar;
Outputs.P_bar = P_bar;
Outputs.N_bar = N_bar;
Outputs.n_bar = n_bar;
Outputs.Q1 = Q1;
Outputs.Q2 = Q2;
Outputs.Qa = Qa;
Outputs.qb = qb;
Outputs.U_min = U_min;
Outputs.U_max = U_max;
```

A.3 MPC Code

The following MATLAB code contains the function *MPC.m* which uses the MATLAB function *quadprog* to solve the MPC optimization problem at every same time of the simulation.

```
function [Controls,DV,t] = MPC(T_bar,S_bar,R_bar,P_bar,N_bar,n_bar,Q1,Q2,Qa,
```

```
      qb,U_min,U_max,X,R0,V0,U0,Dist)
eml.extrinsic('quadprog','optimset','tic','toc')
tic;
x_bar0 = [X;U0];
d = [Dist;R0];
a = [-N_bar;N_bar];
b = [-(U_min-n_bar*U0);(U_max-n_bar*U0)];
H = zeros(size(Q2));
F = zeros(size(Q2,1),1);
H = S_bar'*P_bar'*Q1*P_bar*S_bar + Q2 + N_bar'*Qa*N_bar;
H = 2*(H+H')/2;
F = 2*S_bar'*P_bar'*Q1*P_bar*T_bar*x_bar0 + 2*S_bar'*P_bar'*Q1*P_bar*R_bar*d
    +2*N_bar'*Qa*n_bar*U0 + N_bar'*qb';
options = optimset('MaxIter',50,'algorithm','active-set');
DV = zeros(size(Q2,1),1);
DV = quadprog(H,F,a,b,[],[],[],[],V0,options);
Controls = DV(1:size(n_bar,2));
t = 0;
t = toc;
```

Appendix B

VRF System MATLAB Code

B.1 Parameters and System Development

The following MATLAB code is used to develop the centralized, decentralized, BAS, and ESC controllers for the VRF system from Chapter 7.

```
% Parameters
load Final_Sys_ID
Coeffs.S.Ca1 = 65;
n = 5;
%% State Space Equations
%% Pc dot
% Pi
a01_11 = 1/(Coeffs.C.Cc*Coeffs.V.R_v1) -
Coeffs.C.Kc/Coeffs.C.Cc*(1/Coeffs.C.Rrc +
1/Coeffs.C.Rac)*Coeffs.E.C1/(Coeffs.C.Kc*Coeffs.C.Cwc);
% Twi
a01_12 = Coeffs.C.Kc/Coeffs.C.Cc*(1/Coeffs.C.Rrc +
1/Coeffs.C.Rac)*Coeffs.E.K1*Coeffs.E.Cw1/(Coeffs.C.Kc*Coeffs.C.Cwc);
% Pc
a00_11 = -1/Coeffs.C.Cc*(Coeffs.K.B_k2 + n/Coeffs.V.R_v1 +
Coeffs.C.Kc/Coeffs.C.Rrc*(Coeffs.C.Lambdarc1*Coeffs.K.B_k2 - Coeffs.C.eta_rc
+ Coeffs.C.Lambdarc2)) - Coeffs.C.Kc/Coeffs.C.Cc*(1/Coeffs.C.Rrc +
1/Coeffs.C.Rac)*Coeffs.C.Cc/(Coeffs.C.Kc*Coeffs.C.Cwc);
% Pq
a00_12 =
Coeffs.K.B_k1/Coeffs.C.Cc*(1+Coeffs.C.Kc*Coeffs.C.Lambdarc1/Coeffs.C.Rrc) -
Coeffs.C.Kc/Coeffs.C.Cc*(1/Coeffs.C.Rrc +
1/Coeffs.C.Rac)*Coeffs.J.Cq/(Coeffs.C.Kc*Coeffs.C.Cwc);
% Trq
a00_13 = -
Coeffs.K.B_k4/Coeffs.C.Cc*(1+Coeffs.C.Kc*Coeffs.C.Lambdarc1/Coeffs.C.Rrc);
% avi
b01_11 = -Coeffs.V.K_v1/(Coeffs.C.Cc*Coeffs.V.R_v1);
% wk
```

```
b00_11 =
Coeffs.K.B_k3/Coeffs.C.Cc*(1+Coeffs.C.Kc*Coeffs.C.Lambdarc1/Coeffs.C.Rrc);
% wfc
b00_12 = -Coeffs.C.Kc*Coeffs.C.Kfc/(Coeffs.C.Cc*Coeffs.C.Rac);
% Tac
v00_1 = Coeffs.C.Kc/(Coeffs.C.Cc*Coeffs.C.Rac);
%% Pq dot
% Pi
a01_21 = 1/(Coeffs.J.Cq*Coeffs.P.R_q1)*(1 + Coeffs.P.K_q3*Coeffs.E.eta_r11 +
Coeffs.P.K_q3*Coeffs.E.eta_r31/Coeffs.V.R_v1);
% Pc
a00_21 = 1/Coeffs.J.Cq*(Coeffs.K.B_k2 -
n*Coeffs.P.K_q3*Coeffs.E.eta_r31/(Coeffs.P.R_q1*Coeffs.V.R_v1));
% Pq
a00_22 = -1/Coeffs.J.Cq*(Coeffs.K.B_k1 + n/Coeffs.P.R_q1);
% Trq
a00_23 = Coeffs.K.B_k4/Coeffs.J.Cq;
% avi
b01_21 = -
Coeffs.P.K_q3*Coeffs.E.eta_r31*Coeffs.V.K_v1/(Coeffs.J.Cq*Coeffs.P.R_q1*Coeff
s.V.R_v1);
% wfi
b01_22 = -1/(Coeffs.J.Cq*Coeffs.P.R_q1)*(Coeffs.P.K_q2 -
Coeffs.P.K_q3*Coeffs.E.eta_r21);
% wk
b00_21 = -1/Coeffs.J.Cq*(Coeffs.K.B_k3 + n*Coeffs.P.K_q1/Coeffs.P.R_q1);
%% Pi dot
% Pi
a11_11 = -1/Coeffs.E.C1*(1/Coeffs.V.R_v1*(1 +
Coeffs.P.K_q3*Coeffs.E.eta_r31/Coeffs.P.R_q1) +
1/Coeffs.P.R_q1*(1+Coeffs.P.K_q3*Coeffs.E.eta_r11) +
Coeffs.E.K1/Coeffs.E.Rr1*(-Coeffs.E.eta_r11+Coeffs.E.eta_r31/Coeffs.V.R_v1-
Coeffs.E.Lambdar1/Coeffs.V.R_v1));
% Tw1
a11_12 = -Coeffs.E.K1/Coeffs.E.C1*(1/Coeffs.E.Ra1 + 1/Coeffs.E.Rr1);
% Tai
a11_13 = Coeffs.E.K1/(Coeffs.E.C1*Coeffs.E.Ra1);
% Pc
a10_11 = 1/(Coeffs.E.C1*Coeffs.V.R_v1)*(1 +
Coeffs.P.K_q3*Coeffs.E.eta_r31/Coeffs.P.R_q1 -
Coeffs.E.K1/Coeffs.E.Rr1*(Coeffs.E.eta_r31 + Coeffs.E.Lambdar1));
% Pq
a10_12 = 1/(Coeffs.E.C1*Coeffs.P.R_q1);
% avi
b11_11 = Coeffs.V.K_v1/(Coeffs.E.C1*Coeffs.V.R_v1)*(1 +
Coeffs.P.K_q3*Coeffs.E.eta_r31/Coeffs.P.R_q1 -
Coeffs.E.K1/Coeffs.E.Rr1*(Coeffs.E.Lambdar1 + Coeffs.E.eta_r31));
% wfi
b11_12 = 1/Coeffs.E.C1*(Coeffs.P.K_q2/Coeffs.P.R_q1 -
Coeffs.P.K_q3*Coeffs.E.eta_r21/Coeffs.P.R_q1 +
Coeffs.E.K1*Coeffs.E.Kf1/Coeffs.E.Ra1 +
Coeffs.E.K1*Coeffs.E.eta_r21/Coeffs.E.Rr1);
% wk
b10_11 = Coeffs.P.K_q1/(Coeffs.E.C1*Coeffs.P.R_q1);
%% Twi dot
```

```
% Pi
a11_21 = 1/(Coeffs.E.Cw1*Coeffs.E.Rr1)*(Coeffs.E.eta_r11 + (Coeffs.E.Lambdar1
+ Coeffs.E.eta_r31)/Coeffs.V.R_v1);
% Twi
a11_22 = -1/Coeffs.E.Cw1*(1/Coeffs.E.Ra1 + 1/Coeffs.E.Rr1);
% Tai
a11_23 = 1/(Coeffs.E.Cw1*Coeffs.E.Ra1);
% Pc
a10_21 = -(Coeffs.E.Lambdar1 +
Coeffs.E.eta_r31)/(Coeffs.E.Cw1*Coeffs.E.Rr1*Coeffs.V.R_v1);
% avi
b11_21 = -Coeffs.V.K_v1*(Coeffs.E.Lambdar1 +
Coeffs.E.eta_r31)/(Coeffs.E.Cw1*Coeffs.E.Rr1*Coeffs.V.R_v1);
% wfi
b11_22 = 1/Coeffs.E.Cw1*(Coeffs.E.Kf1/Coeffs.E.Ra1 +
Coeffs.E.eta_r21/Coeffs.E.Rr1);
%% Tai dot
% Twi
a11_32 = 1/(Coeffs.S.Ca1*Coeffs.E.Ra1);
% Tai
a11_33 = -1/(Coeffs.S.Ca1*Coeffs.E.Ra1);
% wfi
b11_32 = -Coeffs.E.Kf1/(Coeffs.S.Ca1*Coeffs.E.Ra1);
% QLi
v11_3 = 1/Coeffs.S.Ca1;
%% Trq dot
% Pi
a01_31 =
1/Coeffs.J.Crq*(1/Coeffs.P.mu_m1*((1+Coeffs.P.mu_m2*Coeffs.P.K_q3/Coeffs.P.R_
q1)*(Coeffs.E.eta_r11+Coeffs.E.eta_r31/Coeffs.V.R_v1)+Coeffs.P.mu_m2/Coeffs.P
.R_q1)-
Coeffs.J.Krq*(1/Coeffs.P.R_q1+Coeffs.P.K_q3/Coeffs.P.R_q1*(Coeffs.E.eta_r11+C
oeffs.E.eta_r31/Coeffs.V.R_v1)));
% Pc
a00_31 =
1/Coeffs.J.Crq*(n*(1/Coeffs.P.mu_m1*((1+Coeffs.P.mu_m2*Coeffs.P.K_q3/Coeffs.P
.R_q1)*(-
Coeffs.E.eta_r31/Coeffs.V.R_v1)+Coeffs.P.mu_m3*(Coeffs.K.gamma_k1*Coeffs.C.et
a_rc-Coeffs.K.gamma_k5-
Coeffs.K.alpha_k1*Coeffs.C.eta_rc)+Coeffs.P.mu_m4*Coeffs.K.B_k2)+Coeffs.J.Krq
*Coeffs.P.K_q3*Coeffs.E.eta_r31/(Coeffs.P.R_q1*Coeffs.V.R_v1))-
(Coeffs.K.gamma_k1*Coeffs.C.eta_rc-Coeffs.K.gamma_k5-
Coeffs.K.alpha_k1*Coeffs.C.eta_rc)-Coeffs.J.Krq*Coeffs.K.B_k2);
% Pq
a00_32 = 1/Coeffs.J.Crq*(n*(1/Coeffs.P.mu_m1*(-
Coeffs.P.mu_m2/Coeffs.P.R_q1+Coeffs.P.mu_m3*Coeffs.K.gamma_k4-
Coeffs.P.mu_m4*Coeffs.K.B_k1)+Coeffs.J.Krq/Coeffs.P.R_q1)-
Coeffs.K.gamma_k4+Coeffs.J.Krq*Coeffs.K.B_k1);
% Trq
a00_33 = 1/Coeffs.J.Crq*(n/Coeffs.P.mu_m1*(-1-
Coeffs.P.mu_m3*(Coeffs.K.gamma_k2-
Coeffs.K.alpha_k2)+Coeffs.P.mu_m4*Coeffs.K.B_k4)+Coeffs.K.gamma_k2-
Coeffs.K.alpha_k2-Coeffs.J.Krq*Coeffs.K.B_k4);
% avi
```

```
b01_31 =
1/Coeffs.J.Crq*(1/Coeffs.P.mu_m1*((1+Coeffs.P.mu_m2*Coeffs.P.K_q3/Coeffs.P.R_
q1)*-
Coeffs.E.eta_r31*Coeffs.V.K_v1/Coeffs.V.R_v1)+Coeffs.J.Krq*Coeffs.P.K_q3*Coef
fs.E.eta_r31*Coeffs.V.K_v1/(Coeffs.P.R_q1*Coeffs.V.R_v1));
% wfi
b01_32 =
1/Coeffs.J.Crq*(1/Coeffs.P.mu_m1*((1+Coeffs.P.mu_m2*Coeffs.P.K_q3/Coeffs.P.R_
q1)*Coeffs.E.eta_r21-Coeffs.P.mu_m2*Coeffs.P.K_q2/Coeffs.P.R_q1-
Coeffs.P.mu_m5)-Coeffs.J.Krq/Coeffs.P.R_q1*(Coeffs.P.K_q3*Coeffs.E.eta_r21-
Coeffs.P.K_q2));
% wk
b00_31 = 1/Coeffs.J.Crq*(n*(1/Coeffs.P.mu_m1*(-
Coeffs.P.mu_m2*Coeffs.P.K_q1/Coeffs.P.R_q1+Coeffs.P.mu_m3*(Coeffs.K.gamma_k3-
Coeffs.K.alpha_k3)-
Coeffs.P.mu_m4*Coeffs.K.B_k3)+Coeffs.J.Krq*Coeffs.P.K_q1/Coeffs.P.R_q1)+Coeff
s.K.alpha_k3-Coeffs.K.gamma_k3+Coeffs.J.Krq*Coeffs.K.B_k3);
%% Generate State-space SubMatrices
a11 = [a11_11 a11_12 a11_13;...
       a11_21 a11_22 a11_23;...
          0   a11_32 a11_33];
a10 = [a10_11 a10_12 0;...
       a10_21    0   0;...
          0      0   0];
a01 = [a01_11 a01_12 0;...
       a01_21 0 0;...
       a01_31 0 0];
a00 = [a00_11 a00_12 a00_13;...
       a00_21 a00_22 a00_23  ;...
       a00_31 a00_32 a00_33];
b11 = [b11_11 b11_12;...
       b11_21 b11_22;...
          0   b11_32];
b10 = [b10_11 0;...
          0   0;...
          0   0];
b01 = [b01_11    0  ;...
       b01_21 b01_22;...
       b01_31 b01_32];
b00 = [b00_11 b00_12;...
       b00_21    0  ;...
       b00_31    0];
v11 = [0;...
       0;...
       v11_3];
v00 = [v00_1;...
         0  ;...
         0];
%% Generate State-space Matrices
A11 = blkdiag(a11,a11,a11,a11,a11);
A10 = [a10;a10;a10;a10;a10];
A01 = [a01 a01 a01 a01 a01];
A00 = a00;
A = [A11 A10; A01 A00];
B11 = blkdiag(b11,b11,b11,b11,b11);
```

```
B10 = [b10;b10;b10;b10;b10];
B01 = [b01 b01 b01 b01 b01];
B00 = b00;
B = [B11 B10; B01 B00];
V11 = blkdiag(v11,v11,v11,v11,v11);
V10 = zeros(3*n,1);
V01 = zeros(3,n);
V00 = v00;
V = [V11 V10; V01 V00];
C = zeros(18);
C(1:17,1:17) = eye(17);
C(18,18) = 1;
C(18,17) = -0.092;
%% Nominal actuator and state inputs
EEV0    = 18.25;
eFan0   = 1190.1;
Comp0   = 2425;
cFan0   = 740.1767;
Pc0     = 1669.1;
Pq0     = 321.9345;
% Actuator costs (actuators for S1,S2,S3,S4,S5 have the same costs)
au_EEV  = 0;
bu_EEV  = 2*au_EEV*EEV0 + 0;
au_eFan = 5.1e-5;
bu_eFan = 2*au_eFan*eFan0 + (-1.18e-2);
au_Comp = 2.9204e-5;
bu_Comp = 2*au_Comp*Comp0 + (2.1364e-1);
au_cFan = 1.5*1.4e-4;
bu_cFan = 1.5*(2*au_cFan*cFan0 + (3.15e-2));
ax_Pc   = 1.6656e-3;
bx_Pc   = 2*ax_Pc*Pc0 + (-2.9835e0);
ax_Pq   = -1.7646e-3;
bx_Pq   = 2*ax_Pq*Pq0 + (1.4343e0);
%% Common Information
% Sample time
Sys.dt      = 10;
% Control Horizon
Sys.Nu      = 15;
% Prediction Horizon
Sys.Np      = 60;
% Objectives vs delta U
Sys.gamma_a     = 0.1;
% Performance vs efficiency
Sys.gamma_b     = 0.1;
% Error vs integral
Sys.gamma_c     = 0.1;
% Ta tracking performance objective
q_Ta        = 1e2;
% SH tracking performance objective
q_SH        = 1e-2;
% Acutator EEV
q_EEV        = 1e2;
% Actuator eFan
q_eFan       = 1e-3;
% Acutator Comp
```

```
q_Comp        = 1e-1;
% Actuator cFan
q_cFan        = 1e-2;
% Constraints
Min_EEV = 0-EEV0;
Min_eFan = 0-eFan0;
Min_Comp = 0-Comp0;
Min_cFan = 0-cFan0;
Max_EEV = 100-EEV0;
Max_eFan = 1500-eFan0;
Max_Comp = 3500-Comp0;
Max_cFan = 1600-cFan0;
%% Centralized System Information
% Model
Sys.A   = [A11 A10; A01 A00];
Sys.B   = [B11 B10; B01 B00];
Sys.V   = [V11 V10; V01 V00];
Sys.C   = C;
% Number of States
Sys.ns      = 18;
% Number of Inputs
Sys.nu      = 12;
% Number of Disturbances
Sys.nd      = 6;
% Number of References
Sys.nr      = 6;
% States with desired values
Sys.M       = blkdiag([0 0 1],[0 0 1],[0 0 1],[0 0 1],[0 0 1],[0 0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Ta q_Ta q_Ta q_Ta q_Ta q_SH];
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Ta q_Ta q_Ta q_Ta q_Ta
q_SH];
Sys.q2 = [q_EEV q_eFan q_EEV q_eFan q_EEV q_eFan q_EEV q_eFan q_EEV q_eFan
q_Comp q_cFan];
Sys.qa = [au_EEV au_eFan au_EEV au_eFan au_EEV au_eFan au_EEV au_eFan au_EEV
au_eFan au_Comp au_cFan];
Sys.qb1 = [bu_EEV bu_eFan bu_EEV bu_eFan bu_EEV bu_eFan bu_EEV bu_eFan bu_EEV
bu_eFan bu_Comp bu_cFan];
Sys.qax = [zeros(1,15) ax_Pc ax_Pq 0];
Sys.qbx1 = [zeros(1,15) bx_Pc bx_Pq 0];
% Min and Max inputs
Sys.Mins =
[Min_EEV;Min_eFan;Min_EEV;Min_eFan;Min_EEV;Min_eFan;Min_EEV;Min_eFan;Min_EEV;
Min_eFan;Min_Comp;Min_cFan];
Sys.Maxs =
[Max_EEV;Max_eFan;Max_EEV;Max_eFan;Max_EEV;Max_eFan;Max_EEV;Max_eFan;Max_EEV;
Max_eFan;Max_Comp;Max_cFan];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
T_bar = Outputs.T_bar;
S_bar = Outputs.S_bar;
R_bar = Outputs.R_bar;
P_bar = Outputs.P_bar;
N_bar = Outputs.N_bar;
n_bar = Outputs.n_bar;
```

```
Q1 = Outputs.Q1;
Q2 = Outputs.Q2;
Qa = Outputs.Qa;
qb = Outputs.qb;
Qax = Outputs.Qax;
qbx = Outputs.qbx;
U_min = Outputs.U_min;
U_max = Outputs.U_max;
%% BAS System Information
% Model
Sys.A   = [a11 a10; 5*a01 a00];
Sys.B   = [b11 b10; 5*b01 b00];
Sys.V   = [v11 zeros(3,1);zeros(3,1) v00];
Sys.C   = [eye(5) zeros(5,1); zeros(1,4) -0.092 1];
% Number of States
Sys.ns      = 6;
% Number of Inputs
Sys.nu      = 4;
% Number of Disturbances
Sys.nd      = 2;
% Number of References
Sys.nr      = 2;
% States with desired values
Sys.M       = blkdiag([0 0 1],[0 0 1]);
% Weightings
Sys.qe = Sys.gamma_a*Sys.gamma_b*Sys.gamma_c*[q_Ta q_SH];
Sys.qi = Sys.gamma_a*Sys.gamma_b*(1-Sys.gamma_c)*[q_Ta q_SH];
Sys.q2 = [q_EEV q_eFan q_Comp q_cFan];
Sys.qa = [au_EEV au_eFan au_Comp au_cFan];
Sys.qb1 = [bu_EEV bu_eFan bu_Comp bu_cFan];
Sys.qax = [zeros(1,3) ax_Pc ax_Pq 0];
Sys.qbx1 = [zeros(1,3) bx_Pc bx_Pq 0];
% Min and Max inputs
Sys.Mins = [Min_EEV;Min_eFan;Min_Comp;Min_cFan];
Sys.Maxs = [Max_EEV;Max_eFan;Max_Comp;Max_cFan];
% Formulated MPC Variables
[Outputs] = MPC_Formulation(Sys);
BAS1_T_bar = Outputs.T_bar;
BAS1_S_bar = Outputs.S_bar;
BAS1_R_bar = Outputs.R_bar;
BAS1_P_bar = Outputs.P_bar;
BAS1_N_bar = Outputs.N_bar;
BAS1_n_bar = Outputs.n_bar;
BAS1_Q1 = Outputs.Q1;
BAS1_Q2 = Outputs.Q2;
BAS1_Qa = Outputs.Qa;
BAS1_qb = Outputs.qb;
BAS1_Qax = Outputs.Qax;
BAS1_qbx = Outputs.qbx;
BAS1_U_min = Outputs.U_min;
BAS1_U_max = Outputs.U_max;
%% ESC Parameters
a0 = 20;
ug0 = 0;
delta = 1;
```

```
omega = 2*pi/400;
phi = 50*pi/180;
k_prime = 5e0;
omega_l_prime = 0.1;
omega_h_prime = 10;
k = omega*delta*k_prime;
omega_l = omega*delta*omega_l_prime;
omega_h = omega*delta*omega_h_prime;
```

B.2 MPC Formulation

The following MATLAB code contains the function *MPC_Formulation.m* which is called in the code above to generate the variables used for the MPC controllers.

```
function [Outputs] = MPC_Formulation(Sys);
A = Sys.A;
B = Sys.B;
V = Sys.V;
C = Sys.C;
ns = Sys.ns;
nu = Sys.nu;
nd = Sys.nd;
nr = Sys.nr;
M = Sys.M;
dt = Sys.dt;
Nu = Sys.Nu;
Np = Sys.Np;
gamma_a = Sys.gamma_a;
gamma_b = Sys.gamma_b;
gamma_c = Sys.gamma_c;
qe = Sys.qe;
qi = Sys.qi;
q2 = Sys.q2;
qa = Sys.qa;
qb1 = Sys.qb1;
qax = Sys.qax;
qbx1 = Sys.qbx1;
Mins = Sys.Mins;
Maxs = Sys.Maxs;
%% Augment system into error system S_hat
A_hat   = C*A/C;
B_hat   = C*B;
V_hat   = [C*V C*A/C*pinv(M)];
C_hat   = eye(ns);
D_hat   = zeros(ns,nu);
%% Discretize System to get S_d
% Centralized System
sys_c   = ss(A_hat,[B_hat V_hat],C_hat,[D_hat zeros(size(V_hat))]);
sys_d_prime = c2d(sys_c,dt);
A_d     = sys_d_prime.a;
B_d     = sys_d_prime.b(:,1:nu);
V_d     = sys_d_prime.b(:,nu+1:nu+nd+nr);
```

```
C_d     = sys_d_prime.c;
D_d     = sys_d_prime.d(:,1:nu);
%% Add Integrator and x_u states to get S_d_bar
A_bar   = [A_d          zeros(ns,nr)    B_d;...
           M*dt         eye(nr)         zeros(nr,nu);...
           zeros(nu,ns) zeros(nu,nr)    eye(nu)];
B_bar   = [B_d; zeros(nr,nu); eye(nu)];
V_bar   = [V_d; zeros(nr,nd+nr); zeros(nu,nd+nr)];
C_bar   = [M            zeros(nr)   zeros(nr,nu);...
           zeros(nr,ns) eye(nr)     zeros(nr,nu)];
%% Generate lifted system matrices
% T_bar
T_bar = zeros((ns+nr+nu)*Np,ns+nr+nu);
for i = 1:Np
    T_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),:) = A_bar^i;
end
% S_bar
S_bar = zeros((ns+nr+nu)*Np,nu*Nu);
for i = 1:Np
    for j = 1:Nu
        if i-j < 0
            S_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),(j-1)*nu+1:j*nu) =
zeros((ns+nr+nu),nu);
        elseif i-j == 0
            S_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),(j-1)*nu+1:j*nu) = B_bar;
        elseif i-j > 0
            S_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),(j-1)*nu+1:j*nu) =
A_bar^(i-j)*B_bar;
        end
    end
end
% R_bar
R_prev = zeros(size(V_bar));
R_bar = zeros((ns+nr+nu)*Np,nd+nr);
for i = 1:Np
    R_bar((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu),:) = A_bar^(i-1)*V_bar+R_prev;
    R_prev = A_bar^(i-1)*V_bar+R_prev;
end
% P_bar
P_bar = zeros(2*nr*Np,(ns+nr+nu)*Np);
for i = 1:Np
    P_bar((i-1)*2*nr+1:i*2*nr,(i-1)*(ns+nr+nu)+1:i*(ns+nr+nu)) = C_bar;
end
% N_bar
N_bar = zeros(nu*Np,nu*Nu);
for i = 1:Np
    for j = 1:Nu
        if i >= j
            N_bar((i-1)*nu+1:i*nu,(j-1)*nu+1:j*nu) = eye(nu);
        end
    end
end
% n_bar
n_bar = zeros(nu*Np,nu);
n_bar(1:nu,1:nu) = eye(nu);
```

```
%% Weighting Matrices
% Q1
q1 = [qe qi];
for i = 1:Np
    q1_all((i-1)*2*nr+1:i*2*nr) = q1;
end
Q1 = diag(q1_all);
% Q2
for i = 1:Nu
    q2_all((i-1)*nu+1:i*nu) = q2;
end
Q2 = (1-gamma_a)*diag(q2_all);
% Qa
for i = 1:Np
    qa_all((i-1)*nu+1:i*nu) = qa;
end
Qa = gamma_a*(1-gamma_b)*diag(qa_all);
% qb
for i = 1:Np
    qb_all((i-1)*nu+1:i*nu) = qb1;
end
qb = gamma_a*(1-gamma_b)*qb_all;
% Qax
for i = 1:Np
    qax_all((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu)) = [qax zeros(1,nr+nu)];
end
Qax = gamma_a*(1-gamma_b)*diag(qax_all);
% qbx
for i = 1:Np
    qbx_all((i-1)*(ns+nr+nu)+1:i*(ns+nr+nu)) = [qbx1 zeros(1,nr+nu)];
end
qbx = gamma_a*(1-gamma_b)*qbx_all;
%% Constraints
% Lifted Constraints
for i = 1:Np
    U_min((i-1)*nu+1:i*nu,1) = Mins;
end
for i = 1:Np
    U_max((i-1)*nu+1:i*nu,1) = Maxs;
end
%% Outputs
Outputs.T_bar = T_bar;
Outputs.S_bar = S_bar;
Outputs.R_bar = R_bar;
Outputs.P_bar = P_bar;
Outputs.N_bar = N_bar;
Outputs.n_bar = n_bar;
Outputs.Q1 = Q1;
Outputs.Q2 = Q2;
Outputs.Qa = Qa;
Outputs.qb = qb;
Outputs.Qax = Qax;
Outputs.qbx = qbx;
Outputs.U_min = U_min;
Outputs.U_max = U_max;
```

B.3 MPC Code

The following MATLAB code contains the function *MPC.m* which uses the MATLAB function *quadprog* to solve the MPC optimization problem at every same time of the simulation.

```
function [Controls,DV] = MPC(T_bar,S_bar,R_bar,P_bar,N_bar,n_bar,Q1,Q2,Qa,qb,
      Qax,qbx,U_min,U_max,X,R0,V0,U0,Dist)
eml.extrinsic('quadprog','optimset')
x_bar0 = [X;U0];
d = [Dist;R0];
a = [-N_bar;N_bar];
b = [-(U_min-n_bar*U0);(U_max-n_bar*U0)];
H = zeros(size(Q2));
F = zeros(size(Q2,1),1);
H = S_bar'*P_bar'*Q1*P_bar*S_bar + Q2 + N_bar'*Qa*N_bar + S_bar'*Qax*S_bar;
H = 2*(H+H')/2;
F = 2*S_bar'*P_bar'*Q1*P_bar*T_bar*x_bar0 + 2*S_bar'*P_bar'*Q1*P_bar*R_bar*d
    +2*N_bar'*Qa*n_bar*U0 + N_bar'*qb'+...
    2*S_bar'*Qax*T_bar*x_bar0 + 2*S_bar'*Qax*R_bar*d + S_bar'*qbx';
options = optimset('MaxIter',50,'algorithm','active-set');
DV = zeros(size(Q2,1),1);
DV = quadprog(H,F,a,b,[],[],[],[],V0,options);
Controls = DV(1:size(n_bar,2));
```