OPTIMAL ENERGY USE IN MOBILE APPLICATIONS WITH STORAGE

BY

TIM OLIVER DEPPEN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Professor Andrew Alleyne, Chair
Professor Geir Dullerud
Professor Daniel Liberzon
Associate Professor Prashant Mehta

Abstract

With growing demands and dwindling resources, the need for energy efficiency is being felt in all sectors. The transportation sector is one of the largest consumers of energy and to reduce fuel consumption and greenhouse gas emissions, hybrid mobile power systems are seeing increased use. A hybrid mobile power system is any vehicle that includes a power source and a means of storing that power. These vehicles offer an opportunity for improved efficiency by partially decoupling power generation from demand, enabling more efficient operation. This decoupling is achieved via energy storage which offers new opportunities for how energy is utilized. To realize the potential of hybrid architectures, an energy management strategy (EMS) is needed to regulate the generation, distribution, and storage of energy. Hybrid vehicles span wide power and weight scales from small passenger vehicles to large delivery trucks and the energy storage mechanisms come in many domains including mechanical, thermal, and electrical. Therefore, if one is to enable effective wide spread use of hybrid vehicles, a method for designing EMS's which is effective across applications and energy domains is needed.

In this work a procedure for design of EMS's is given, which is intended to be generalizable to the entire class of hybrid mobile power systems. This procedure begins by decomposing the vehicle operation into modes characterized by which power sources are needed. Then convex quadratic objective functions are designed for each mode which attempt to maximize operational efficiency while meeting a performance goal. Finally, a supervisory logic is used to regulate switching between modes. The model predictive control (MPC) framework is used to setup the optimization problem within each mode as a receding horizon optimal controller which can be implemented in real-time. The proposed method facilitates online implementation because it constrains the optimization problem to be convex and quadratic.

Furthermore, MPC allows flexibility in how much knowledge one assumes about the future, enabling this approach to be applied equally well to highly uncertain applications, like passenger vehicles, and well known systems, like city busses.

The generalizability of the proposed method is tested through application in two different hybrid vehicles; a series hydraulic hybrid vehicle (SHHV) and a refrigerated delivery truck with thermal storage. The SHHV is a passenger vehicle which uses a hydrostatic transmission with a high pressure gas charged accumulator for energy storage. The goal of this system is to meet the driver's speed demand while maximizing operational efficiency. This case study includes experimental validation of the EMS performance using a hardware-in-the-loop system. The refrigerated delivery truck uses a vapor compression cycle system that has been augmented with thermal storage to maintain a desired box temperature while maximizing operational efficiency. These case studies employ different architectures, different energy domains, and different degrees of knowledge of the system and duty cycle. However, the proposed EMS design method is able to yield energy savings for both.

To the next generation of dreamers,

have faith in your own power

Acknowledgements

By their very nature these acknowledgements are incomplete, and cannot sufficiently express my appreciation. Nevertheless, I have been fortunate to have met many amazing and supportive individuals throughout my academic journey and would like to recognize some of them here. My Advisor, Prof. Andrew Alleyne, has been a guiding hand in my graduate studies. Not only has he helped me to mature as scientist, and learn the value of always asking questions, but he has also been a strong supporter of my goal to have a societal impact. Through his mentorship I have further developed my research and leadership skills and have been given many opportunities to leverage both.

I would also like to thank my family for their continuous and firm support. They gave me the foundation upon which I have built my career. My parents have always believed in my ability to succeed and been eager to aid me in any way possible. Their faith was critical to me reaching this milestone. My brother, Raymond, has always been and will continue to be a major influence in my life. He has never expected less of me than my very best. He has also been a strong advocate for what's in my best interest and, despite my stubbornness, has led me to some of my most fortunate opportunities.

During my time in the Alleyne research group, I have had the privilege to work with many wonderful individuals and form strong friendships. Kira Barton, Sandipan Mishra, Doug Bristow, and Dave Hoelzle are all alumnus of the lab who helped me get started in grad school. During my first years, they were very welcoming, friendly, and quick to give time and advice. Since joining the lab, I have made many friends: Bin, Neera, Erick, Nanjun, Yangmin, and Vikas. Each of you has helped to shape my experience in grad school and I hope to maintain our friendship moving forward. Bin has been a reliable friend these past years and his advice has

helped me to make many critical decisions. I have always looked up to Neera as an exemplary lab mate and I have relied on her support numerous times. Erick has helped me to see many issues from a different perspective and, despite different views and ideologies, we have always had open and respectful discussions. I wish our friendship was the model for how all critical issues are debated. Nanjun has been a very generous friend and has helped me to seize many opportunities I would have never considered. Yangmin has always been a very kind and helpful presence in the lab. Vikas and I joined the lab at the same time and through our mutual support I was able to make it through many tough times. I have also had the opportunity to welcome many new students to the lab. Joey, Justin, Matt, Megan, and Sara; I have enjoyed our time together and hope that I have been able to impart to you some small measure of wisdom. Over these past five and a half years, the Alleyne research group has become my home and I will always cherish my time there.

In addition to the Alleyne research group, I also conducted much of my research through the Center for Compact and Efficient Fluid Power (CCEFP). Within the CCEFP I also found a network of great people. Dr. Kim Stelson, Alyssa Burger, and Don Haney, who all work tirelessly to run the CCEFP, have always been welcoming and supportive. Jonathan Meyer, a student at the University of Minnesota Twin-Cities, has been my collaborator in the CCEFP from the beginning. Jon and I have shared many tight deadlines, written many papers together, and presented more posters than I care to count. Jon's assistance and friendship has been a valuable addition to my graduate career. I would also like to acknowledge Alex Shorter who was a senior graduate student from UIUC within the CCEFP. Alex helped me to become integrated with the CCEFP and showed me opportunities to become a student leader in that organization.

Finally, I would not have accomplished this work without several close friends who contributed much to my success and happiness. Harrison Parker is the most earnest and loyal person I've ever met. His unwavering confidence in me often exceeds my own and that faith has helped me to persevere. Simon Ellis has helped me to work through many difficult decisions. I have relied on his advice and support to navigate choices relating to school, research, and personal life. My cousin, Frank Fiore, and I have shared many of life's experiences together. He has helped me to see past my immediate concerns and appreciate the bigger picture. Last but not least, my closest friend Nishana Ismail. She has opened my eyes to the world, been a steadfast

partner, and given me the confidence to realize my dreams. Without her support I wouldn't be the person I am today. I count myself blessed to have known each of you.

Table of Contents

List of Tables

List of Figures

Nomenclature

α	Convection coefficient	$W.m^{-2}.K^{-1}$
A	Area	m^2
β	Bulk Modulus	MPa
b	Viscous friction coefficient	$N.m.(rad.s^{-1})^{-1}$
C_{drag}	Air drag coefficient	--
Cap	Accumulator capacity	cm^3
C_P	Specific heat	$kJ.(kg.K)^{-1}$
δ	Difference with respect to operating point	--
D	Displacement	$cm^3.rad^{-1}$
η	Number of times an event occurred	--
γ	Cooling capacity	W
H	Hessian matrix	--
h	Specific enthalpy	$kJ.kg^{-1}$
I	Inertia	$kg.m^2$
i	Discrete time index	--
κ	Friction coefficient	$N.m.(rad.s^{-1})^{-2}$
k	Specific heat ratio	--
k_f	Fuel consumption coefficient	$kg.rad^{-1}$
K_P	Pump flow gain	$cm^3.rad^{-2}$
k_{TH}	Thermal conductivity	$W.(m.K)^{-1}$
λ, ζ, ς	Objective function weighting	--
$\dot{m}$	Mass flow rate	$kg.s^{-1}$

N	Prediction horizon length	--
N_s	Number of nodes in thermal storage model	--
ω	Angular velocity	$rad.s^{-1}$
P	Pressure	MPa
p	Probability of an event occurring	--
ψ	Hose leakage coefficient	$cm^3.(s.MPa)^{-1}$
Q	Volumetric flow rate	$cm^3.s^{-1}$
ρ	Density	$kg.m^{-3}$
r	Radius	m
ref	Reference	--
τ	Torque	N.m
θ	Throttle command	degrees
T	Temperature	C
t	Time	s
U	Vector of control inputs	--
u	Control input	--
u_{air}	Air mass flow rate across storage	$kg.s^{-1}$
u_e	Engine torque	N.m
u_{eev}	Electronic expansion valve opening	--
u_{IV}	Ideal valve flow split command	--
u_ω	Compressor speed command	RPM
u_p	Pump swashplate angle	rad
u_v	Valve command	V
V	Volume	cm^3
x	A state	--
Y	Vector of outputs	--
y	An output	--
z	Discrete integral of tracking error	--

Subscripts

comp	Compressor inlet
d	Downstream of valve
des	Desired value
e	Engine
ev	Evaporator
eev	Electronic expansion valve outlet
inner	Inner nodal boundary
L	Load
m	Hydraulic Motor
max	Maximum value
outer	Outer nodal boundary
pr	Precharge
PCM	Phase Change Material
s	Thermal storage
u	Upstream of valve
w	Wheel
wall,a	Outer storage wall interface with air
wall,r	Inner storage wall interface with refrigerant
wall,si	Inner storage wall interface with PCM
wall,so	Outer storage wall interface with PCM

Abbreviations

AEVPS	Augmented Earthmoving Vehicle Powertrain Simulator
DDP	Deterministic Dynamic Programming
EEV	Electronic Expansion Valve
EMS	Energy Management Strategy
MPC	Model Predictive Control
PCM	Phase Change Material
PRV	Pressure Relief Valve
PVCC	Parallel Vapor Compression Cycle
RMS	Root Mean Square
SDP	Stochastic Dynamic Programming
SHHV	Series Hydraulic Hybrid Vehicle
UDDS	Urban Dynamometer Driving Schedule
VCC	Vapor Compression Cycle

Chapter 1 Introduction

This thesis examines energy optimization in the context of mobile systems that include energy storage. This is a very broad class of systems that spans large power and weight scales. Furthermore, these systems can be used to meet a variety of demands, or duty cycles, from acceleration/braking demands to maintaining temperature. These hybrid systems have seen increased use as their excess degrees of freedom make it possible to meet a performance demand while improving operating efficiency. The most prevalent example of such systems is vehicles incorporating electric storage into the powertrain. In Section 1.1 a general discussion of this class of systems is provided, including different mechanisms for energy storage and different architectures for integrating storage. To realize the potential energy savings of these systems, an energy management strategy (EMS) is needed to regulate the excess degrees of freedom. Section 1.2 reviews different methods of deriving real-time EMS's and their limitations. In this work, an approach for designing a real-time implementable EMS is given. The proposed methodology is intended to be broad enough to accommodate the entire class of hybridized vehicles. Therefore, it must allow for varying levels of duty cycle preview and be executable quickly with limited processing power. This is accomplished through a combination of decomposing system operation into operating modes and then using a system wide component analysis within each mode to characterize the overall operating efficiency as a summation of quadratic cost terms. The optimization problems within each mode are then solved using the model predictive control (MPC) framework. The flexibility and performance of this methodology is demonstrated through the two case studies: a hydraulic hybrid vehicle and a refrigerated delivery truck. The thermal hybrid is unique from those introduced in Section 1.1 since additional work is required to extract potential from the storage unit and storage is integrated using different system architectures. This hybrid is introduced in Section 1.3. The two case studies represent different energy domains, different architectures, and are evaluated with different constraints on system knowledge and preview of the duty cycle. From these two

studies, the generalizability of the EMS design method will be demonstrated. A complete discussion of the proposed EMS design method is presented in Chapter 2.

1.1 Hybrid Mobile Power Systems

The transportation sector is one of the largest consumers of energy, accounting for 28% of total US energy consumption in 2011. Furthermore, 93% of this consumption is fueled by petroleum and the demand is projected to grow in the coming decades [1]. The need to stem this consumption and reduce greenhouse gas emissions has stimulated the development of hybrid mobile power systems. A hybrid mobile power system is any vehicle that includes a power source and a means of storing that power. These vehicles offer an opportunity for improved efficiency by partially decoupling power generation from demand, allowing for more efficient operation. Furthermore, in some applications energy which is typically lost to the environment can be regenerated. For example a regenerative braking system can capture energy in the vehicle's momentum which is otherwise dissipated via mechanical brakes [2]. Hybrid mobile power systems come in many scales and domains: from electric hybrid passenger vehicles to large hydraulic hybrid delivery trucks [3], [4]. Despite the wide variance in application, all hybrid mobile power systems include a storage mechanism and there are some common architectures for integrating said storage into the vehicle.

1.1.1 Storage Mechanisms

Energy within a vehicle is typical transferred in one of two domains: mechanical, and electrical. As such, there are storage mechanisms which can be employed within each of these domains. In the mechanical domain energy can be stored using a flywheel, material strain, or pressurized gas [5–11]. In the electrical domain batteries are common storage mechanisms due to their relatively high energy density [6], [8], [12–14]. However, ultracapacitors and fuel cells are receiving growing interest [3], [13], [15], [16]. Fuel cells, like fossil fuels, release energy through a chemical reaction so they could also be classified as chemical energy storage [8], [13]. However, they are used to produce electricity. Examples of mechanical and electrical storage mechanisms are shown in Fig. 1.1.

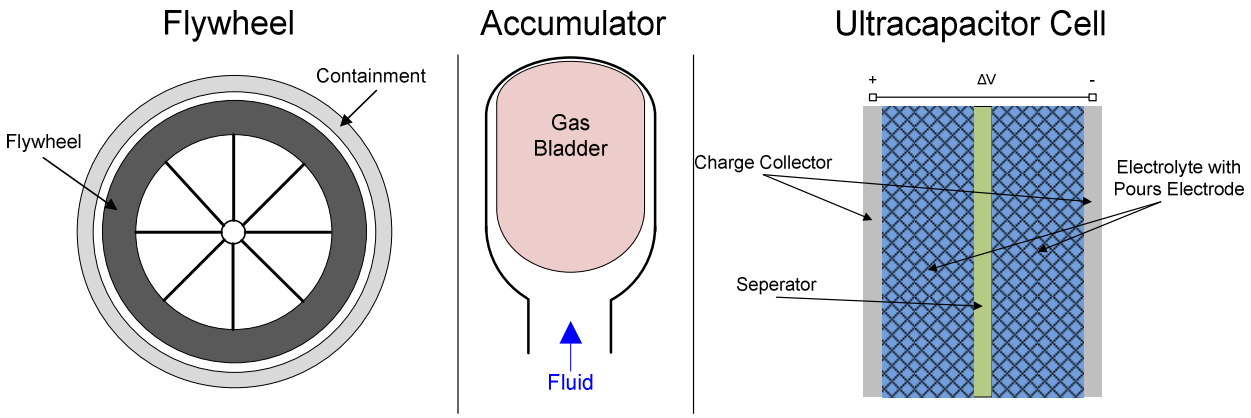

Figure 1.1: Energy storage mechanisms

The high speed flywheel stores energy as rotational kinetic energy and therefore is designed to have low inertia and operate on the order of 10,000 – 100,000 RPM [6], [7]. It has high transmission efficiency and high power density but low energy density compared to electric batteries [6]. Therefore, it has been studied in large vehicle applications which require large acceleration and braking power [7]. The main drawback of this storage technology is the air drag losses and bearing friction which can quickly dissipate the stored energy. To overcome these losses a vacuum can be induced or low friction gases can be pumped into the flywheel containment vessel [7]. An alternate method of storing mechanical energy is via pressurized fluid. The hydraulic hybrid has received growing attention from the academic and industry communities. This vehicle uses a high pressure accumulator for energy storage because fluid power has a high power density and the accumulator can be fully charged and discharged safely for many cycles without loss in performance [17]. These characteristics make fluid power particularly attractive for urban driving applications where there are frequent starts and stops [11], [18–22]. To improve the energy density of accumulators and eliminate the thermal losses associated with maintaining a pressurized gas, researchers are looking into energy storage via material strain [9].

In the electric domain batteries have seen widespread use in large and small vehicle applications [8], [12], [23]. These devices have higher energy density than flywheels and gas charged accumulators but cannot tolerate fast charge/discharge rates and have a limited band of available charge. Furthermore, they have limited charge/discharge cycles [8]. To overcome the life cycle and power density limitations of batteries, ultracapacitors have been developed. These

devices have similar properties to gas charged accumulators in that they can accommodate fast charge/discharge rates. However, they sacrifice the energy density of traditional batteries [13].

1.1.2 Architectures

Most hybrid vehicles involve hybridization of the vehicle powertrain. In this case, the prime mover is an engine and the storage unit is integrated into the vehicle powertrain. There are three common architectures for integrating energy storage into a vehicle powertrain and they are differentiated by what paths are available for power to be delivered from the engine to the load. These three architectures are: parallel, series, and power-split. The parallel configuration, or power assist, uses an energy storage mechanism in parallel with a mechanical power transmission path to store, disperse, and reclaim energy. One of the advantages of this architecture is that the highly efficient mechanical transmission between the prime mover and the load is maintained [3], [21]. The series configuration removes the traditional transmission path altogether and puts the energy storage mechanism in series with the engine and load. This typically requires a change in energy domain between the engine and load so some transmission efficiency is lost. The benefit of this architecture is that engine operation is completely decoupled from power demand so optimal engine management is possible [3], [21]. Finally, the power-split architecture combines the parallel and series into a single architecture which increases the powertrain complexity but offers the greatest flexibility [3], [21]. Examples of these architectures are shown in Fig. 1.2.

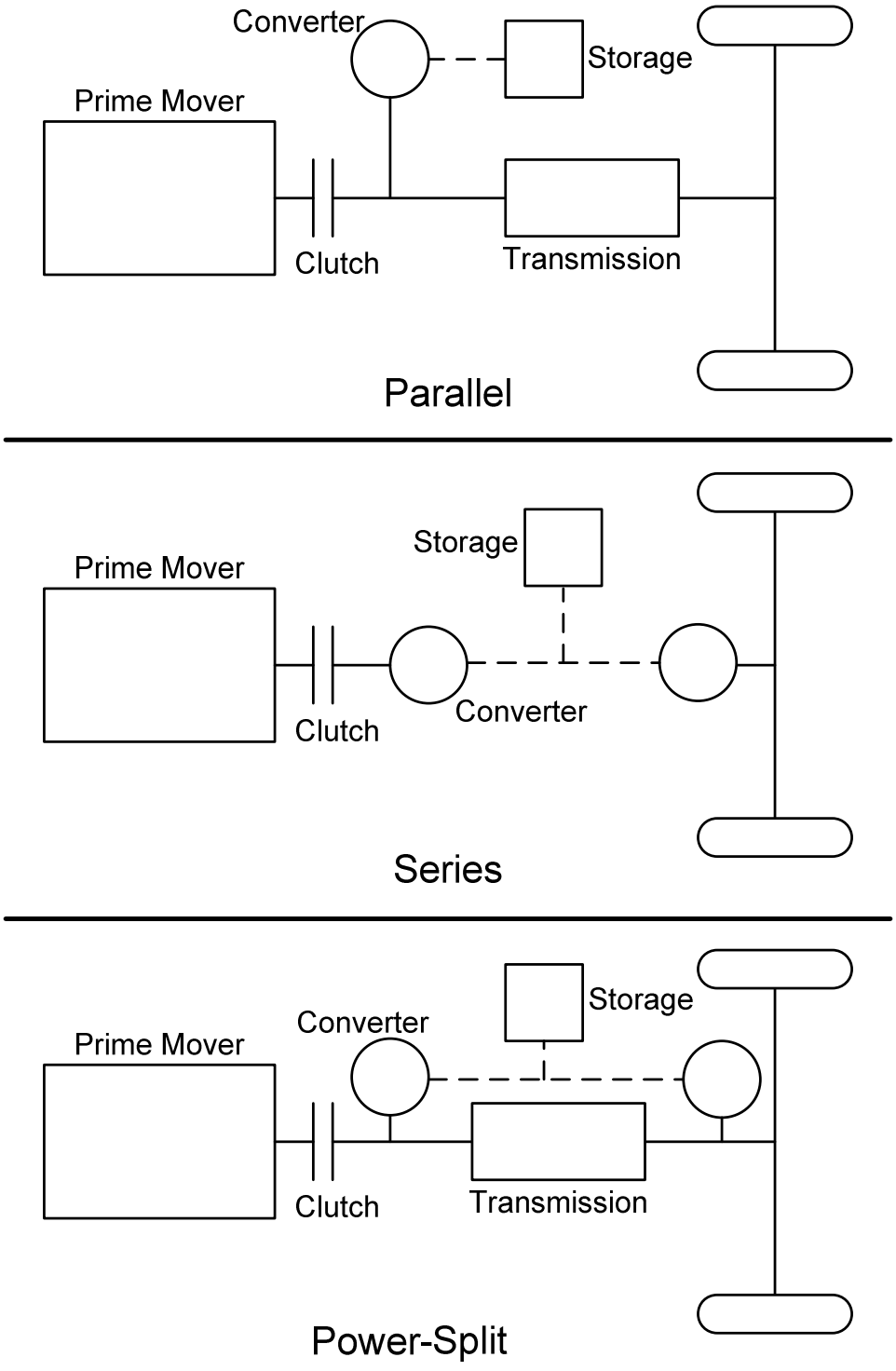

Figure 1.2: Hybrid powertrain architectures

1.2 Energy Management

Design of EMS's for hybrid vehicles has been an active area of research for many years. There are numerous approaches to designing these strategies ranging from computationally demanding off line optimization techniques to heuristically derived rules [2], [3]. For real-time implementable EMS's, rule-based, stochastic dynamic programming (SDP), and MPC are common design methods [14], [19], [20], [22–28].

Rule based EMS's use a set of rules or logic to control the powertrain [19], [23], [27], [28]. They are typically extrapolated from global optimization assessment performed using deterministic dynamic programming (DDP) over an assumed duty cycle. Since the optimization is not causal, it is approximated with a causal logic. This logic can then be implemented in real-time but it is usually suboptimal. In the case of [27] there was nearly a 10% decrease in fuel economy between the rule based strategy and the optimum benchmark derived using dynamic programming. Due to the cycle-dependent nature of this derivation, the performance cannot be guaranteed under arbitrary duty cycles.

SDP uses probability maps in place of an assumed duty cycle to make an estimate of what the vehicle will be required to do in the future and optimizes using this estimate [20], [24], [25]. The benefits of this approach over the rule based design are that the solution is not limited to a specific duty cycle and a causal control strategy is determined without further analysis of the results. However, this optimization procedure still includes some implicit assumption of the duty cycle.

MPC is an attractive control method for hybrid vehicle applications because the duty cycle need not be known a priori. Unlike the rule based and SDP solutions, MPC does not require any knowledge of the future duty cycle, or its statistical nature, to compute the control solution. Rather, a model of the system is used to predict how the powertrain will respond to a sequence of inputs. This enables one to express a finite horizon objective function as just a function of the control sequence. However when computing the cost of a control sequence, one can choose to include information about the future duty cycle in the prediction horizon. Since the MPC algorithm has the flexibility to be implemented with complete to no future knowledge, it has seen recent application in design of EMS's for hybrid vehicles [14], [22], [26]. The

drawback of MPC compared to the rule-based and SDP methods is that more intense online computation is required.

The proposed EMS design method will make use of MPC's flexibility to design real-time implementable control strategies that can be extended to all hybrid vehicle applications. To overcome the computational demand, a systematic approach for designing convex quadratic objective functions will be employed. These objective function will seek to maximize overall operational efficiency while meeting a duty cycle and observing system constraints.

1.3 Generality of Approach

To demonstrate the generalizability of the proposed EMS method, a non-traditional hybrid vehicle is considered. This system is a refrigerated delivery truck which incorporates thermal storage. It will be demonstrated that the proposed method can be applied to this system with minimal changes to the procedure. This system is chosen because it operates in a different energy domain than electric and mechanical hybrids while still encountering rapid transient loads, so the potential for storage is similar. In thermal applications, the stored potential cannot be extracted without additional work. Therefore, these systems do not employ the standard architectures outlined in Section 1.1.2. For thermal energy storage (TES), either refrigerant or air must be pumped through the device in order for a controlled heat exchange to occur (there will be some passive heat transfer but this is typically minimized by insulation). These systems can have many different configurations depending on the method for storing and extracting energy from the storage device. For example the prime mover could be used to only charge the storage unit and a separate actuator could be used to discharge the storage unit to meet the operator's demand.

Thermal storage can be achieved through sensible and latent heat. Sensible heat storage is the energy storage which is accomplished through a temperature change in the storage medium and latent heat storage is accomplished through a phase change in the storage medium [29]. Latent heat storage provides a high thermal storage density with relatively little change in temperature and volume [29]. Therefore, phase change materials (PCM) are particularly attractive for TES's. Such systems have already been studied in the context of building heating and cooling systems in the form of large chilled water tanks [30]. However, due to the high

thermal storage density, novel PCM's and storage architectures are starting to be evaluated for small scale storage in refrigerated transport systems [31]. Figure 1.3 shows a schematic of one TES embodiment.

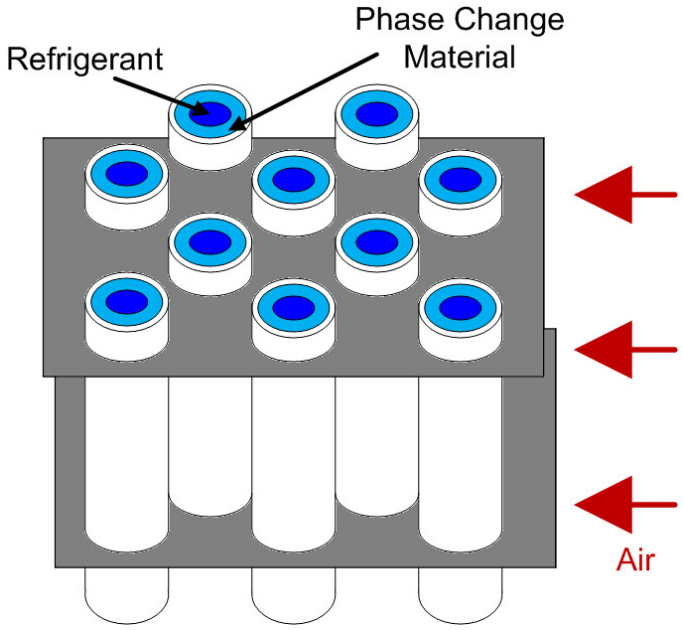

Figure 1.3: Example of thermal energy storage device with bank of tube geometry

1.4 Thesis Scope

In this thesis, there will be no discussion of how to optimize the design of a hybrid vehicle or an analysis of which storage technology to choose for a particular application. The design of hybrid power systems is a complex engineering challenge in its own right with many considerations such as peak demand, length of duty cycle, and storage charge/discharge rates. This work assumes that one is working within the constraints of a predefined architecture and the goal is to design a control strategy which will utilize the storage capability to improve operating efficiency. To achieve this goal, a method for formulating an EMS is presented which can accommodate varying degrees of duty cycle preview (from none to complete), different degrees of system knowledge (first principles models, black-box system ID, etc.), and is computationally compact enough to facilitate online implementation in vehicle systems. To formulate the online optimization problem, linear discrete MPC is used. The benefit of the proposed method is that it is a flexible, generally applicable process. However, it is not guaranteed to produce the global

optimal solution which one could achieve using offline optimization techniques applied to an assumed duty cycle [32]. The generalizability of the proposed method will be demonstrated through two case studies. In the first case study optimization of a series hydraulic hybrid vehicle (SHHV) is considered. For this system a linear prediction model is defined from a first principle analysis of the system and no future knowledge of the duty cycle is used. The performance of the control is validated using a hardware-in-the-loop system and the robustness of the proposed method versus other common EMS design techniques is demonstrated. For the second case study, optimization of a refrigerated truck is considered, in which system ID is used to derive a prediction model and complete knowledge of the duty cycle is assumed. It will be shown that, despite differences in system architecture and knowledge, the proposed method is able to achieve reduction in energy consumption for both cases.

1.5 Organization of Thesis

This thesis is organized as follows. Chapter 2 introduces the proposed EMS design method as well as provides a brief overview of MPC. An application of the proposed methodology to a SHHV powertrain is given in Chapter 3. This chapter includes a first principles analysis of the hydraulic hybrid powertrain as well as a discussion of the hardware-in-the-loop system used for control validation. In addition, rule-based and SDP approaches to energy management are also presented in Chapter 3 for benchmarking of the proposed method. In Chapter 4 one will find a refrigerated delivery truck case study. A discussion of the TES device using PCM is presented along with simulation results. In this study, two duty cycles are evaluated to characterize their impact on energy optimization. Finally, concluding remarks and future work are presented in Chapter 5.

Chapter 2 Energy Management Strategy Design

A methodology for designing energy management strategies (EMS's) applied to mobile power systems with energy storage is presented in this chapter. The purpose of this method is to provide a real-time implementable control strategy which satisfies operator demands while improving energy use within the vehicle. Mobile applications represent a significant portion of global energy consumption and present a number of unique challenges. The energy management challenge in the context of these systems is a constrained optimization problem in which one may have complete to no knowledge of the duty cycle and the control complexity is constrained by the need for fast update rates. The goal is to develop a tool which is generalizable to this entire class of systems. Therefore, the EMS must satisfy two criteria:

1. Computationally compact enough to be implemented in real-time with limited computing power
2. Allow for varying levels of duty cycle preview to be used.

The proposed methodology is outlined in Fig. 2.1. It utilizes a hybrid modeling approach to decompose the operation of this class of systems into modes corresponding to how power is generated and transmitted within the system. For example, in the parallel electric hybrid vehicle architecture shown in Fig. 2.2 there are two modes of operation; one when the prime mover is used to generate energy and one when it is disengaged and energy can be stored or drawn from the battery. For each mode, an online optimization problem is defined which is solved using model predictive control (MPC). MPC is used because it is an online optimization framework that allows for constraints on the inputs and outputs [33]. Finally, a supervisory logic is designed which regulates the switching between these modes. The advantage of decomposing the system operation into modes is that only one set of system dynamics is considered in the optimization

problem and the objective of the optimization problem can be unique to each mode. If this decomposition was not used, then the optimization problem would likely be a mixed integer programming problem since the system dynamics would include a combination of continuous and discrete states [34], [35]. In this case, the number of integer variables scales linearly with the prediction horizon length and since the mixed integer programming problem belongs to the class of *NP*-complete, the computation time scales at worst exponentially with problem size [35]. In contrast, a convex quadratic programming problem belongs to the class of *P* and its computation time is upper bounded by a polynomial whose size is that of the problem [36]. By considering only continuous dynamics and carefully defining the objective function it will be shown how this approach to EMS design results in convex quadratic optimization problems.

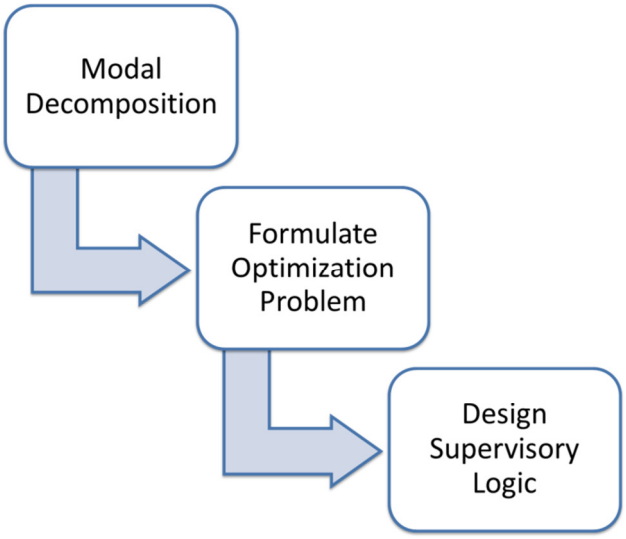

Figure 2.1: EMS design process

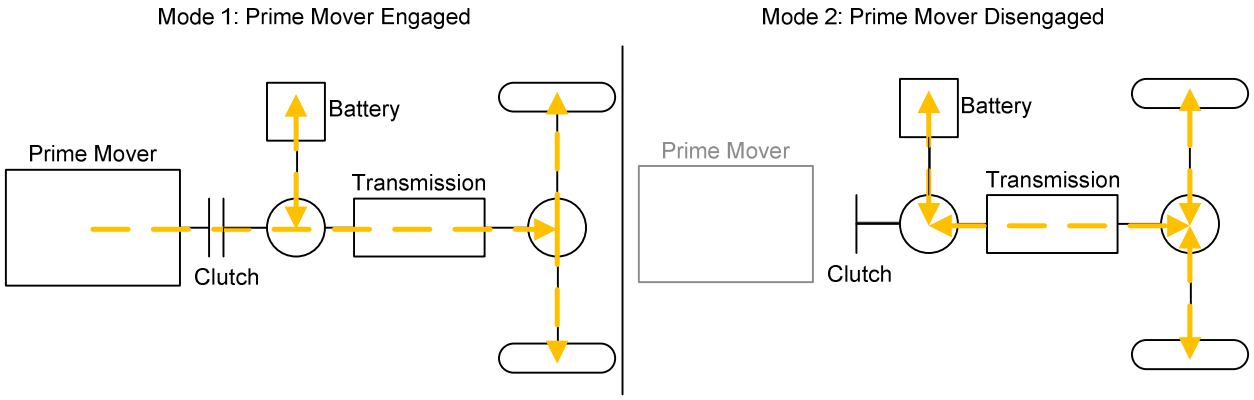

Figure 2.2: Example of modal decomposition for a parallel electric hybrid powertrain

Within each mode of operation, a system wide perspective is taken to define the optimization problem. The objective function is constructed by including quadratic cost terms associated with the efficiency characteristics of each actuator as well as cost for the performance objectives. Then the weighting of each efficiency cost term which leads to maximum overall efficiency is identified. The resulting optimization problem is coupled with the MPC framework to derive a real-time implementable optimal controller capable of integrating various levels of duty cycle knowledge. Other investigations into energy management in vehicles with energy storage have focused on fuel consumption minimization to define their optimization problem [19], [20], [23], [27]. Unfortunately fuel consumption measures, such as the Willans line model [37], are often complex nonlinear functions extrapolated from empirical mappings that do not lend themselves to fast online optimization. Furthermore, these objective functions are not readily expanded to other systems since they require detailed fuel consumption maps. However, minimizing fuel consumption does not require the additional system analysis to define efficiency characteristics for each actuator. Furthermore, the global minimum solution to the fuel minimization problem is the lowest possible fuel consumption whereas the proposed method is an approximation of this objective.

2.1 Model Predictive Control

MPC is a finite horizon optimal control framework which uses a model of the system to express future values of the outputs in terms of previous control decisions within the prediction horizon. Through this transformation, one is able to restate the objective function, which is typically a function of the states, outputs, and inputs, as just a function of the control decisions. In this way, solving for the trajectory which minimizes one's objective function over the prediction horizon reduces to solving for the optimal control sequence. The first element of this sequence is then applied to the system and the process is repeated at every discrete instance the control is updated [33].

This control method has been utilized for EMS's in many different applications [22], [26], [38]. It is attractive for hybrid vehicle applications because it is an online optimization method that can be formulated with different degrees of assumed future information. One has flexibility in choosing the length of the prediction horizon and what duty cycle is used in the

prediction horizon. For some applications, like passenger vehicles, it is difficult to predict future demand because environmental variables can have a significant impact on driver behavior and driver behavior can vary significantly between individuals. In this case a MPC formulation with a constant demand over a short prediction horizon could lead to the most robust performance. One of the major drawbacks to using MPC is the intense online computation required. A cornerstone of the proposed method is to use a system wide analysis to motivate the design of a convex quadratic objective function to produce an online optimal control strategy which maximizes overall system efficiency.

In this work linear discrete MPC is considered. The design of the controller can be broken down into two parts. First is the construction of the prediction model which is used to estimate system response in the prediction horizon. Second is the construction of the objective function which mathematically defines the goals of the controller.

2.1.1 Prediction model

There are many methods for deriving discrete linear representations of complex systems, such as deriving a linear model approximation, performing Taylor series expansions of nonlinear system equations then discretizing linearized system dynamics, or system identification. From each of these methods, one arrives at the familiar discrete state space representation for a *n* state and *m* input dynamic system. Note that in the model given by Eq. 2.1 it has been assumed there is no direct feed through of the inputs.

$$\begin{aligned} X(i+1) &= A \cdot X(i) + B \cdot U(i) \\ Y(i) &= C \cdot X(i) \end{aligned} \tag{2.1}$$

$$X = \left[x_1, x_2, \cdots, x_n\right]^T \tag{2.2}$$

$$U = \left[u_1, u_2, \cdots, u_m\right]^T \tag{2.3}$$

Here *A*, *B*, and *C* are the system matrices, *X* is the vector of states, and *U* is the vector of inputs. The discrete state space representation is then used to transform an objective function into a collection of cost terms that are just functions of the inputs. Consider the objective function given by Eq. 2.4, this function penalizes deviation of the output, *y*, from the reference,

ref. Using the prediction model, each future value of *y* in the prediction horizon can be expressed as an initial state measurement and a summation of control decisions, as shown in Eq. 2.5.

$$J = \sum_{i=1}^{N} \left(y(i) - ref(i) \right)^2 \tag{2.4}$$

$$\begin{aligned}
Y(i+1) &= C\left(AX(i) + BU(i)\right) \\
Y(i+2) &= C\left(A\left(AX(i) + BU(i)\right) + BU(i+1)\right) \\
Y(i+3) &= C\left(A\left(A\left(AX(i) + BU(i)\right) + BU(i+1)\right) + BU(i+2)\right) \\
&\vdots \\
Y(i+N) &= CA^N X(i) + CA^{N-1} BU(i) + \ldots + CABU(i+N-2) + CBU(i+N-1)
\end{aligned} \tag{2.5}$$

The transformed optimization problem is given by Eq. 2.6 and Eq. 2.7, where *V* is the stacked vector of each input signal over the prediction horizon. The Hessian matrix, *H,* and the vector *F* contain all of the coefficients which result from expanding the transformed summation. Note that in this objective function all terms which are independent of *U* have been suppressed since the value of *U* which minimizes *J* is independent of these constant terms.

$$J = V^T HV + F^T V \tag{2.6}$$

$$V^T = [u_1(i), \cdots, u_1(i+N-1), \cdots, u_m(i), \cdots, u_m(i+N-1)] \tag{2.7}$$

In this example, the objective function is quadratic and the prediction model is a set of linear relationships. Therefore the transformed optimization problem is a quadratic programming problem. This is a structure that lends itself to quick evaluation and if the Hessian is positive semi-definite the optimization problem is convex [39]. A convex optimization problem with a convex space over which one is searching for solutions has the property that a local minimum is also a global minimum [40]. In this EMS design method, when constructing the objective function only quadratic functions of the inputs and outputs will be considered so the quadratic programming structure can be utilized to reduce computational demand and prevent local minima. Appendix E gives a more detailed discussion of how a linear discrete prediction model is used to transform quadratic cost terms.

2.1.2 Objective function design

When formulating the goal of minimizing energy consumption or maximizing the efficiency of a system, there are many different quantities one could consider, such as fuel consumption, power, and exergy destruction [26], [38], [41]. Many of these metrics are complex nonlinear functions of the system inputs/states/outputs and therefore lead to nonlinear optimization problems. Solving a nonlinear optimization problem online in a vehicle system poses many challenges; large computational demand, no guarantee of a solution, and the presence of local minima which typically require iterative searches. These challenges further increase memory and processing demand. Alternatively, a quadratic cost on output or input tracking is a strictly convex function and a summation of such terms is a convex optimization problem. Solving such problems is much more tractable for online implementation in mobile power systems. Therefore, the challenge is to translate the goal of optimizing energy use into a collection of quadratic set point tracking terms.

To perform this transformation, an analysis of the system and its component efficiencies is utilized. In most power systems the primary consumers of energy are the actuators and therefore, the first step in constructing the objective function is to identify each actuator, and characterize its efficiency as a function of the system inputs, states, and outputs. These relationships are often available in the form of efficiency maps which can be generated through experimental data or first principles system analysis. Examples of efficiency maps for a variable displacement pump, internal combustion engine, and gas compressor are shown in Fig. 2.3.

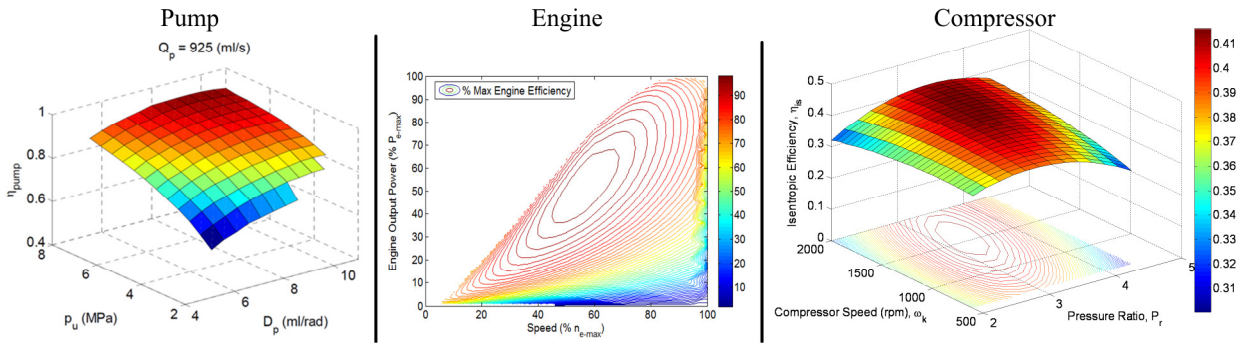

Figure 2.3: Efficiency maps of different components

From these efficiency maps one can construct a quadratic cost term in the form of Eq. 2.8 which penalizes deviation from the operating condition of highest efficiency. Note that each tracking cost is normalized so that all terms by default have an equal weighting. Each actuator's efficiency could be characterized by summation of many input, state, and output tracking terms. To characterize the whole system, a weighted sum of these individual objective functions is taken as shown in Eq. 2.9. The weighting on each of these terms is used to add greater emphasis to the component(s) which dominate the overall system efficiency. One means of determining this weighting is to do an iterative search over different weighting combinations for a nominal duty cycle. These weights are tuned offline and do not need to be recomputed during operations. To formulate the overall objective function, this efficiency objective would be combined with a performance objective.

$$J_{actuator,j} = \sum_{i=1}^{N} \left(\frac{x(i) - x_{des}}{x_{\max}} \right)^2 + \left(\frac{y(i) - y_{des}}{y_{\max}} \right)^2 + \left(\frac{u(i) - u_{des}}{u_{\max}} \right)^2 \quad (2.8)$$

$$J_{eff} = \sum_{j} \lambda_j \cdot J_{actuator,j} \quad (2.9)$$

In addition to the performance and efficiency objectives, the cost function can be augmented with an integral cost term to compensate for steady state tracking error resulting from modeling errors introduced by the linear, discrete system approximation. The discrete integral is formulated as the sum of the previous tracking error times the time step (Δt) given by Eq. 2.10. The integral cost term is then simply the sum of this error over the prediction horizon, see Eq. 2.11, and the cost term with all of the constant terms suppressed is given by Eq. 2.12.

$$z(i) = z(0) + \Delta t \sum_{j=1}^{i-1} y(j) - ref(j) \quad (2.10)$$

$$J = \sum_{i=1}^{N} \left(z(i) \right)^2 = \sum_{i=1}^{N} \left(z(0) + \Delta t \sum_{j=1}^{i-1} y(j) - ref(j) \right)^2 \quad (2.11)$$

$$J = 2 \cdot z(0) \cdot \Delta t \sum_{i=1}^{N} \sum_{j=1}^{i-1} y(j) + \Delta t^2 \sum_{i=1}^{N} \left(\sum_{j=1}^{i-1} y(j) \right)^2 - 2 \cdot \Delta t^2 \sum_{i=1}^{N} \left(\sum_{j=1}^{i-1} y(j) \sum_{j=1}^{i-1} ref(j) \right) \quad (2.12)$$

2.1.3 Solution methods

There are a variety of methods for solving quadratic programming problems. In the Matlab optimization toolbox, the 'quadprog' command is specifically designed to solve these optimization problems subject to both equality and inequality constraints [42]. However, under some additional constraints these problems can be made to be convex enabling the use of fast, simple solution methods, such as Newton's method. For a quadratic programming problem to be convex, the Hessian matrix must be positive semi-definite [39]. Using the method described in Section 2.1.2 one is guaranteed that the efficiency objective is convex because it is a convex quadratic optimization problem subject to a set of linear constraints from the prediction model. Therefore if the performance objective is also convex, then the overall optimization problem will be convex. Strict convexity can be achieved by choosing the performance and efficiency objectives such that, when coupled with the prediction model, the value of every input is penalized in the prediction horizon.

In some instances it may be desirable to transform a convex quadratic optimization problem subject to inequality constraints into an unconstrained optimization problem. For example if one wishes to apply Newton's method. This is accomplished through the use of logarithmic barrier functions [39]. The barrier function approximates the inequality constraint by adding convex cost terms which have small cost away from the boundaries and a large cost near the boundaries, see Fig. 2.4. The logarithmic barrier function is given by Eq. 2.13 and the augmented objective function is given by Eq. 2.14. Here q is an additional weighting placed on the original objective function and p is the number of inequality constraints.

$$a < x < b \approx -\log(a-x) - \log(b-x) \qquad (2.13)$$

$$J_{augmented} = q \cdot J - \sum_{k=1}^{p} \log\left(f_k(x,a)\right) \qquad (2.14)$$

$$f_k(x,a) = a - x \qquad (2.15)$$

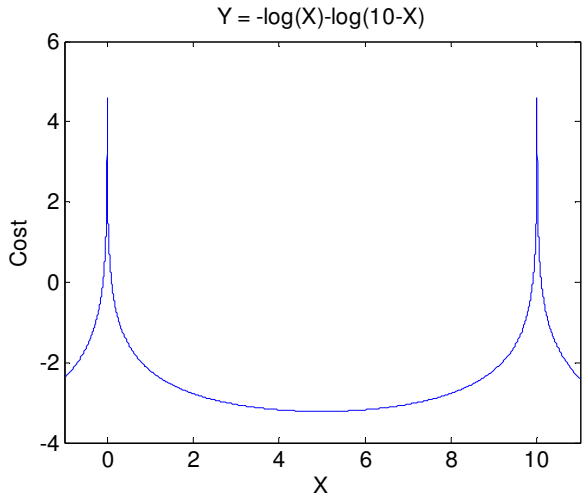

Figure 2.4: Example of logarithmic barrier function

2.2 Supervisor Design

The final component of the proposed method is to construct a supervisory logic for regulating switching between each operating mode. This logic could be derived heuristically from an informed understanding of the system behavior or extrapolated from simulation and experimental studies. One possibility for formulating this logic is to use an offline optimization approach, such as dynamic programming, to compute the optimal behavior over a prescribed duty cycle and then use those results to motivate the conditions for mode switching. In many applications this decision will be a function of the state of charge of the storage unit as this will directly indicated whether there is capacity available for storing energy or there is energy available to power the vehicle. When designing these strategies, care should be taken to avoid Zeno behavior as this rapid switching could damage components [43]. One method for preventing this behavior which will be demonstrated in Chapter 3 is inclusion of a dwell time constraint [43]. The dwell time constraint prevents the system from switching operating mode until a predefined amount of time has elapsed. In this way, one can ensure that the transient dynamics associated with mode switching have died out. This prevents components, like the IC engines, from cycling on/off too rapidly.

Chapter 3 Hydraulic Hybrid Vehicle Study

There is interest in hydraulic hybrids because fluid power has a higher power density and the accumulator can tolerate more charge/discharge cycles than conventional electric storage [17]. Research into hydraulic hybrids spans a wide range of applications from heavy duty vehicles, like city buses, to small passenger vehicles [11], [18–22]. In this chapter, the design of a model predictive energy management strategy (EMS) is presented using the method outlined in Chapter 2. Furthermore, two other methods for designing EMS's are presented: rule-based and stochastic dynamic programming (SDP). Each of these methods use information about the system and potential duty cycle differently and therefore will be uniquely affected by variations in these characteristics. In this investigation, a simulation study is used to quantify how a rule-based, SDP, and model predictive control (MPC) strategy are affected by variations in duty cycle and system parameters. For this study, only real-time implementable control strategies that have been validated experimentally are considered. This validation is conducted using the Augmented Earthmoving Vehicle Powertrain Simulator (AEVPS); a hardware-in-the-loop system containing the components of the hydraulic hybrid transmission while the engine and vehicle loads are emulated.

3.1 System Description and Model

For this study, a series hydraulic hybrid vehicle (SHHV) is considered. A series architecture was chosen because it decouples the engine and wheel operation, allowing for optimal engine management, without the additional hardware and complexity of a power-split architecture [21]. The SHHV is composed of a hydrostatic transmission with an accumulator, for energy storage, connected in series with the prime mover [21]. The accumulator can be used to

capture excess power generated by the engine, assist the engine in supplying the operator's power demand, or completely supply the demand power. A schematic of the SHHV is presented in Fig. 3.1.

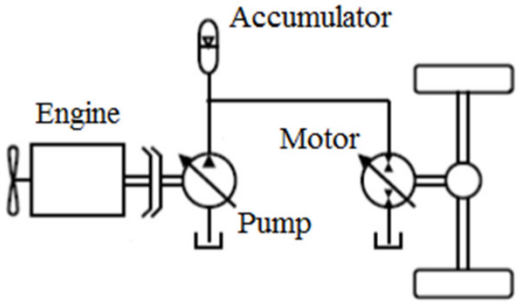

Figure 3.1: Schematic of a series hydraulic hybrid powertrain

For the purpose of this study, the hydraulic hybrid vehicle will be represented experimentally using the AEVPS. This testbed was originally developed as part of the Caterpillar Electromechanical Systems Lab at the University of Illinois at Urbana-Champaign and is used as a MIMO hardware-in-the-loop testbed for studying mobile electro-hydraulic powertrains [44–46]. A picture of the AEVPS is given in Fig. 3.2 and a schematic of the electro-hydraulic powertrain is shown in Fig. 3.3. From this schematic one can see that a throttling valve is used to regulate the transmission of energy to the load unit. For heavy mobile applications a variable displacement pump/motor is preferred over valve control because this allows for regenerative braking. However, for this study we are interested in evaluating the potential improvement that can be achieved with just improving powertrain operation and therefore no regeneration is considered. When evaluating the performance of an EMS without regeneration, this system has the same number of actuators as an optimally designed series hybrid and presents similar challenges and opportunities. See Appendix F for a discussion of how to operate the AEVPS.

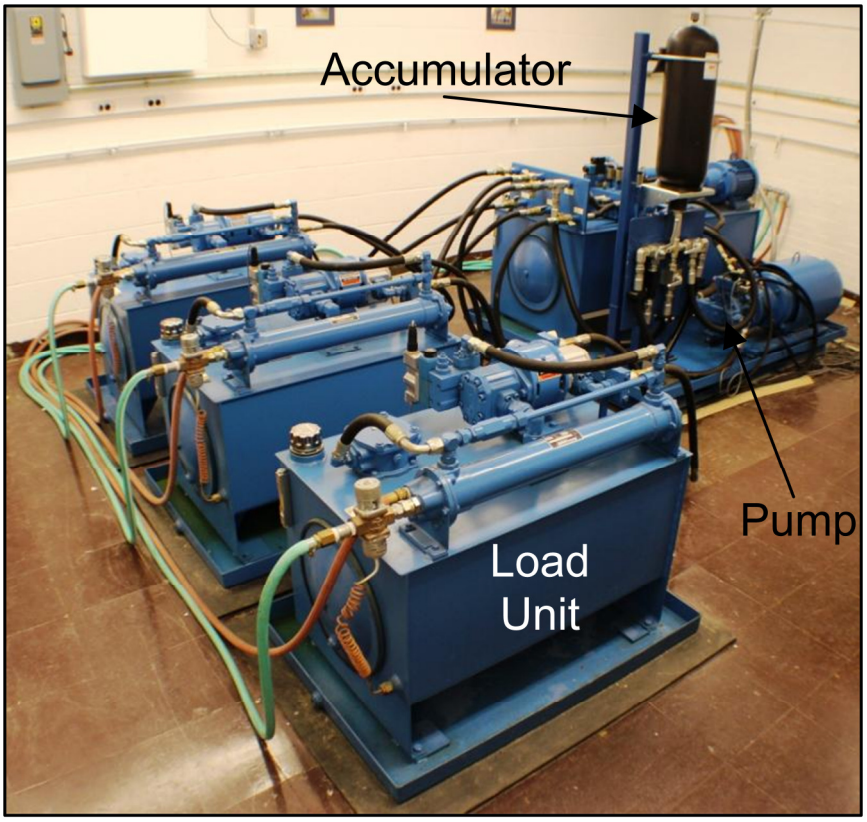

Figure 3.2: AEVPS at the University of Illinois at Urbana-Champaign

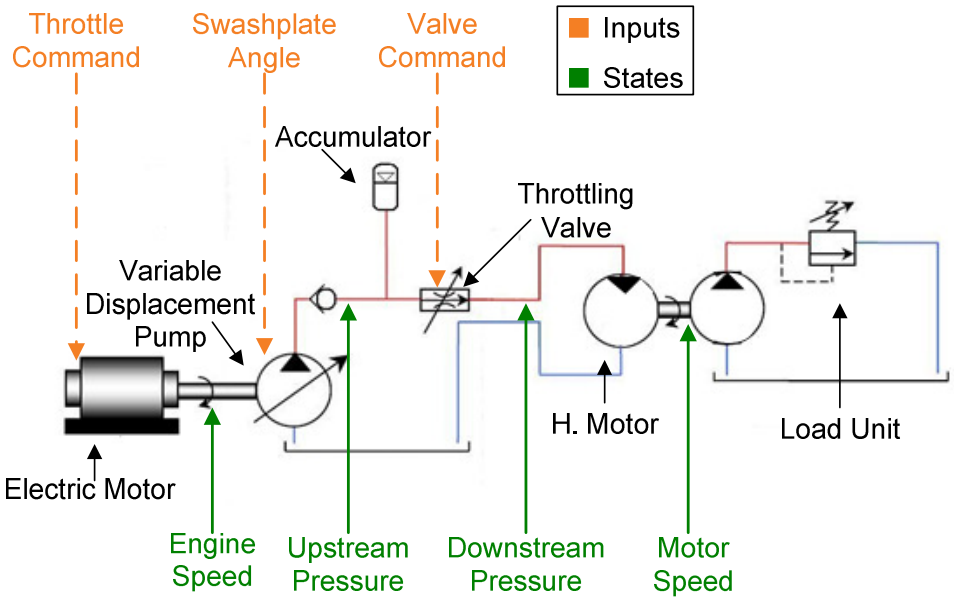

Figure 3.3: Schematic of the AEVPS powertrain

There are two primary sections to the AEVPS: the powertrain and the load unit. The major components of the AEVPS powertrain are: the prime mover, a variable displacement axial piston pump with a maximum displacement of 71 cc/rev (max. swashplate angle 0.314 rad) and a maximum flow rate of 128 L/min, a 18.9 liters gas charged accumulator with a precharge pressure of 5.17 MPa, an electronic proportional valve, and a 26.5 cc/rev hydraulic gear motor with a maximum flow rate of 79 L/min. The maximum operating pressure for the AEVPS powertrain is 20 MPa. The major components of the load unit are a 26 cc/rev hydraulic gear pump and an electronically controlled pressure relief valve (PRV). The load unit is used to emulate the driving loads experienced by a passenger vehicle via regulation of the pressure required to activate the PRV [47]. The maximum pressure of the load unit is 20 MPa and only one of the three available load units within the AEVPS is used in this study.

3.1.1 Powertrain model

For this study a ¼ scale powertrain model is used in which the engine power and vehicle loads have been scaled down by a factor of 4. This ensures that the simulation results comply with the hardware limitations of the AEVPS but should be scalable and applicable to a full-sized vehicle. The prime mover is a diesel engine emulated by an AC motor and computer control [44], [46], [48]. By using computer control to force the AC motor to behave according to modeled engine dynamics, the AEVPS is able to emulate a variety of different engines. This added flexibility enables the experimental system to represent a variety of engine/powertrain combinations. In this case, the maximum power output of the emulated engine was chosen to be one fourth that of a 2009 Toyota Prius' engine (18 kW) [49]. The AC motor can provide up to 22.37 kW of power with a maximum speed of 188.5 rad/sec and a maximum torque of 121 Nm. When used to emulate the scaled diesel engine, the maximum supplied power is 18.1 kW. The efficiency characteristics of the emulated engine are shown in Fig. 3.4. From the engine efficiency map one can see that the peak operating efficiency occurs between 50-60% max engine speed and 40-60% max engine power. Constraining the engine operation to this regime is not possible in traditional vehicle powertrains where the engine speed and output power are determined by the operator's power demand. However, for hybrid architectures, power generation and demand can be decoupled.

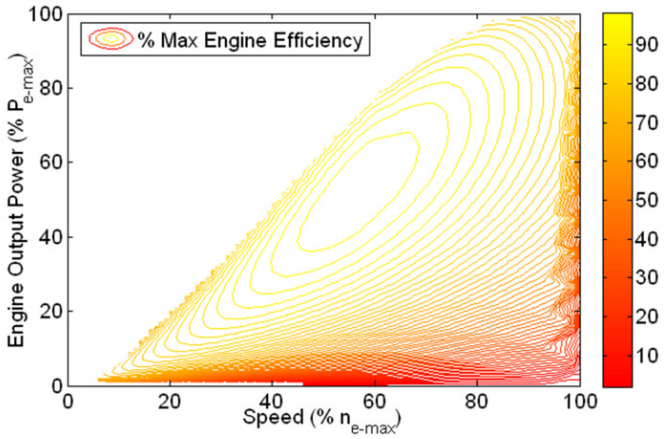

Figure 3.4: Diesel engine efficiency map

A dynamic model of the AEVPS powertrain is derived from a first principles analysis of the powertrain dynamics. This model is the basis for a linear discrete powertrain model which is used for the prediction model in the MPC. For this model, the dynamics of the swashplate and valve were ignored since they are much faster than the update rate of the MPC (settling times less than 0.1 second) [44]. Furthermore, the engine is treated as a torque source to reduce the model complexity. Equation 3.1 relates the engine torque to the engine speed and accounts for inertia, friction, and loading due to pressure across the pump.

$$I_e \cdot \dot{\omega}_e = u_e - \frac{\kappa_e}{2} \cdot \omega_e^2 - b_e \cdot \omega_e - K_P \cdot u_p \cdot P_u \qquad (3.1)$$

Here I_e is the engine inertia, ω_e is the engine speed, b_e and κ_e are engine friction coefficients, K_P is the pump flow gain, u_p is the pump's swashplate angle, and P_u is the upstream pressure.

The dynamics of the pressure upstream and downstream of the valve, denoted by P_u and P_d respectively, are a consequence of conservation of mass flow. The difference between flow into and out of the hoses causes the pressure within the hoses to change. For the upstream hose, it has been assumed that the accumulator pressure and upstream hose pressure are equal. This flow balance for the upstream hose is captured in Eq. 3.2 where Eq. 3.3 through Eq. 3.5 define each flow term.

$$\dot{P}_u = \frac{\beta_u}{V_u}\left(Q_P - Q_a - Q_v - \psi_u \cdot P_u\right) \quad (3.2)$$

Here β_u is the fluid bulk modulus of the upstream hose, V_u is the volume of the upstream hose, Q_p is the flow from the pump (Eq. 3.3), Q_a is flow into the gas charged accumulator (Eq. 3.4), Q_v is the flow through the valve (Eq. 3.5), and ψ_u is the upstream leakage coefficient.

$$Q_P = K_P \cdot u_p \cdot \omega_e \quad (3.3)$$

$$Q_a = \frac{Cap}{k} \cdot P_{pr}^{\frac{1}{k}} \cdot P_u^{\frac{-(k+1)}{k}} \cdot \dot{P}_u \quad (3.4)$$

Here *Cap* is the accumulator capacity, k is the specific heat ratio of the gas, and P_{pr} is the precharge pressure of the gas.

The flow through the valve is a function of the pressure drop across the valve and the voltage command sent to the valve via the valve's flow gain. The flow gain is approximated using a third order polynomial fit to data collected by stepping the valve command up from 0 to 5 V for a fixed swashplate angle and throttle command (Eq. 3.6). The curve fit is shown in Fig. 3.5.

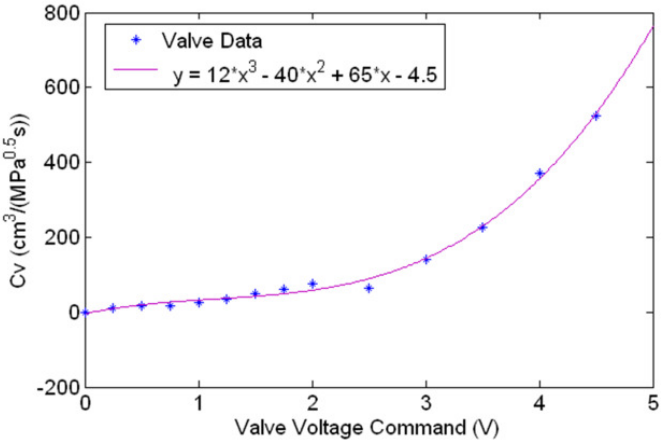

Figure 3.5: Curve fit for valve flow gain

$$Q_v = C_V \cdot \sqrt{\Delta P} \quad (3.5)$$

$$C_V \approx 12 \cdot u_v^3 - 40 \cdot u_v^2 + 65 \cdot u_v - 4.5 \quad (3.6)$$

$$\Delta P = P_u - P_d \qquad (3.7)$$

Here C_V is the valve flow gain and u_v is the valve voltage command. The downstream pressure dynamics, Eq. 3.8, relates the flow through the valve to the flow across the hydraulic motor and accounts for losses in the downstream hose.

$$\dot{P}_d = \frac{\beta_d}{V_d}\left(C_V \cdot \sqrt{\Delta P} - D_m \cdot \omega_m - \psi_d \cdot P_d\right) \qquad (3.8)$$

Here β_d is the fluid bulk modulus of the downstream hose, V_d is the volume of the downstream hose, D_m is the displacement of the hydraulic motor and ψ_d is the downstream leakage coefficient.

Finally, Eq. 3.9 relates the torque applied by the pressure difference across the motor to the motor speed. This accounts for the inertia and damping characteristics of the combined pump/motor couple as well as the load torque imposed by pressure within the load unit.

$$I_m \cdot \omega_m = D_m \cdot P_d - b_m \cdot n_m - \tau_L \qquad (3.9)$$

Here I_m is the motor inertia, b_m is the hydraulic motor damping, and τ_L is the load torque. The load torque accounts for the steady state loading which is encountered in urban driving environments and is defined by Eq. 3.10. This load model accounts for the viscous friction losses at the wheel and air drag. Note, for this study a no slip condition was assumed at the vehicle's drive wheels. Finally, the load torque is scaled down by a factor of 4 to be consistent with the engine power scaling.

$$\tau_L = \frac{1}{4}\left(b_w \cdot \omega_m + \frac{r_w}{2} \cdot \rho_{air} \cdot C_{drag} \cdot A\left(r_w \cdot \omega_m\right)^2\right) \qquad (3.10)$$

Here b_w is the lumped viscous friction coefficient of the wheel and transmission coupling, r_w is the wheel radius, ρ_{air} is the air density, C_{drag} is the drag coefficient, and A is the vehicle area. Equations 3.1 through 3.10 provide a complete description of the AEVPS powertrain dynamics.

Finally, the fuel consumption of the engine is estimated using Eq. 3.11 where θ is the throttle command, k_f is the fuel consumption coefficient, and 0.43 $g.s^{-1}$ is the fuel consumption under idle conditions.

$$\text{fuel rate} = \max\left(k_f \cdot \omega_e \frac{\theta}{\theta_{\max}}, 0.00043 \right) \qquad (3.11)$$

3.1.2 Load emulation

The driving loads described in Eq. 3.10 are approximated in experimentation using load emulation. The load unit emulates the driving loads via computer control of a two-stage PRV's cracking pressure [47]. There are two components to the load emulation: the driving load model and the valve control. The driving load model, Eq. 3.10, computes the desired load torque based on a motor speed measurement. A PI controller is then used to regulate the voltage commend supplied to the PRV. The input to the load controller is the error between the desired load torque and a measurement of the pressure within the load unit multiplied by the pump's displacement, see Eq. 3.12.

$$e = \tau_{L,Des} - D_p \cdot P_L \qquad (3.12)$$

Here e is the error signal, $\tau_{L,Des}$ is the desired load torque, D_p is the pump displacement, and P_L is the pressure within the load unit. Since the load unit is incapable of providing an overriding load, the integral term of the PI controller is reset whenever the load torque crosses zero. A schematic showing the implementation of the driving load is shown in Fig. 3.6.

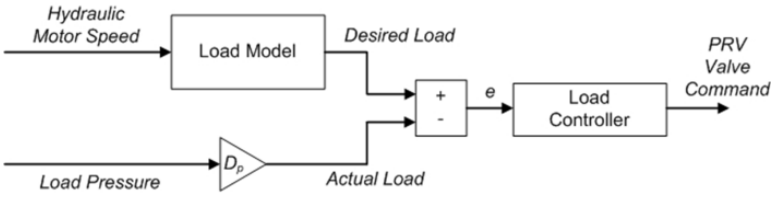

Figure 3.6: Driving load model implementation

3.1.3 Model validation

A comparison of the dynamic model and the physical system with load emulation is shown in Fig. 3.7. For this test each actuator was given a 1V step increase in command, held constant for 100s, and then stepped back to their nominal values (5V throttle command, 2.5V swashplate angle command, and 3.5V valve command). The valve command was stepped up at

200s, the swashplate command was stepped up at 400s, and the throttle command was stepped up at 600s. During this test, the load emulation unit was used. For a complete list of all parameters used to define the AEVPS model see Appendix A.

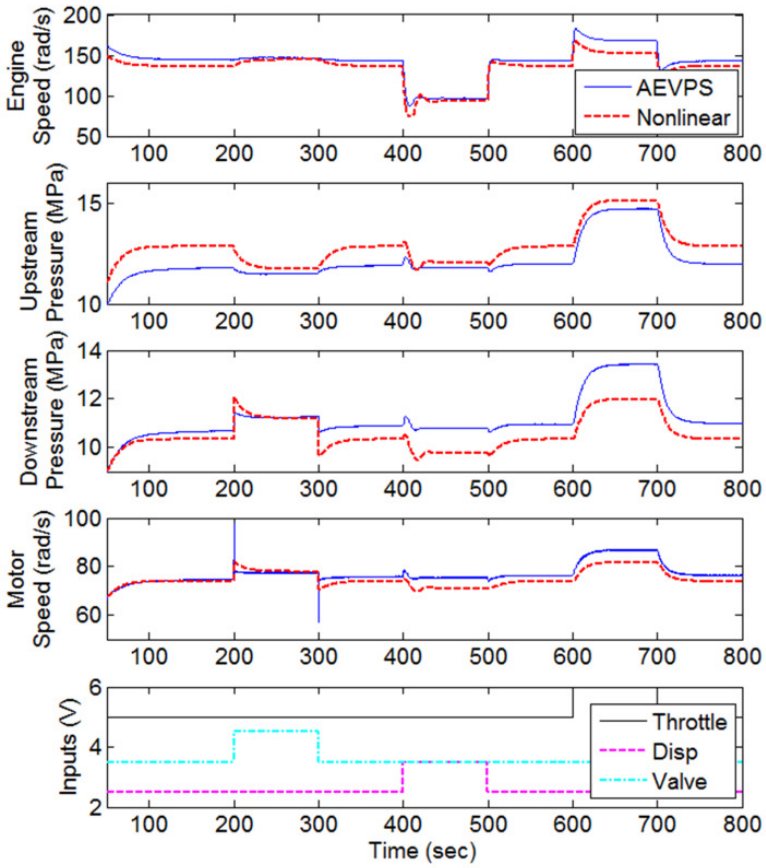

Figure 3.7: Comparison between AEVPS with load emulation and model for step changes in each actuator command

From Fig. 3.7 one can see that the model captures the time scales and approximate magnitudes of each powertrain state during the various step changes. Larger steady state errors are observed in the downstream pressure and motor speed predictions for the displacement and throttle command step tests due to the approximation of the downstream leakage losses as a single proportionality term. However, for all states, the steady state errors were within 15% throughout the test and the transients, which are of primary concern for the MPC, are well captured. Therefore, this level of agreement was found to be sufficient for control analysis.

3.2 Model Predictive Energy management

The MPC based EMS is implemented using a discrete linear model of the hybrid vehicle for prediction and objective functions motivated by an analysis of individual component efficiencies. The prediction model is derived by using a first order Taylor series expansion and a zero order hold to linearize and discretize a nonlinear system model about the current operating point. The controller solves for the sequence of throttle commands, pump swashplate angles, and valve openings which minimizes the objective function over the horizon, and applies the first element of this sequence to the system. The vehicle operation is decomposed into two modes and a supervisory logic is used to regulate switching between these modes. Furthermore, a dwell time constraint is employed to prevent high frequency engine on/off cycling. During the "ON" mode the MPC utilizes the power generation capabilities of the engine to track the desired vehicle speed while maximizing the powertrain's operational efficiency. During the "OFF" mode the MPC minimizes the engine use while tracking the desired vehicle speed. For both objective functions the desired vehicle speed is held constant throughout the prediction horizon because no prediction of the duty cycle is used in this formulation. The MATLAB code used to define the MPC in this study is given in Appendix G.

3.2.1 Prediction model

The prediction model used within the MPC algorithm is a discrete linear approximation of the model presented in Section 3.1.1. The linearization is done using a first order Taylor series approximation and the discretization is done using the zero order hold method. Throughout this section subscript o will be used to denote the operating point about which the model is linearized and δ will be used to denote the difference with respect to the operating point. First, the linearized system dynamics will be given, followed by the state space matrices.

Equation 3.13 is the linearized engine dynamics, Eq. 3.14 through 3.19 define the linearized upstream pressure dynamics, and Eq. 3.20 is the linearized downstream pressure dynamics. Note that within the prediction model the engine is treated as an ideal torque source to reduce the model complexity. Finally, Eq. 3.9 is already linear in the states so no further approximation is necessary. However, in order to keep the prediction model linear, the effect of

the load emulation is approximated by increasing the damping coefficient by b_L. The final motor dynamic equation is given by Eq. 3.21.

$$I_e \cdot \delta\dot{\omega}_e = \delta u_e - \left(b_e + \kappa_e \cdot \omega_{e,o}\right) \cdot \delta\omega_e - K_P \cdot u_{p,o} \cdot \delta P_u - K_P \cdot \delta u_p \cdot P_{u,o} \tag{3.13}$$

$$\begin{aligned} \delta\dot{P}_u = {} & A_1 \cdot K_P \cdot u_{p,o} \cdot \delta\omega_e + \left(A_2 \cdot A_3 - A_1 \left(\psi_u + \frac{C_{V,o}}{2\sqrt{\Delta P_o}} \right) \right) \delta P_u \\ & + A_1 \cdot \frac{C_{V,o}}{2\sqrt{\Delta P_o}} \cdot \delta P_d + A_1 \cdot K_P \cdot \omega_{e,o} \cdot \delta u_p - A_1 \cdot C_o \cdot \sqrt{\Delta P_o} \cdot \delta u_v \end{aligned} \tag{3.14}$$

$$A_1 = \left(\frac{V_u}{\beta_u} + \frac{Cap}{k} \cdot P_{pr}^{\frac{1}{k}} \cdot P_{u,o}^{\frac{-(k+1)}{k}} \right)^{-1} \tag{3.15}$$

$$A_2 = \frac{Cap \cdot (k+1)}{k^2} \cdot P_{pr}^{\frac{1}{k}} \cdot A_1^2 \cdot P_{u,o}^{\frac{-(2k+1)}{k}} \tag{3.16}$$

$$A_3 = K_P \cdot u_{p,o} \cdot \omega_{e,o} - C_{V,o} \cdot \sqrt{\Delta P_o} - \psi_u \cdot P_{u,o} \tag{3.17}$$

$$C_{V,o} = \frac{D_m \cdot n_{m,o}}{\sqrt{\Delta P_o}} \tag{3.18}$$

$$C_o = 36 \cdot u_{v,o}^2 - 80 \cdot u_{v,o} + 65 \tag{3.19}$$

$$\begin{aligned} \delta\dot{P}_d = {} & \frac{\beta_d}{V_d} \cdot \frac{C_{V,o}}{2\sqrt{\Delta P_o}} \cdot \delta P_u - \frac{\beta_d}{V_d} \left(\psi_d + \frac{C_{V,o}}{2\sqrt{\Delta P_o}} \right) \delta P_d \\ & - \frac{\beta_d}{V_d} \cdot D_m \cdot \delta\omega_m + \frac{\beta_d}{V_d} \cdot C_o \cdot \sqrt{\Delta P_o} \cdot \delta u_v \end{aligned} \tag{3.20}$$

$$\left(I_m\right) \cdot \delta\dot{\omega}_m = D_m \cdot \delta P_d - \left(b_m + b_L\right) \cdot \delta n_m \tag{3.21}$$

Combining Eq. 3.13, Eq. 3.14, Eq. 3.20, and Eq. 3.21 one arrives at the usual state space representation of the AEVPS powertrain dynamics. The system matrices are given by Eq. 3.22 and Eq. 3.23; the state and input vectors are given by Eq. 3.24 and Eq. 3.25. The following

system representation provides a compact linear description of the powertrain dynamics which is suitable for online implementation of the MPC algorithm.

$$A=\begin{bmatrix} -\frac{b_e+\psi_e\cdot\omega_{e,o}}{I_e} & -\frac{K_P u_{p,o}}{I_e} & 0 & 0 \\ A_1 K_P u_{p,o} & A_2A_3-A_1\left(\psi_u+\frac{C_{V,o}}{2\sqrt{\Delta P_o}}\right) & A_1\frac{C_{V,o}}{2\sqrt{\Delta P_o}} & 0 \\ 0 & \frac{\beta_d C_{V,o}}{2V_d\sqrt{\Delta P_o}} & -\frac{\beta_d}{V_d}\left(\psi_d+\frac{C_{V,o}}{2\sqrt{\Delta P_o}}\right) & -\frac{\beta_d D_m}{V_d} \\ 0 & 0 & \frac{D_m}{I_m} & -\frac{b_m+b_L}{I_m} \end{bmatrix} \quad (3.22)$$

$$B=\begin{bmatrix} \frac{1}{I_e} & -\frac{K_P P_{u,o}}{I_e} & 0 \\ 0 & A_1 K_P \omega_{e,o} & A_1 C_o\sqrt{\Delta P_o} \\ 0 & 0 & \frac{\beta_d C_o\sqrt{\Delta P_o}}{V_d} \\ 0 & 0 & 0 \end{bmatrix} \quad (3.23)$$

$$x=\begin{bmatrix}\delta\omega_e & \delta P_u & \delta P_d & \delta\omega_m\end{bmatrix}^T \quad (3.24)$$

$$u=\begin{bmatrix}\delta u_e & \delta u_p & \delta u_v\end{bmatrix}^T \quad (3.25)$$

3.2.2 Supervisory logic

The next component of the MPC based EMS is a supervisory logic used to regulate switching between operating modes and impose a dwell time constraint on engine operation. The default mode of operation is the "OFF" mode and the first part of the logic, shown in Fig. 3.8, checks when the desired engine torque has exceeded a threshold ($\tau_{theshold}$) or the upstream pressure falls below a threshold ($P_{theshold}$). This step is a check for when the accumulator no longer has sufficient energy stored to meet the demands of the control. The time when this condition occurs is stored as $t_{trigger}$.

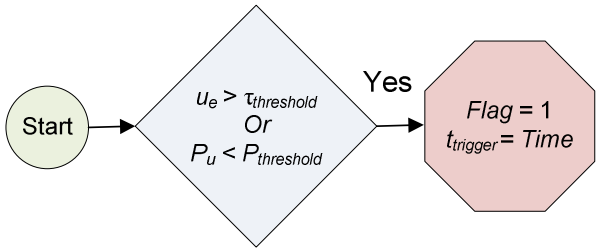

Figure 3.8: Supervisory logic, engine torque and pressure threshold detection

Once the thresholds have been checked the second part of the logic, shown in Fig. 3.9, determines which mode of operation to select for the next control update. It first checks whether the Flag indicating insufficient accumulator state of charge is still 1. If this is the case, it then checks if the ON mode has been maintained for t_{dwell} time. Once t_{dwell} time has expired, it then checks if the desired motor velocity is less than or equal to a threshold. By defining this threshold as done in Eq. 3.26 this condition can be used to ensure that there is sufficient charge in the accumulator to sustain the desired hydraulic motor speed.

$$\omega_{threshold} = \sqrt{\Delta P}\left(12 \cdot v^3 - 40 \cdot v^2 + 65 \cdot v - 4.5\right) / D_m \qquad (3.26)$$

Here v is the valve voltage command at which one evaluates the threshold. By adjusting v one can tune how aggressively the logic seeks to return to the OFF mode. The MATLAB code used to define the supervisory logic in this study is given in Appendix H.

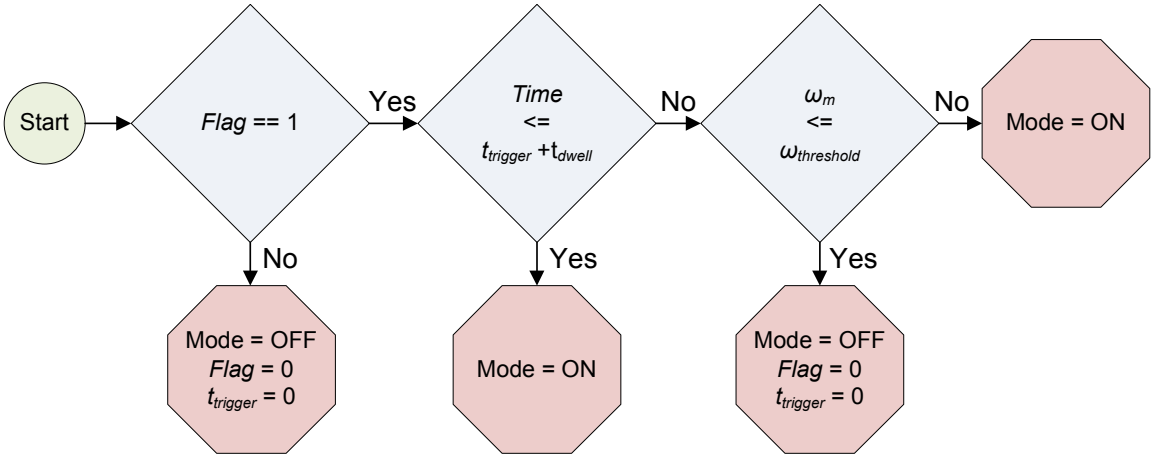

Figure 3.9: Supervisory logic, mode selection

3.2.3 Objective functions

The MPC uses convex quadratic objectives functions for both the "ON" and "OFF" modes to reduce the computational demands of the online optimization. The objective function used for the "ON" mode, given by Eq. 3.27 – Eq. 3.29, is composed of three parts. The first term enforces tracking of a desired hydraulic motor speed and is the primary objective. The second and third terms enforce efficient operation of SHHV powertrain. The λ_1 term seeks to optimize pump efficiency by maximizing upstream pressure and displacement, see Fig. 3.10 for a pump efficiency map at a fixed flow rate. This is accomplished by having the engine track low speed and high torque operation. To achieve these conditions the pump displacement is necessarily maximized, yielding greater upstream pressures. The λ_2 term seeks to minimize valve losses by having the AEVPS operate at low upstream pressure, reducing the pressure drop across the valve. Through a simulation study, the weighting of these cost terms which minimizes fuel consumption over the Urban Dynamometer Driving Schedule (UDDS) [50] was found to be $\lambda_1 = 1\text{x}10^{-4}$ and $\lambda_2 = 9\text{x}10^{-4}$. See Appendix B for the details of this study. Finally, the summation of the cumulative deviation from desired set points is used when evaluating the cost of a policy. This more heavily penalizes deviation from the desired values at the beginning of the prediction horizon, thereby improving tracking performance.

$$J_1 = \sum_{j=1}^{N}\sum_{i=1}^{j}\left[\underbrace{\left(\frac{\omega_m(i)-\omega_{m,des}}{\omega_{m,\max}}\right)^2}_{\text{Motor Tracking}} + \lambda_1 \cdot \underbrace{C_1}_{\substack{\text{Pump}\\ \text{Efficiency}}} + \lambda_2 \cdot \underbrace{C_2}_{\substack{\text{Valve}\\ \text{Efficency}}}\right] \tag{3.27}$$

$$C_1 = \left(\frac{\omega_e(i)-\omega_{e,des1}}{\omega_{e,\max}}\right)^2 + \left(\frac{u_e(i)-u_{e,des1}}{u_{e,\max}}\right)^2 \tag{3.28}$$

$$C_2 = \left(\frac{P_u(i)-P_{u,des}}{P_{u,\max}}\right)^2 \tag{3.29}$$

Here $\omega_{m,des}$, $n_{e,des1}$, $u_{e,des1}$, and $P_{u,des}$ are the desired motor speed, engine speed, engine torque, and upstream pressure respectively and N is the length of the prediction horizon. All of the cost terms are normalized with respect to their maximum values.

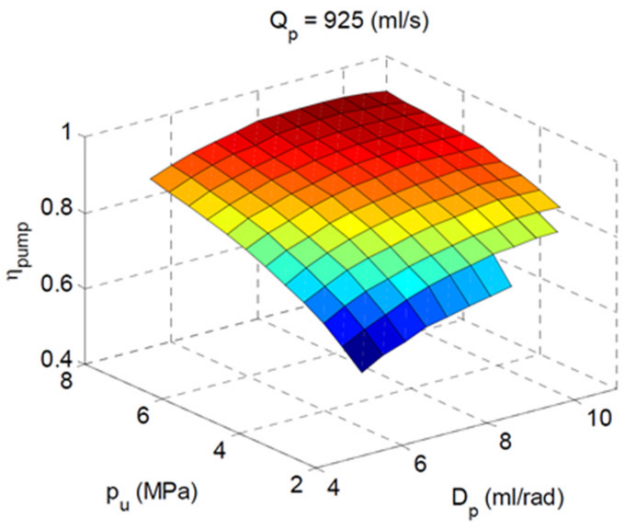

Figure 3.10: Efficiency map of the variable displacement pump for a fixed flow rate [48]

When power generation is not necessary, because the vehicle is decelerating or the accumulator can supply the requested power, the SHHV is operated in "OFF" mode. During this mode the engine should be operated under idle conditions to minimize fuel consumption. The objective function for this mode is given by Eq. 3.30 and Eq. 3.31.

$$J_2 = \sum_{j=1}^{N}\sum_{i=1}^{j}\left[\underbrace{\left(\frac{\omega_m(i)-\omega_{m,des}}{\omega_{m,\max}}\right)^2}_{\text{Motor Tracking}} + \lambda_3 \cdot \underbrace{C_3}_{\text{Engine Operation}}\right] \qquad (3.30)$$

$$C_3 = \left(\frac{\omega_e(i)-\omega_{e,des2}}{\omega_{e,\max}}\right)^2 + \left(\frac{u_e(i)-u_{e,des2}}{u_{e,\max}}\right)^2 \qquad (3.31)$$

Similar to the ON mode, λ_3 is set to $1\text{x}10^{-3}$ to ensure motor speed tracking has the higher priority. One of the advantages of MPC is that constraints can be easily applied on both the control signals and state values. For control of this system, upper and lower constraints are placed on each control variable as well as on the accumulator pressure. The constraints on control variables exist to ensure that actuator limitations are not exceeded by the controller. The upper limit on upstream pressure is used to prevent loss of energy through activation of a PRV (max operating pressure 20 MPa) and the lower limit ensures the control never attempts to draw

oil from an empty accumulator (accumulator precharge pressure: 5.17 MPa). The constraints are given by Eq. 3.32 through Eq. 3.35.

$$5.2 \text{ MPa} < P_u < 19 \text{ MPa} \quad (3.32)$$

$$0 \text{ Nm} < u_e < 121 \text{ Nm} \quad (3.33)$$

$$0 \text{ rad} < u_p < 0.314 \text{ rad} \quad (3.34)$$

$$0 \text{ V} < u_v < 5 \text{ V} \quad (3.35)$$

3.3 Alternate Energy management Methods

In addition to the MPC, two other methods for deriving an EMS are considered. The first is a rule-based strategy, which is derived from a deterministic dynamic programming (DDP) solution [51]. Even though DDP gives the global optimal solution, it assumes all future knowledge is known and therefore is not implementable on a physical system. The rule-based strategy develops rules to attempt to replicate the DDP accurately while being implementable on a physical system. The second method is SDP. Rather than using an exact duty cycle, the SDP method uses transition probabilities of driving behavior. This produces a causal control strategy since the future is not known exactly, but given as a probability map. One drawback of this method is it could lead to a suboptimal result if the transition probabilities do not accurate reflect the probabilities of the actual cycle. These alternate methods will be used to compare and contrast the robustness versus optimality tradeoffs of the proposed predictive energy management method.

3.3.1 Rule-based strategy

DDP is an optimization algorithm that calculates the global optimal solution for a system by starting at the end of a cycle and progressing backwards through time. For this system, a discrete-state, discrete-time method is used with a backwards facing model that assumes the trajectory is achievable at each time step [32]. The states of the system are the upstream pressure, downstream pressure, and motor speed, and the control variables are the valve command, swashplate angle, and engine speed. The engine speed is discretized into 5 rad/s

increments, the upstream pressure is discretized into 0.01 MPa increments, and the swashplate angle is discretized into 0.01 radian increments. Motor speed is given by the duty cycle and downstream pressure can be calculated from the other variables. Time is discretized into 1 second increments. The objective function is to minimize the fuel consumption over the entire duty cycle.

$$J = \min \sum_{k=o}^{N_c} fuel = \min \sum_{i=0}^{N_c} \left[\max \left(K_f \omega_e(i) \frac{\theta(i)}{\theta_{\max}}, 0.00043 \right) \Delta t \right] \qquad (3.36)$$

Here N_c is the total number of time steps in the duty cycle and Δt is the length of each time step. Since the DDP algorithm starts at the end of the duty cycle, the results are acausal because the future is assumed to be known exactly without external disturbances or model uncertainties. Therefore, these results must be transformed into a set of rules to develop a causal relationship between the outputs and control variables. To accomplish this transformation, a regression analysis of the DDP results over the UDDS was conducted to produce polynomial fits for throttle and swashplate angle commands based on engine speed, upstream pressure, and motor speed. These fits are given by Eq. 3.37 and Eq. 3.38. The valve command is regulated using a PI controller for tracking motor speed.

$$f_1(\omega_e, \omega_m) = -434.775 + 10.026 \cdot \omega_e + 0.017 \cdot \omega_m - 0.055 \cdot \omega_e^2 + 0.002 \cdot \omega_m^2 \qquad (3.37)$$

$$f_2(P_u, \omega_m) = -0.034 + 0.012 \cdot P_u + 0.001 \cdot \omega_m - 0.001 \cdot P_u^2 + 4.219 \cdot 10^{-6} \cdot \omega_m^2 \qquad (3.38)$$

In the above equations, f_1 is the throttle command function and f_2 is the swashplate angle command function. The r^2 values for the fits are 0.80 and 0.71 respectively. Despite good r^2 values for the polynomials they poorly capture the DDP results at high and low vehicle speeds. When the vehicle is travelling at highway speeds, the throttle and swashplate angle commands are too low, and when the vehicle is stopped, the throttle command is too high. To account for these discontinuities switching condition are added to the rules. At low speed the throttle command is set to the idle condition. When the vehicle is traveling at highway speeds, the pressure drop across the valve is low. Therefore, when the pressure drop across the valve is

below a minimum threshold the throttle command is set to 55 degrees and the swashplate angle is set to 0.16 rad. The final set of rules is given by Eq. 3.39-3.42.

$$Flag_1(t)=\begin{cases}0 & \text{For } \Delta P \geq 1\\ 1 & \text{For } \Delta P \leq 0.5\\ Flag_1(t-1) & \text{For } 0.5<\Delta P<1\end{cases} \quad (3.39)$$

$$Flag_2(t)=\begin{cases}0 & \text{For } n_m \geq 20\\ 1 & \text{For } n_m \leq 15\\ Flag_2(t-1) & \text{For } 15<n_m<20\end{cases} \quad (3.40)$$

$$\gamma=\begin{cases}55 & \text{if } Flag_1=1\\ 7 & \text{else if } Flag_2=1\\ f_1(n_e,n_m) & \text{otherwise}\end{cases} \quad (3.41)$$

$$u_p=\begin{cases}0.16 & \text{if } Flag_1=1\\ f_2(P_u,n_m) & \text{if } Flag_1=0\end{cases} \quad (3.42)$$

A comparison between the rule-based strategy and the dynamic programming results for urban driving are shown in Fig. 3.11. The throttle command, swashplate angle command, and valve command all closely follow the DDP results.

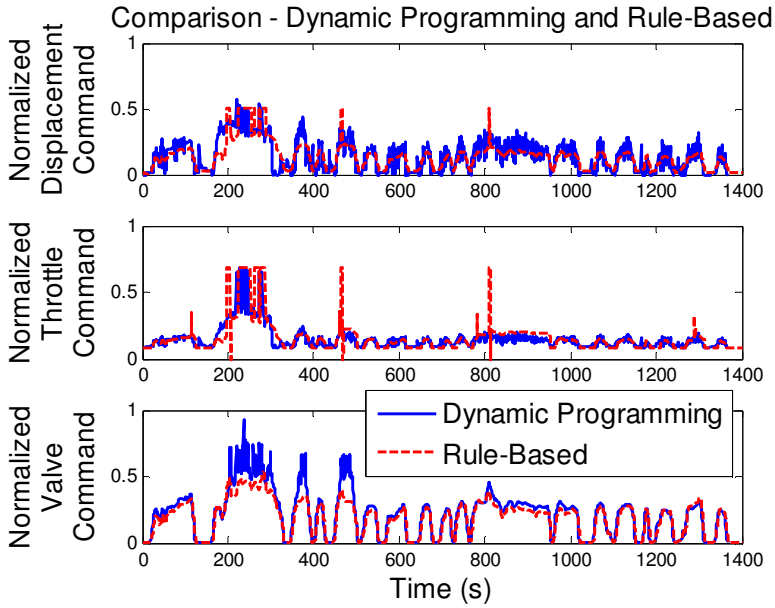

Figure 3.11: Comparison of final rule-based strategy and dynamic programming results over UDDS

3.3.2 Stochastic dynamic programming

One of the drawbacks of DDP is the inability to develop a control law directly from the results since the algorithm assumes the duty cycle and future are completely known. To obtain an implementable control law, a set of rules is developed using trends from the DDP results. However, there is no guarantee that these rules are the optimal solution. To overcome this obstacle, SDP can be used to develop a causal control law. Rather than using a specific duty cycle, probabilities for a driving behavior are used to formulate a control law that is directly implementable from the results [52].

The SDP algorithm requires the transition probabilities to go from one state to another. A Markov chain model is developed for a typical driving behavior by combining numerous standard duty cycles. This is shown in Fig. 3.12, which combines the UDDS, West Virginia Highway, West Virginia Suburban, and West Virginia City duty cycles [50].

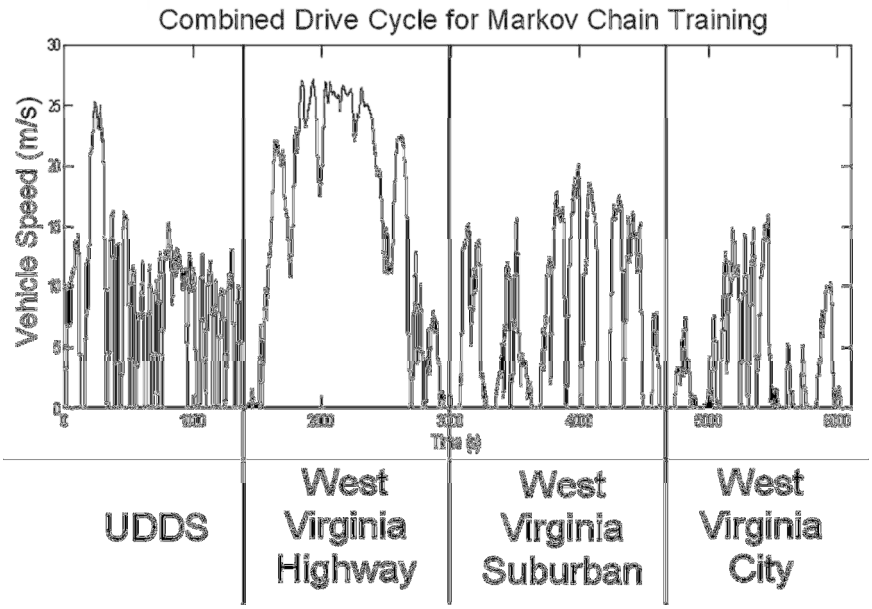

Figure 3.12: Combined duty cycle to determine transition probabilities

The transition probabilities are calculated from this combined duty cycle. The current state is defined by the vehicle velocity and acceleration at the current time step, and the next state

is defined as the acceleration at the next time step. The acceleration at each time step is calculated using backward difference to keep the system causal. The velocity and acceleration at each time step is discretized into 20 uniformly spaced points from the minimum to maximum values using nearest-neighbor approximation. The transition probabilities ($p_{ij,k}$) are then calculated using Eq. 3.43.

$$p_{jk,q} = \frac{\eta_{jk,q}}{\sum_{i=1}^{N_c} \eta_{jk}(i)} \qquad (3.43)$$

In the above equation, j is the index for the current velocity, k is the index for the current acceleration, q is the index for the acceleration at the next time step, and η is the number of counts each occurs in the duty cycle. The probability of the next acceleration being equal to index q given the current velocity index j and the current acceleration index k is equal to the number of times this occurred during the duty cycle divided by the total number of times the duty cycle has a current velocity index of j and a current acceleration index of k. Figure 3.13 shows the transition probability map for a given vehicle speed. Since the peaks lie on the diagonal, if the vehicle is accelerating or decelerating at the current time step, the probability to continue to accelerate or decelerate at the next time step is high.

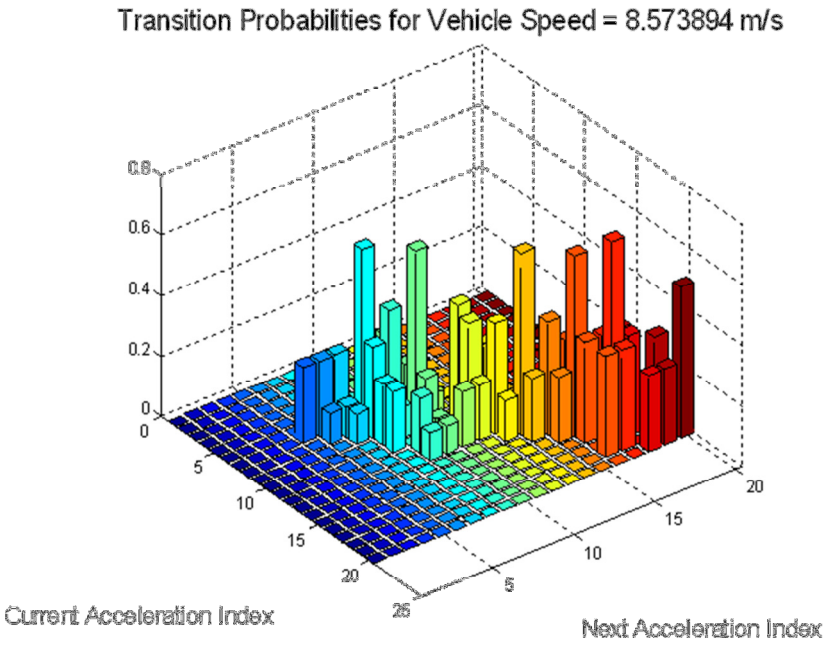

Figure 3.13: Transition probability map for a certain vehicle speed

Once the transition probabilities are known for each vehicle speed, the SDP algorithm is used to determine the optimal action for each state. The discounted policy iteration method is used, which starts with an initial policy and iterates until the solution converges [52]. The first step is the policy evaluation step, which evaluates the current policy and finds the value function, *l*.

$$l_q(j) = \bar{c}\left(j, \mu_q(j)\right) + \varepsilon \sum_{k=1}^{|S|} p\left(j, \mu_q(j), k\right) \cdot l_q(k) \qquad (3.44)$$

ε is the discount factor, which is less than 1. The meaning of this discount factor is that future costs do not matter as much as the same costs incurred at the present time. The lower this discount factor, the lower the importance of the future costs. *S* is the total number of states and $\bar{c}$ is the average cost at state *j*, which is given by Eq. 3.45.

$$\bar{c}\left(j, \mu_q(j)\right) = \sum_{k=1}^{|S|} p\left(j, \mu_q(j), k\right) \cdot c\left(j, \mu_q(j), k\right) \qquad (3.45)$$

The cost function *c* is the fuel consumption to go from state *j* to state *k* using control policy μ_q at state *j*. The fuel consumption is a function of the throttle command and the engine speed.

$$c\left(j, \mu_q(j), k\right) = fuel\left(\gamma, \omega_e\right) = \max\left(K_f \omega_e \frac{\theta}{\theta_{\max}}, 0.00043\right) \Delta t \qquad (3.46)$$

The value function is solved at each state using the set of linear equations given in Eq. 3.44. Once the value function at each state is known, the policy improvement step is performed according to Eq. 3.47.

$$\mu_{q+1}(j) \in \arg\min_{u \in U(j)} \left[\bar{c}\left(j, \mu_q(j)\right) + \varepsilon \sum_{k=1}^{|S|} p\left(j, u, k\right) \cdot l_q(k) \right] \qquad (3.47)$$

The policy evaluation and policy improvement steps repeat until the policy for each state between iterations is the same, which is the optimal policy. The end result is a lookup table

which inputs the current vehicle velocity, acceleration, and accumulator oil volume and outputs the control decision.

The outputs and control variables are discretized uniformly between their minimum and maximum values. The output variables are the upstream pressure and motor speed. The control variables are the swashplate angle command, engine speed, and valve command. The values of upstream pressure are discretized from 6 MPa to 19 MPa in 1 MPa increments. The values for the swashplate angle are discretized between 0 rad and 0.30 radians in 0.01 radian increments. The engine speed is discretized from 75 rad/s to 185 rad/s in 5 rad/s increments. The throttle command is then found using a map of engine speed and engine torque. The valve command is determined using a feedback PI controller on motor speed to improve speed tracking. The discount factor is set to 0.95.

3.4 Experimental Validation

To validate the performance of the control strategies experiments are conducted using the UDDS as a reference trajectory. When implementing the rule-based, SDP, and MPC EMS's onto the AEVPS there are several hardware and computing constraints which had to be satisfied. For both the rule-based and SDP strategies a proportional plus integral feedback controller is employed to regulate the valve opening and ensure tracking of the desired hydraulic motor speed. The error signal which is sent to this controller is the difference between the desired and measured motor speed. Due to bandwidth limitations of the electronic proportional valve, a proportional gain of 0.01 and an integral gain of 0.05 are selected. In addition, due to noise in the motor speed measurement, a first order low pass filter with a cut off frequency of 10 Hz is applied to this signal before calculating the tracking error. Figure 3.14 through Fig. 3.16 show comparisons of the simulated and experimental outputs, inputs, and engine response respectively for the UDDS. The outputs and inputs of the two cases agree except the average throttle command in the experimental case is less than that of the simulation case. This is due to the engine idle speed being slightly less in the experimental case. Since the engine is operating at a lower mean throttle command the accumulator is discharged at a greater rate to meet the operator's demand. However, the engine operations, shown in Fig. 3.16, for the two cases are

very similar. Using the rule-based EMS, the simulated and experimental fuel consumptions are within 5% of each other (Urban: simulation: 1.099 kg, experiment: 1.058 kg).

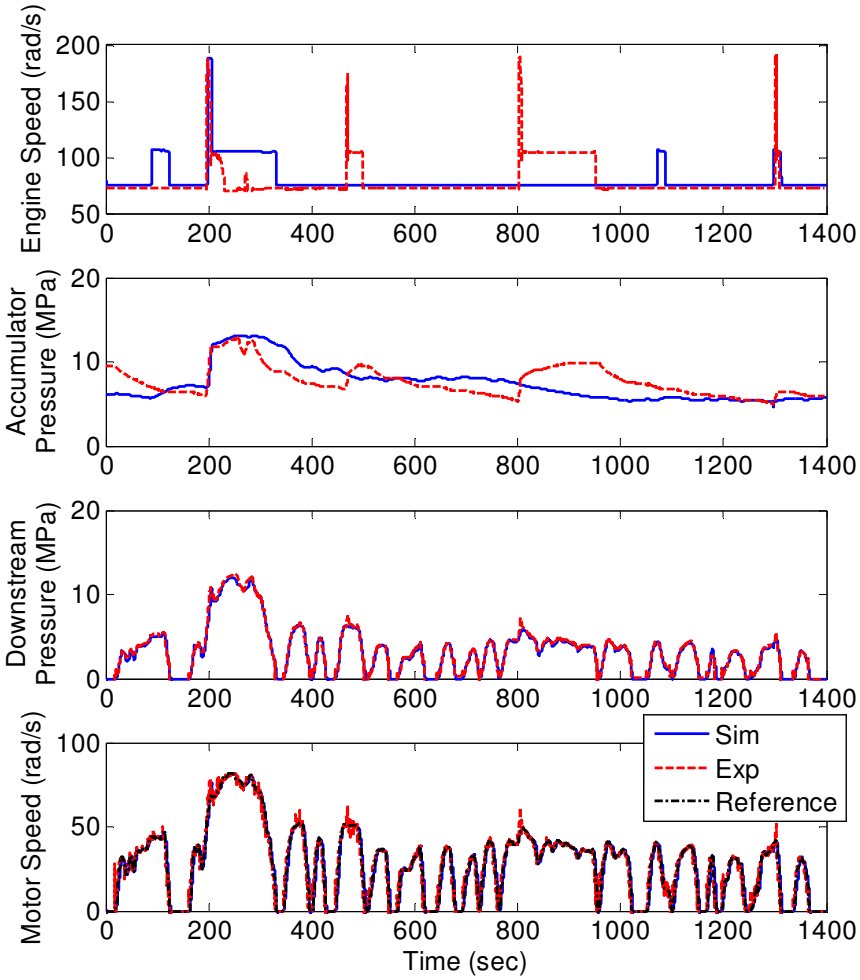

Figure 3.14: Comparison of simulated and experimental SHHV outputs for the rule-based strategy

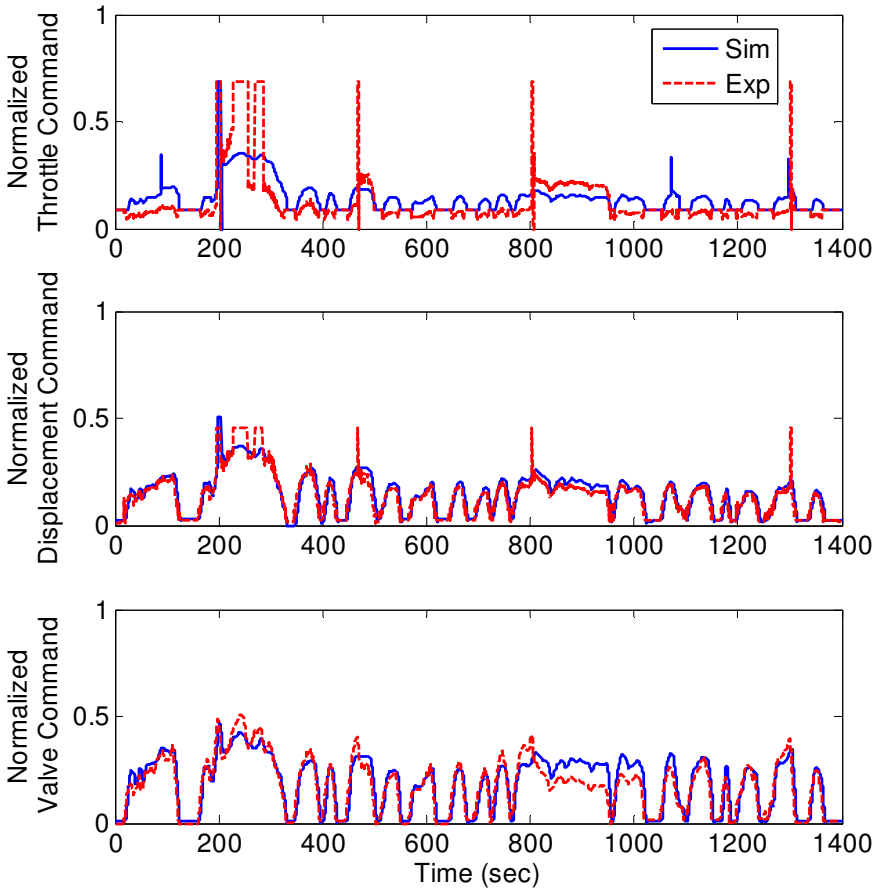

Figure 3.15: Comparison of simulated and experimental SHHV inputs for the rule-based strategy

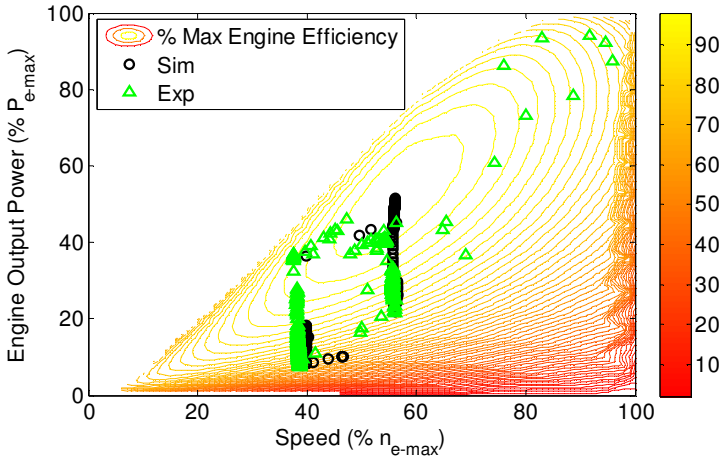

Figure 3.16: Comparison of simulated and experimental engine response for the rule-based strategy, markers denote engine operating point

When implementing the SDP strategy on the AEVPS a discrete update rate of 10 Hz is used for the look-up tables to ensure time for the controller to complete the interpolation between update steps. Furthermore, the high frequency behavior of the DDP solution could not be implemented in the AEVPS hardware due to bandwidth limitations of the actuators. Therefore, the outputs of the tables resulting from the SDP formulation are passed through first order low pass filters with a cut off frequency of 0.5 Hz to prevent the high frequency response. Imposing these limitations did affect the tracking performance of the controller but are ultimately necessary for physical implementation. From Fig. 3.17 and Fig. 3.18 one can see that there is strong agreement in the both the outputs and inputs for the UDDS simulation and experimental cases. The most significant difference is that the engine speed is able to fall below the idle speed in the experimental case. This discrepancy can also be seen in the engine operating points shown in Fig. 3.19. Given the strong agreement between the outputs and inputs, it is not surprising that the simulated and experimental fuel consumptions are within 5% of each other (Urbana: simulation: 1.008 kg, experiment: 0.975 kg).

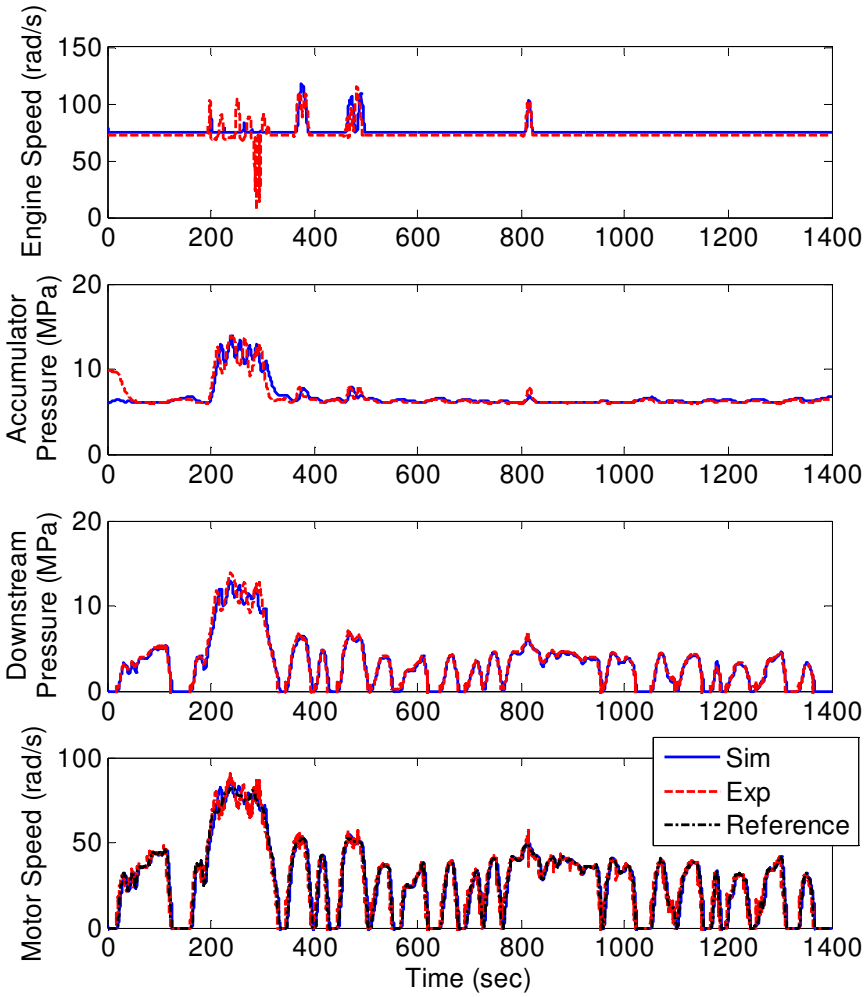

Figure 3.17: Comparison of simulated and experimental SHHV outputs for the SDP strategy

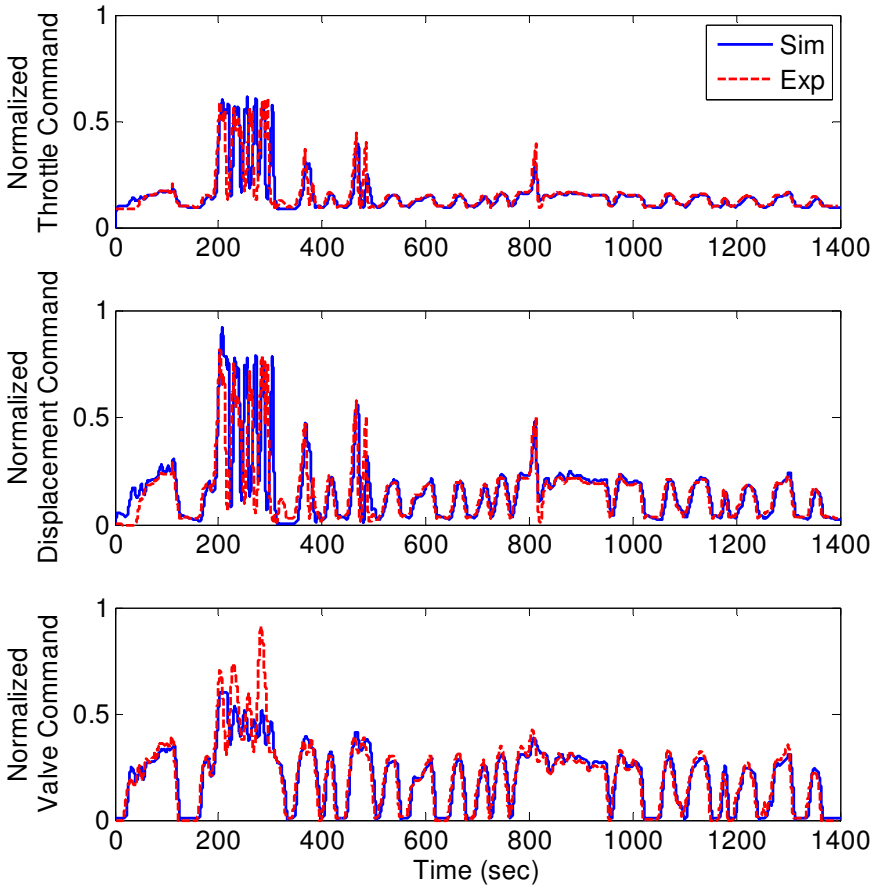

Figure 3.18: Comparison of simulated and experimental SHHV inputs for the SDP strategy

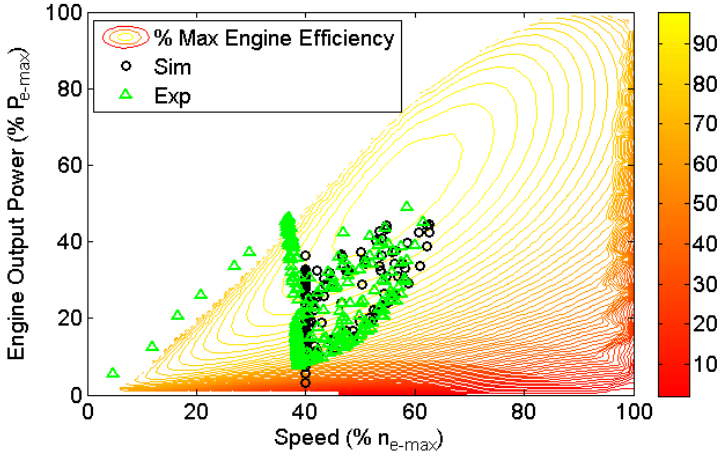

Figure 3.19: Comparison of simulated and experimental engine response for the SDP strategy, markers denote engine operating point

The MPC based EMS is implemented with an update rate of 1 Hz and a prediction horizon length of 5 steps. The update rate and horizon length are chosen such that they allow for real-time execution while balancing the step size against the prediction horizon length. A 10 second dwell time is used in the supervisory logic along with $\tau_{threshold}$ equals 30 Nm, $P_{threshold}$ equals 6.5 MPa, and v equals 1.9 V. The set points for the ON mode objective function are $\omega_{e,des1}$ = 76 rad/sec (close to emulated engine idle speed) $u_{e,des1}$ = 96.8 Nm (max emulated engine torque), and $P_{u,des}$ equal to 7 MPa to ensure that the upstream pressure does not fall below the accumulator precharge pressure (5.17 MPa). For the OFF mode, $\omega_{e,des2}$ is set below the idle speed and $u_{e,des2}$ is set to 0 Nm. In addition, the control signals are passed through first order low pass filters with a cut off frequency of 1 Hz to smooth the inputs and prevent violations of actuator bandwidth limitations. The response of the hybrid powertrain with MPC is nearly identical for simulation and experimental response over the UDDS as can be seen in Fig. 3.20 through Fig. 3.22. The most noticeable difference is that the accumulator loses charge faster in the experimental case due to un-modeled losses. For this EMS the simulated fuel consumption is 1.300 kg and the experimental fuel consumption is 1.340 kg. Again, the two fuel consumptions are within 5% of each other.

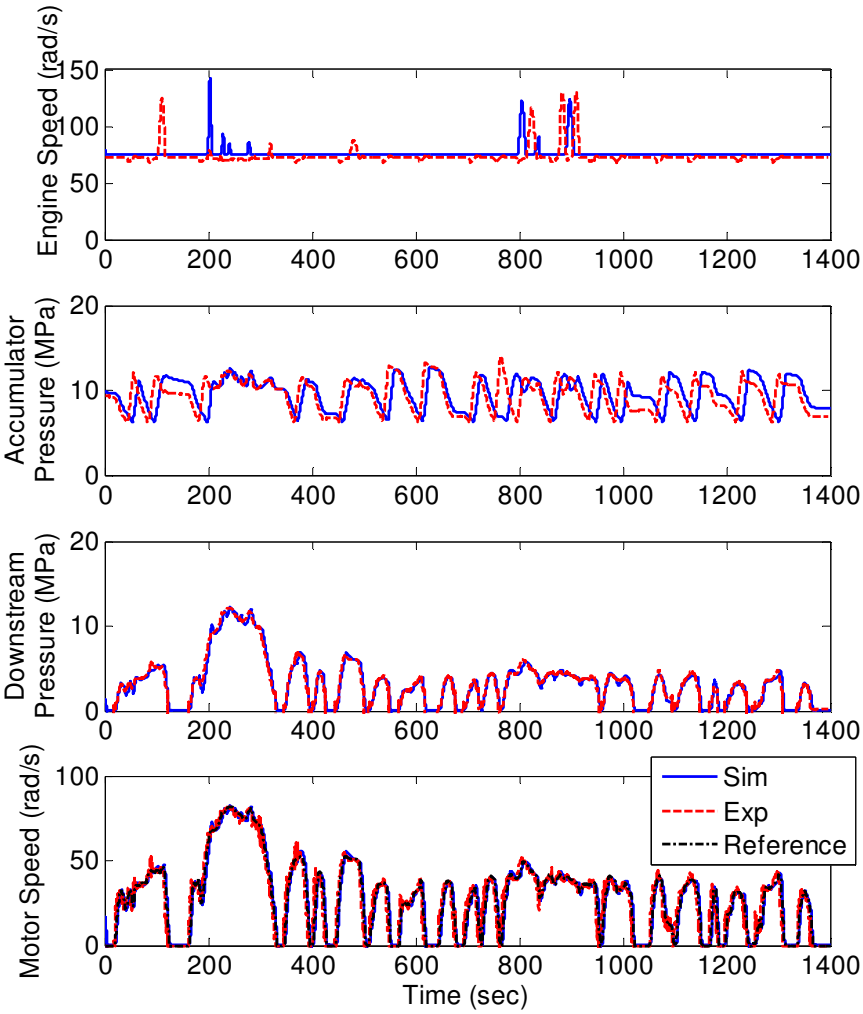

Figure 3.20: Comparison of simulated and experimental SHHV outputs for the MPC strategy

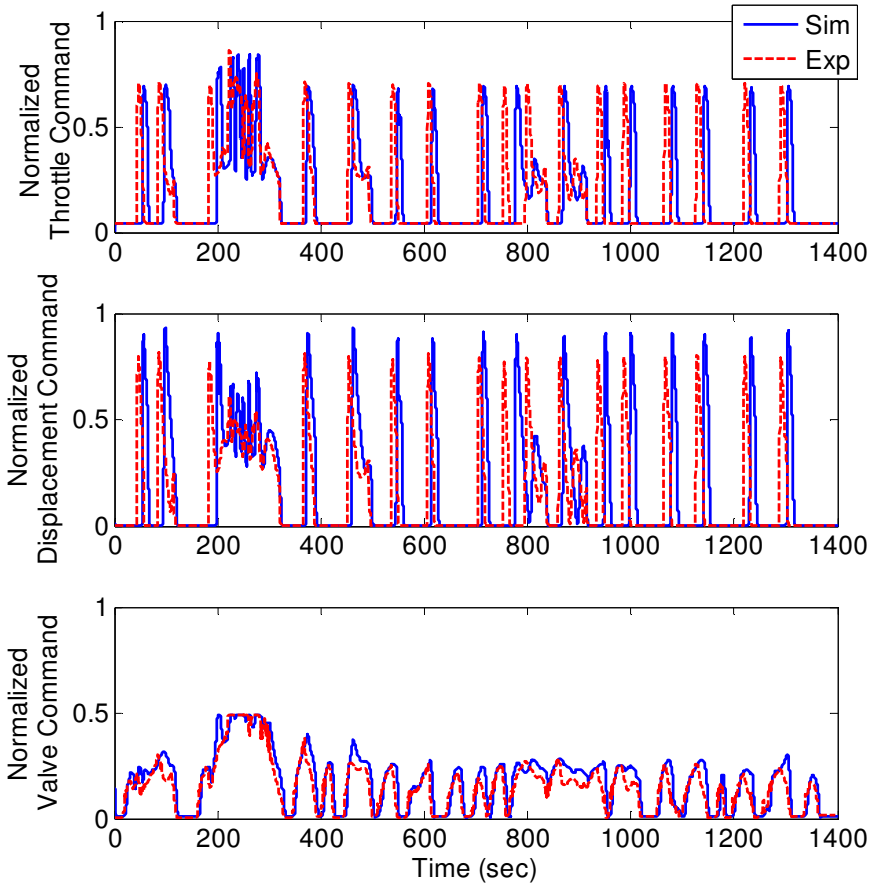

Figure 3.21: Comparison of simulated and experimental SHHV inputs for the MPC strategy

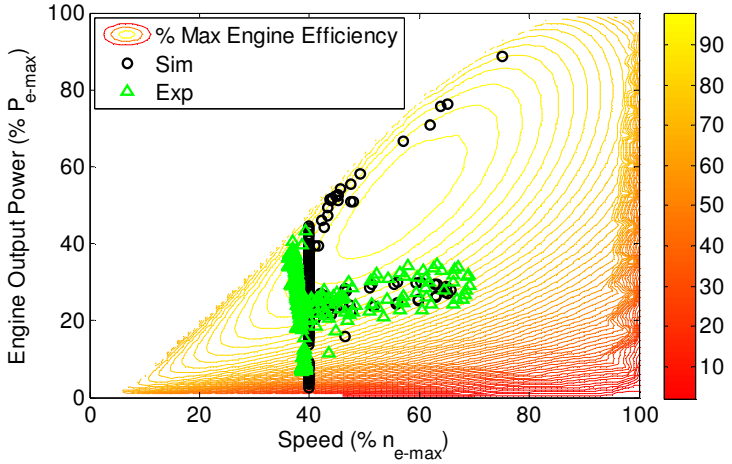

Figure 3.22: Comparison of simulated and experimental engine response for the SDP strategy, markers denote engine operating point

3.5 Comparison Study

A simulation study is presented for all three of the experimentally validated EMS's described in this chapter to evaluate the behavior of these strategies in the face of uncertainty in the duty cycle and system model. The effect of these uncertainties will be explored by varying the duty cycle and physical parameters within the system model and then evaluating the resulting fuel consumption and tracking error. For the duty cycle variation study urban and highway driving scenarios are considered and each control strategy is simulated over 100 duty cycles of each scenario. For the parameter variation study five powertrain parameters are varied (variable displacement pump flow gain, upstream hose loss coefficient, downstream hose loss coefficient, motor displacement, and vehicle viscous friction). The results of these studies will be used to compare the relative performance of each EMS design method to duty cycle and system uncertainty. From these comparisons, one can determine when one design methodology would be advantageous depending on the knowledge available.

3.5.1 Duty cycle variation

When evaluating the effect of duty cycle uncertainty on the performance of the EMS urban and highway driving scenarios are considered. To allow for a rich set of duty cycles urban and highway duty cycles are generated from the transition probability maps of the UDDS and West Virginia Highway cycles. A uniform random number generator is used with the transition probability matrix for the current state to calculate the acceleration at the next time step. This is repeated until a specified time length is reached. An example of an urban duty cycle generated using this method is shown in Fig. 3.23. In this way, the transition probabilities for all generated duty cycles are the same as the root driving scenario, ensuring that each duty cycle is representative of the same driving scenario while allowing for different time sequences of desired vehicle speed.

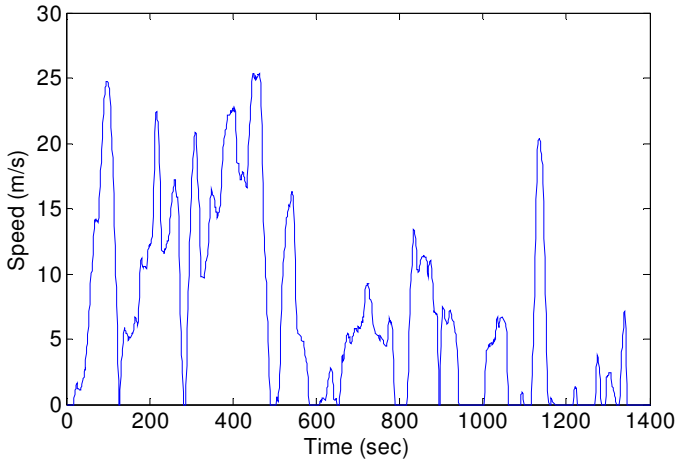

Figure 3.23: Urban duty cycle generated from UDDS transition probability map

To quantify the performance of each EMS for a duty cycle two metrics are considered: the fuel consumption and tracking error. The fuel consumptions estimated in simulation for each EMS are normalized with respect to the fuel consumption of a non-hybrid vehicle for the same duty cycle. The non-hybrid vehicle is modeled using the same engine and vehicle loads as the SHHV model with a lossless three gear transmission (7:1, 3.5:1, 1.75:1) and a speed dependent gear shifting policy given by Eq. 3.48– Eq. 3.50. In simulation, the non-hybrid vehicle consumed 1.51 kg of fuel over the UDDS. Note that the transmission ratios and shifting policy were chosen to ensure the vehicle could satisfy all speed demands but were not optimized to minimize fuel consumption. This provides a common reference point to assess the relative improvement of each control strategy for all duty cycles. The other metric for determining how well a strategy performs is its ability to track the desired reference. The root mean square (RMS) tracking error is a measure of how closely the vehicle speed agrees with the desired vehicle speed profile.

$$Flag_3(t) = \begin{cases} 2, & \text{For } \omega_m > 24.2 \\ 1, & \text{For } \omega_m < 21.0 \\ Flag_1(t-1) & \text{Else} \end{cases} \qquad (3.48)$$

$$Flag_4(t)=\begin{cases}1, & \text{For } \omega_m > 48.4\\ 0, & \text{For } \omega_m < 41.9\\ Flag_2(t-1) & \text{Else}\end{cases} \quad (3.49)$$

$$Gear(t)=\begin{cases}7, & \text{If } Flag_3+Flag_4=1\\ 3.5, & \text{If } Flag_3+Flag_4=2\\ 1.75, & \text{If } Flag_3+Flag_4=3\end{cases} \quad (3.50)$$

The results of the urban driving study are summarized in Fig. 3.24 and Fig. 3.25. From Fig. 3.24, one can clearly see that the SDP approach achieved the greatest improvement in fuel economy with a small variance between urban duty cycles. This is what one would expect since the SDP strategy achieved the lowest fuel consumption in experimental validation and it is derived using the same transition probability maps. The MPC and Rule-based strategies achieved similar levels of improvement with the rule-based approach having the largest variance. Since the rule-based strategy is tuned for a single duty cycle it is not surprising that its performance would be most sensitive to variations in the urban driving scenario. When comparing the tracking errors given in Fig. 3.25, one finds that the rule-based EMS achieved the lowest tracking error with the SDP strategy having the largest tracking error and the MPC being between the two. The large tracking error of the SDP solution is a consequence of its sensitivity to errors in the model used for offline optimization. One of the limitations of this approach is that there is no mechanism for fine tuning the lookup table based on simulation and experimental response. Therefore, to improve the control performance the SDP algorithm would need to be rerun with progressively more detailed models (including limitations due to hardware and computational constraints) until the desired tracking performance is achieved. Based on the results shown in Fig. 3.24 and Fig. 3.25, the rule-based strategy is the best compromise between fuel improvement and tracking.

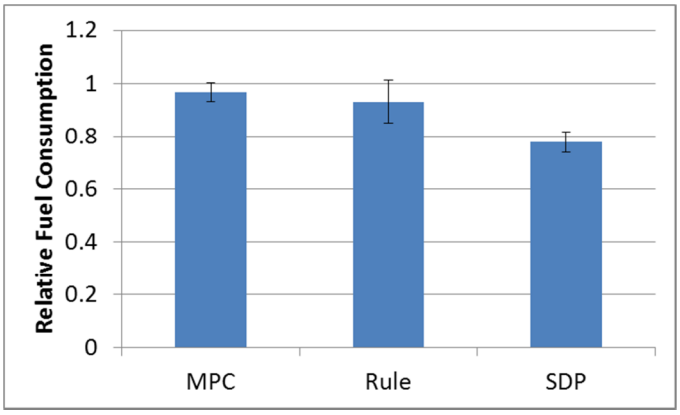

Figure 3.24: Mean fuel consumption relative to non-hybrid for urban driving scenarios, error bars are ±1 standard deviation

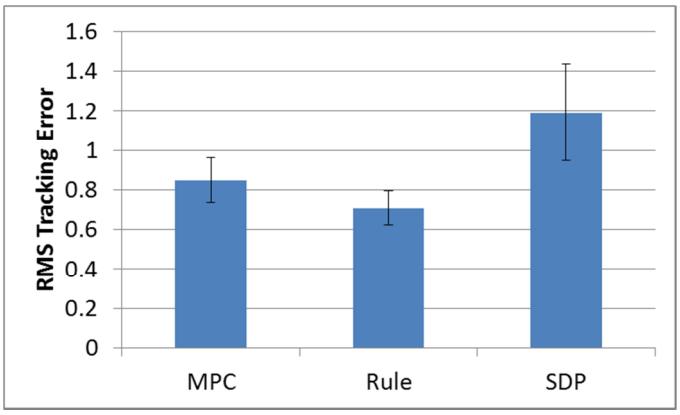

Figure 3.25: Mean RMS tracking error for urban driving scenarios, error bars are ±1 standard deviation

Figure 3.26 and Fig. 3.27 summarize the performance of the EMS's for highway driving scenarios. Like the urban driving case, the SDP strategy achieves the lowest average fuel consumption but with the poorest mean tracking performance. The rule-based strategy achieved the worst fuel economy performance with tracking performance nearly as bad as the SDP solution. This overall loss in performance for the rule-based strategy is expected since it is tuned for an urban driving profile. Finally, the MPC approach has the best tracking performance and superior fuel reduction than the rule-based strategy. Furthermore, it had the lowest variance in performance for both metrics of all there EMS's. For the highway driving study, the MPC approach demonstrated the best overall performance.

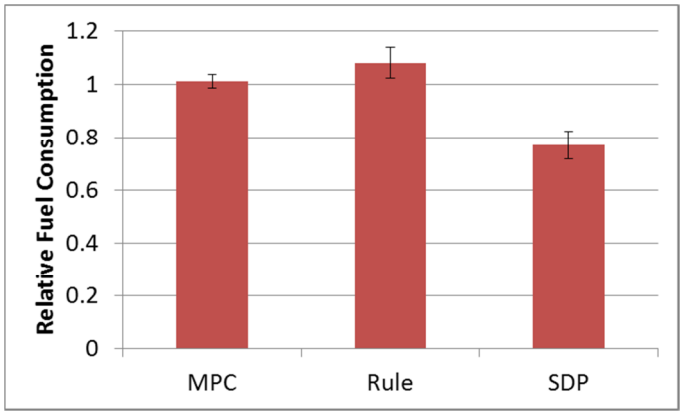

Figure 3.26: Mean fuel consumption relative to non-hybrid for highway driving scenarios, error bars are ±1 standard deviation

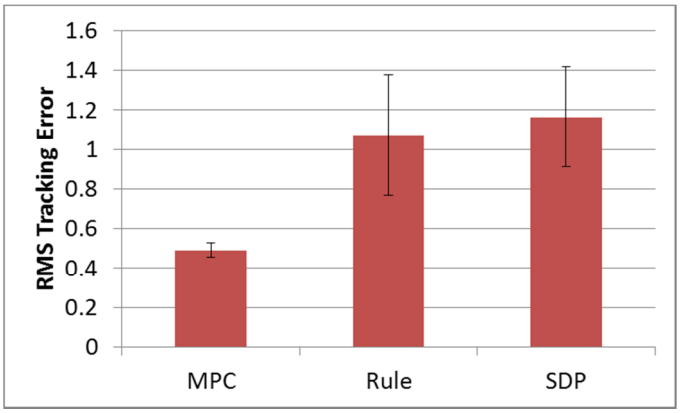

Figure 3.27: Mean RMS tracking error for highway driving scenarios, error bars are ±1 standard deviation

Finally, to determine which strategy would be the most robust in a general driving scenario, the two data sets are combined and those results are summarized in Fig. 3.28 and Fig. 3.29. For these results one can see that the SDP strategy offers the greatest improvement in fuel economy but with large tracking errors. This suggests that for this strategy to be implemented effectively there must be strong agreement between the models used for control development and the physical system. The rule-based and MPC solution offer similar fuel economy performance but the MPC achieved the lowest overall tracking error. From this duty cycle study it can be seen that the rule-based strategy is most sensitive to variations in duty cycle and that its performance cannot be guaranteed for duty cycles which differ from its design cycle. The MPC solution is the most robust to duty cycle variation, giving similar performance for highway and

urban driving. This is as one would expect since duty cycle information is not used explicitly in the control design.

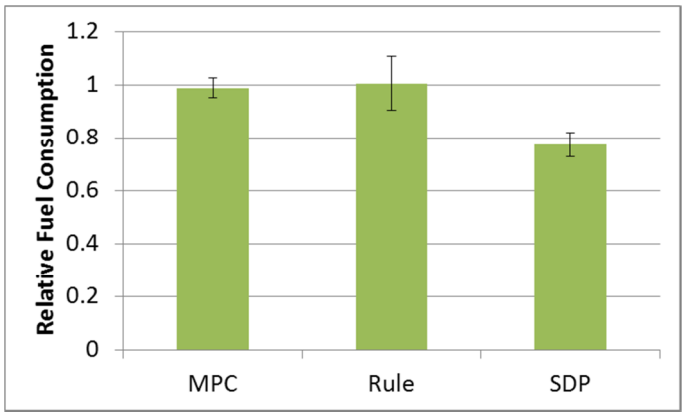

Figure 3.28: Mean fuel consumption relative to non-hybrid for combined data sets, error bars are ±1 standard deviation

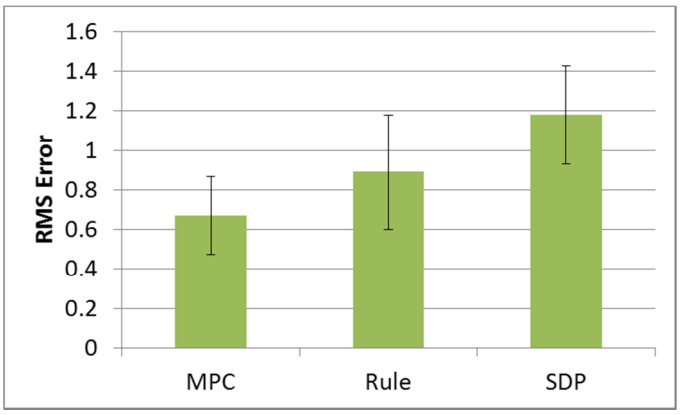

Figure 3.29: Mean RMS tracking error for combined data sets, error bars are ±1 standard deviation

3.5.2 Parameter variation

For the parameter variation study 5 parameters are varied by ±10% in increments of 5% (cases 1-4: variable displacement pump flow gain, cases 5-8: upstream hose loss coefficient, cases 9-12: downstream hose loss coefficient, cases 13-16: motor displacement and cases 17-20: vehicle viscous friction). In each case the percent change in fuel consumption and percent change in RMS tracking error are calculated relative to the case without parameter variation and all tests are done over the UDDS. The results of this study are captured in Fig. 3.30 and Fig.

3.31. From these results one can see that the fuel consumptions of the MPC and SDP approaches are insensitive to parameter variations, with a 10% change in parameters causing less than a 10% change in fuel consumption. However, the rule-based strategy showed greater sensitivity to parameter variations. For a 10% decreases in pump flow gain the rule-based strategy demonstrated greater than 15% change in fuel consumption. From the RMS tracking error results shown in Fig. 3.31, the MPC approach is found to be the least sensitive to changes in parameters. The rule-based strategy exhibited percent changes in tracking error similar in magnitude to the change in parameters for increases in pump flow gain and decreases in motor displacement. Finally, the tracking performance of the SDP approach is the most sensitive to changes in system parameters with a 10% decrease in motor displacement causing nearly a 200% increases in tracking error. This observation is consistent with the behavior that is found in the duty cycle study. Since a system model is assumed in the SDP formulation, an accurate model is critical to achieving good tracking performance.

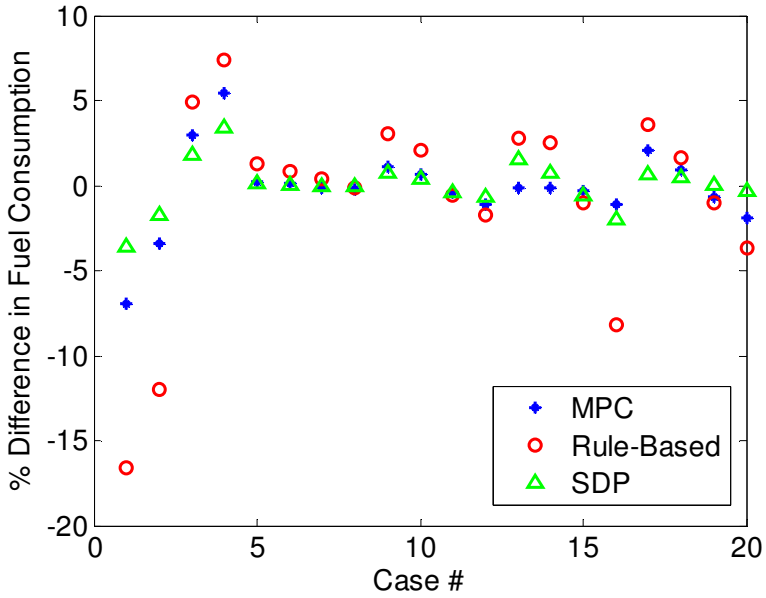

Figure 3.30: Variation in fuel consumption as a result of changes in system parameters

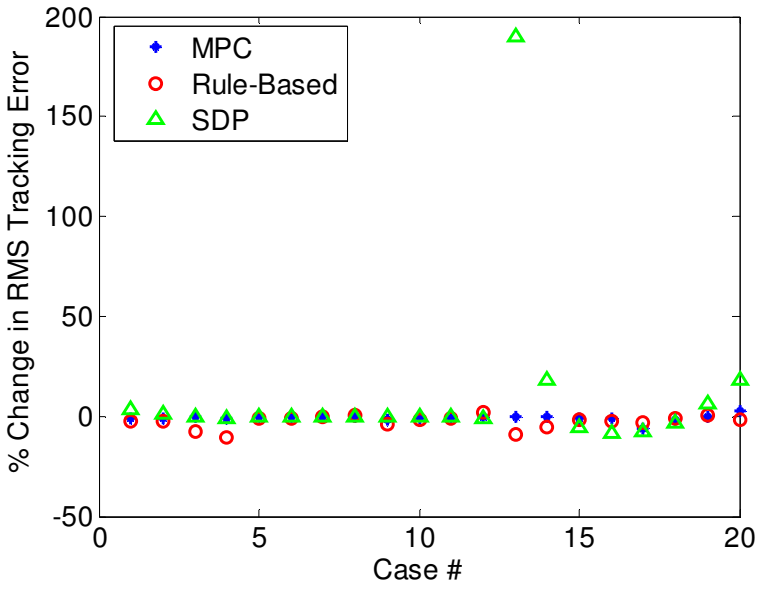

Figure 3.31: Variations in RMS tracking error as a result of changes in system parameters

3.6 Concluding Remarks

Each of the three design methods responded differently to variations in the duty cycle and model parameters and a tradeoff between optimality and robustness was observed. The SDP strategy is the most direct implementation of the dynamic programming solution. As such it achieved the lowest fuel consumption in all trials but was also the most sensitive. Since this method yields a lookup table which can be implemented directly, care must be taken to ensure that the model and duty cycles used to define the optimization problem accurately reflect the physical system. The rule-based strategy was found to be less sensitive to variations because it could be further tuned to ensure satisfactory performance over a desired duty cycle. This tuning process increased the fuel consumption but improved robustness. However, if the commanded duty cycle is not represented in the tuning process then large tracking errors and poor performance may result, as was the case with the highway driving cases. Finally, the MPC strategy is the most robust design method studied. Its performance is consistent for both urban and highway driving. However, it is not able to achieve the fuel savings of the SDP strategy. From the simulation study, the following qualitative guidelines can be concluded:

- The SDP strategy is best suited for applications in predictable environments with well-defined models.

- The rule-based method is best suited for applications with known trajectories, like city buses, where one has strong confidence in duty cycle predictions.
- For highly uncertain applications, such as a passenger vehicle, the MPC strategy is the most reliable.

Chapter 4 Thermal Management Study

In the previous case study, it was demonstrated how the proposed method for designing energy management strategies (EMS's) can use a first principle analysis of system dynamics and no knowledge of the future duty cycle to produce a robust control strategy. This EMS was found to give consistent fuel consumption for different duty cycles and enforced modeling uncertainty. In this chapter the operation of a refrigerated delivery truck with thermal energy storage (TES) will be optimized. System ID will be used to derive the prediction model and complete knowledge of the duty cycle will be assumed. This will demonstrate the generalizability of the proposed method by considering a different type of system with different constraints on system modeling and preview. Refrigerated transport systems are used throughout the world to deliver food and other temperature-sensitive goods. As of 2002 there were over 1 million refrigerated road vehicles and 400,000 refrigerated containers in use [53]. A refrigerated transport system uses a vapor compression cycle (VCC) system to either cool or heat an enclosed space so that a desired temperature can be maintained. For refrigerated road vehicles the, VCC system is either powered by the vehicle's engine or an auxiliary diesel engine [31]. Therefore, the efficiency with which one is able to maintain the desired volume temperature directly affects the fuel consumption and emissions for these vehicles.

Like a hybrid vehicle, the efficiency of refrigerated transport systems can be improved by including thermal energy storage [31]. The use of phase change materials (PCM) as a mechanism for storing thermal energy has been explored in building systems and has the potential to yield significant energy savings [29]. However, their use has not been studied in detail for mobile applications. As in the case of hybrid vehicles, adding a storage device increases the complexity of the system architecture and necessitates an EMS. Previous control

studies of systems with thermal storage have focused on large building systems [38] where the time scale for changes in thermal demand are much slower than the time scale in mobile applications. Therefore the challenge of controlling a refrigerated transport system with thermal storage is analogous to that of the electric and hydraulic hybrid vehicles [22], [2]. However, there is no mechanism for regenerating energy in a refrigeration system. Therefore, any efficiency improvements must be made by using the storage capability to operate the actuators more efficiently. In addition, the duty cycle of a refrigerated transport vehicle may be known in advance, and so this information can possibly be utilized by the EMS.

4.1 Parallel Vapor Compression System

For this study, a parallel configuration of a vapor compression system with storage is considered. The parallel vapor compression cycle (PVCC) has a TES device in parallel with the traditional VCC, and the storage device is equipped with its own fan for regulating heat transfer between the TES and the environment. A schematic of the PVCC architecture is shown in Fig. 4.1. There are four inputs to this system: compressor speed (u_ω), electronic expansion valve opening (u_{eev}), flow split for an ideal flow control valve (u_{IV}), and air mass flow rate across the TES (u_{air}). This system is capable of meeting a desired cooling capacity in 3 ways:

- Evaporator Only
- TES Only
- Combined Evaporator and TES

Within the PVCC there is a traditional VCC circuit. This circuit is composed of four main components: the compressor, the condenser, an expansion valve, and the evaporator. The compressor is used to compress superheated vapor refrigerant to a higher temperature and pressure. Then the refrigerant passes through the condenser and rejects energy to ambient air passing over the condenser coils. Refrigerant then exits the condenser as saturated liquid and expands isenthalpically through a valve causing a drop in pressure and temperature. Here, an electronic expansion valve (EEV) is considered as the expansion device so that the pressure drop can be electronically controlled. After passing through the expansion valve, two-phase refrigerant enters the evaporator where it exchanges heat with air passing over the evaporator coils. Finally, refrigerant exits the evaporator as superheated vapor and the cycle is repeated.

For the VCC circuit model, only the dynamics of the compressor, condenser, EEV, and evaporator are captured. For the heat exchangers (condenser and evaporator), a moving boundary model is used. Detailed nonlinear models of these components exist in the literature; for details on how these components are modeled, the reader is referred to [54] and [55].

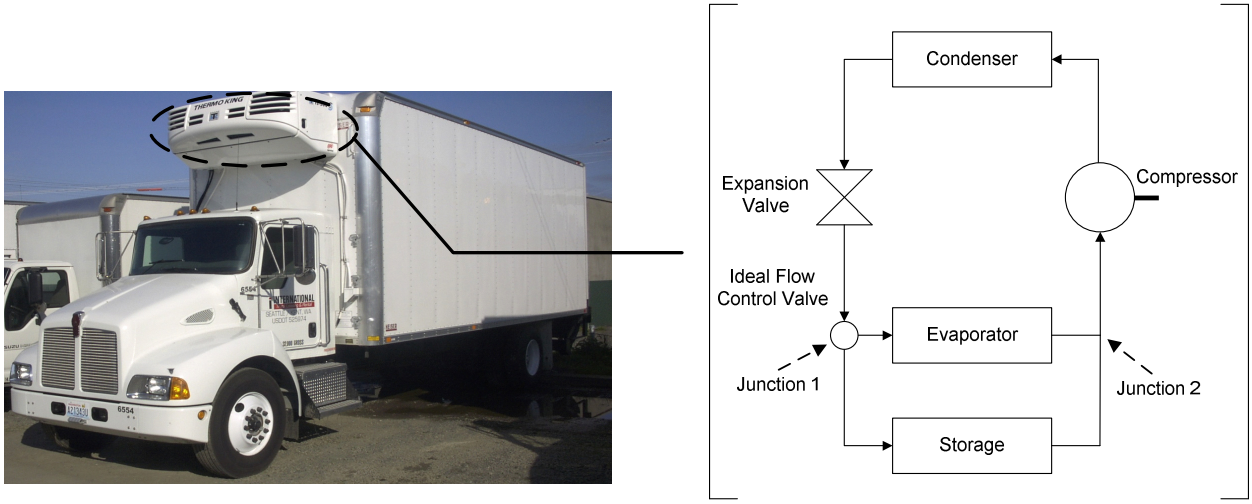

Figure 4.1: Schematic of the Parallel Vapor Compression Cycle

To integrate the thermal storage device into the VCC circuit model, several continuity equations are needed. For the storage device, it is assumed that the mass flow rate of refrigerant entering and exiting the device are equal and that the pressures at the inlet and outlet of the storage device and evaporator are equal. At junction 1 (see Fig. 4.1) an ideal flow control valve is used to regulate the percent of refrigerant mass flow which is directed to the evaporator. This device is ideal since there are no losses associated with the flow division. From mass flow and energy conservation at Junction 1, one arrives at the constraints given by Eq. 4.1 through Eq. 4.4. Similarly, conservation equations are applied at Junction 2 (see Fig. 4.1) to derive the constraints given by Eq. 4.5 and Eq. 4.6.

$$\dot{m}_{ev,in} = u_{IV} \cdot \dot{m}_{eev} \qquad (4.1)$$

$$\dot{m}_{s,in} = (1 - u_{IV}) \cdot \dot{m}_{eev} \qquad (4.2)$$

$$u_{IV} \in [0,1] \qquad (4.3)$$

$$h_{s,in} = h_{ev,in} = h_{eev} \tag{4.4}$$

$$\dot{m}_{ev,out} + \dot{m}_{s,in} = \dot{m}_{comp} \tag{4.5}$$

$$h_{comp} = \frac{h_{e,out} \cdot \dot{m}_{e,out} + h_{s,out} \cdot \dot{m}_{s,in}}{\dot{m}_{comp}} \tag{4.6}$$

Here $\dot{m}$ is air mass flow rate, h is specific enthalpy, subscript *e,in* denotes refrigerant at the evaporator inlet, *e,out* denotes refrigerant at the evaporator outlet, subscript *s,in* denotes refrigerant at the storage inlet, *s,out* denotes refrigerant at the storage outlet, subscript *comp* denotes refrigerant at the compressor inlet, and subscript *eev* denotes refrigerant at the exit of the EEV. When deriving Eq. 4.5 and Eq. 4.6, it is assumed that refrigerant mass flow rates into and out of the storage device are equal.

Many approaches have been developed for modeling PCM's, including very sophisticated finite element analysis [56], [57]. Here, a model of the TES is presented which is detailed enough to capture the internal dynamics of the moving phase change boundary but computationally compact enough for control analysis. For full details of how this model is derived and validation of the storage model the reader is referred to [58]. This heat exchanger is modeled with a concentric tube inner geometry and a bank of tubes exterior geometry. A schematic of the exterior geometry is shown in Fig. 1.3 and the inner geometry is shown in Fig. 4.2. Each tube is decomposed into three fluid regions; the most internal region is refrigerant which is used to cool the PCM. Moving radially outward, the next region is the PCM. This region is discretized into N_s nodes. The final fluid region is the ambient air. Each of the three fluid regions are separated by walls which have their own thermal inertia.

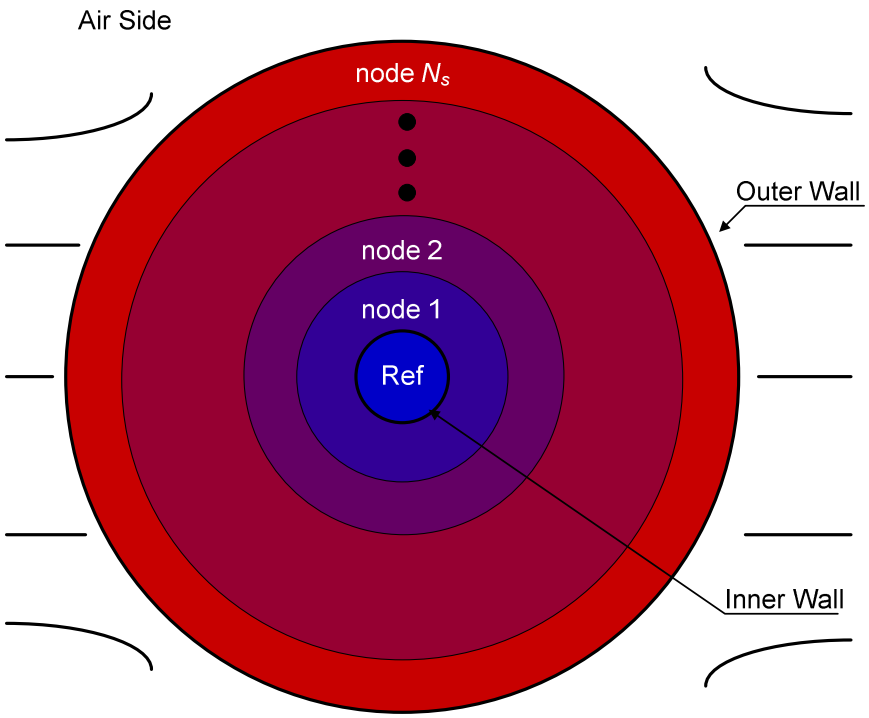

Figure 4.2: Schematic of the concentric tube thermal energy storage system in which node *i* represents a control volume

This model mathematically describes the properties of the three fluid regions in the TES and the heat transfer interactions among these regions. Similar to the evaporator model, the refrigerant section is described by the lumped parameter, moving boundary, switched state space method developed in [55]. For the PCM region, an enthalpy formulation is used to describe the mechanism by which the material undergoes a change of phase. This method is used because the rate of change of enthalpy is a continuous state over a phase change. The boundary conditions for the inner and outer walls respectively are given by Eq. 4.7 and Eq. 4.8.

$$k_{TH}\nabla T\big|_{r=r_{inner}} = \alpha_{inner}\left(\Delta T\right) \qquad (4.7)$$

$$k_{TH}\nabla T\big|_{r=r_{outer}} = \alpha_{outer}\left(\Delta T\right) \qquad (4.8)$$

Here T is temperature, α is wall convection coefficient, and k_{TH} is the thermal conductivity. For the j^{th} nodes of the PCM, a one dimensional radial enthalpy distribution is assumed, and control volumes are used to derive the governing equations. The enthalpy of each

node is given by Eq. 4.9. The radial temperature derivatives are given through the finite difference form shown in Eq. 4.10 and Eq. 4.11 where ρ is the PCM density, V is the PCM volume within a node, A is the surface area at a nodal boundary, Δr is the radial spacing between nodes, and the subscripts *inner* and *outer* refer to the inner and outer boundaries, respectively.

$$\frac{dh_j}{dt} = \left(\left(\frac{k_{TH}}{\rho_{PCM} V_{PCM}} \right) \cdot \left(A_{inner} \left(\frac{dT}{dr} \right)_{j-1} + A_{outer} \left(\frac{dT}{dr} \right)_{j+1} \right) \right)_j \quad (4.9)$$

$$\left(\frac{dT}{dr} \right)_{j-1} = \frac{T_{j-1} - T_j}{\Delta r} \quad (4.10)$$

$$\left(\frac{dT}{dr} \right)_{j+1} = \frac{T_{j+1} - T_j}{\Delta r} \quad (4.11)$$

From the nodal enthalpies, temperatures are found from tabular data, assuming constant pressure, and algebraic relations are used to relate temperatures to the solidification fraction of each node. This allows one to accurately compute the location of the solidification front.

The boundary equations for the inner and outer walls are formulated using a similar process and are given by Eq. 4.12 and Eq. 4.13.

$$\frac{dT_{wall,si}}{dt} = \frac{4k_{TH} \cdot r_{wall,si}}{\rho \cdot C_p \left(r_{wall,si}^2 - r_{wall,r}^2 \right) \Delta r} \left(T_1 - T_{wall,si} \right) + \frac{2 \cdot h_{wall,si} \cdot r_{wall,r}}{\rho \cdot C_p \left(r_{wall,si}^2 - r_{wall,r}^2 \right)} \left(T_{wall,r} - T_{wall,i} \right) \quad (4.12)$$

$$\frac{dT_{wall,a}}{dt} = \frac{4k_{TH} \cdot r_s}{\rho \cdot C_p \left(r_{wall,a}^2 - r_{wall,so}^2 \right) \Delta r} \left(T_{N_s} - T_{wall,a} \right) + \frac{2 \cdot h_{wall,a} \cdot r_{wall,a}}{\rho \cdot C_p \left(r_{wall,a}^2 - r_{wall,so}^2 \right)} \left(\Delta T_{NTU} \right) \quad (4.13)$$

Here subscript *wall,si* and *wall,r* refers to the inner and outer surfaces of the wall between the refrigerant and the PCM, subscript *wall,so* and *wall,a* refer to the inner and outer surfaces of the wall between the PCM and the air, ΔT_{NTU} is the air temperature difference calculated in the air side formulation (see Eq. 4.15), and C_p is the specific heat.

The advantage of using the control volume enthalpy method for this TES geometry is that it accurately characterizes the location of the phase change front while being computationally compact. This allows one to model changes in heat transfer rates during freezing/melting cycles.

This is important for mobile applications which can experience rapid changes in cooling demand and consequently sudden transitions between heating and cooling of the PCM.

The air side formulation is taken from an effectiveness-NTU method used in [59]. This method is valid for a bank of tube heat exchangers with air as the exterior fluid. The method begins by finding the outer convection coefficient using Eq. 4.14.

$$\frac{2\alpha_{wall,a}\cdot r_{wall,a}}{k_{air}} = \varphi_1\cdot\varphi_2\cdot \mathrm{Re}_{D,\max}^{\varphi_3}\cdot \mathrm{Pr}^{0.36}\left(\frac{\mathrm{Pr}}{\mathrm{Pr}_s}\right)^{1/4} \qquad (4.14)$$

Here $Re_{D,max}$ is the maximum Reynolds number throughout the tube bank, Pr is the Prandtl number of air evaluated at the air inlet temperature, Pr_s is the Prandtl number of air evaluated at the tube surface temperature, and φ_1, φ_2, and φ_3 are correlation constants given in [59]. The temperature difference across the tube bank is then calculated using Eq. 4.15. Once ΔT_{NTU} is calculated, the air outlet temperature is calculated using Eq. 4.16 where the subscripts *air,in* and *air,out* refer to the air inlet and outlet of the TES.

$$\Delta T_{NTU} = \left(T_{air,in} - T_{wall,a}\right)\cdot\left(1-\exp\left(\frac{-\alpha_o\cdot A_s}{u_{air}\cdot C_{p,air}}\right)\right) \qquad (4.15)$$

$$T_{air,out} = T_{air,in} - \Delta T_{NTU} \qquad (4.16)$$

For this study the component sizes are taken from a laboratory training stand located at the University of Illinois at Urbana-Champaign [60]. The system model includes 5 major components: the compressor, condenser, evaporator, expansion valve, and thermal storage device. The variable speed semi-hermetic compressor has a maximum speed of 2000 RPM. The tube-and-fin condenser has a tube diameter of 8.1 mm, a tube length of 10.7 m, and a fixed fan speed of 0.23 kg/s. The condenser tube mass is 4.7 kg and the tube specific heat is 0.47 kJ/kg. Similarly, the evaporator has the same tube diameter with a length of 11.5 m, a fixed fan speed of 0.17 kg/s, a tube mass of 2.7 kg, and a tube specific heat of 0.49 kJ/kg. Finally, a small TES with 2.5 kg of water is used for the storage medium and R134a is used as the refrigerant.

4.2 Thermal Management Strategy

Like the hybrid vehicle, the operation of the PVCC can be decomposed into two distinct modes of operation: an ON mode when the compressor is used to pump refrigerant through the refrigeration system and an OFF mode when the compressor is shut off and the cooling demand is met by the TES. Using the approach outlined in Chapter 2, logic is presented for switching between these two modes and discrete linear model predictive controllers are implemented to control the actuators within each mode of operation. For each mode, the output signals used by the controllers are the pressure at the inlet and outlet of the compressor (respectively $P_{comp,in}$ and $P_{comp,out}$), the TES air inlet temperature ($T_{air,in}$), the TES outer wall temperature ($T_{wall,a}$), the cooling capacity delivered by the evaporator (γ_{ev}), the cooling capacity delivered by the storage unit (γ_s), and the temperature/pressure of refrigerant exiting the evaporator ($T_{ev,ref}$, $P_{ev,ref}$). For this study the evaporator and condenser fans are assumed to operate at fixed speeds. Unlike the previous case study, in this example a linear discrete prediction model is derived using system ID. This will demonstrate how the control design method can be adapted to different constraints on system knowledge and modeling. The MATLAB code used to define the MPC in this study is given in Appendix I.

4.2.1 Prediction model

The complete PVCC system model is highly nonlinear including fluid and heat transfer dynamics as well as phase changes in the refrigerant and storage material. The prediction model is a discrete linear approximation of these dynamics derived using a time-domain system identification procedure. Four inputs (compressor speed, expansion valve opening, flow control valve command, and refrigerant mass flow rate across the TES) were given random Gaussian perturbations about a set of nominal conditions shown in Table 4.1. A detailed nonlinear model of the PVCC was used to predict four desired outputs of the system: the pressure at the inlet of the compressor, cooling capacity delivered by the evaporator, a modulated cooling capacity delivered by the TES ($\gamma_{s,mod}$), and the superheat temperature of refrigerant exiting the evaporator ($T_{SH,ev}$). Superheat is the difference between the refrigerant temperature and its boiling temperature. The cooling capacity delivered by the evaporator and the superheat temperature of refrigerant exiting the evaporator are calculated from states of the nonlinear model and fluid

properties tables according to Eq. 4.17 and Eq. 4.18. The modulated cooling capacity delivered by the TES is given by Eq. 4.19. This quantity will be used in the objective function to express the predicted cooling capacity of the storage unit relative to the temperature difference between the inlet air and TES outer wall measured at the beginning of the prediction horizon. The time scale of temperature change within the PCM is slow relative to the update rate of the model predictive control (MPC) so this modification allows one to simplify the identified system dynamics without sacrificing accuracy.

$$\gamma_{ev} = C_{p,air} \cdot \dot{m}_{ev,air} \cdot \Delta T_{ev,air} \qquad (4.17)$$

$$T_{SH,ev} = T_{ev,ref} - T_{sat} \qquad (4.18)$$

$$\gamma_{s,\mathrm{mod}} = \frac{C_{p,air} \cdot u_{air} \cdot \Delta T_{NTU}}{\left(T_{air,in} - T_{wall,a}\right)} \qquad (4.19)$$

Here $\Delta T_{e,air}$ is the change in air temperature entering and exiting the evaporator and T_{sat} is the refrigerant saturation temperature. Using this data a 5-state linear, discrete-time model is identified using a standard prediction error/maximum likelihood system identification algorithm from the MATLAB System Identification Toolbox [61]. The time step of the discrete model is 10 seconds since a 10 second update rate will be used for the MPC. See Appendix C for the matrices comprising the state space model. Since the states of the prediction model have no physical meaning, a state estimator is used to estimate the values of these states from input/output measurements.

Table 4.1: Nominal Inputs for System Identification

Compressor Speed	1200 RPM
Expansion Valve	14%
Flow Control Valve	0.05
Air Mass Flow Rate	0.001

The graphs below show the response of the discrete linear prediction model and the detailed nonlinear model for two sets of inputs. In each case all four inputs are sinusoidal signals

with amplitudes of 20% of their mean value. For the first case the sinusoids were biased by the following values: u_{ω} = 1200 RPM, u_{eev} = 12.2%, u_{IV} = 0.5, u_{air} = 0.01 kg/s. These points were chosen because they are close to the nominal inputs used for the system ID while respecting actuator boundaries. For the second case the inputs were biased by values away from the system ID nominal inputs: u_{ω} = 900 RPM, u_{eev} = 10%, u_{IV} = 0.7, u_{air} = 0.01 kg/s. From Fig. 4.3, one can see strong agreement between the two models for case 1. However, as the inputs move away from the system ID values, the prediction model becomes less accurate. Therefore, when constructing the objective function, the integral of the tracking error will also be penalized to compensate for model inaccuracies. Alternatively, one could identify additional state space models and then switch between these models based on the current system states and inputs.

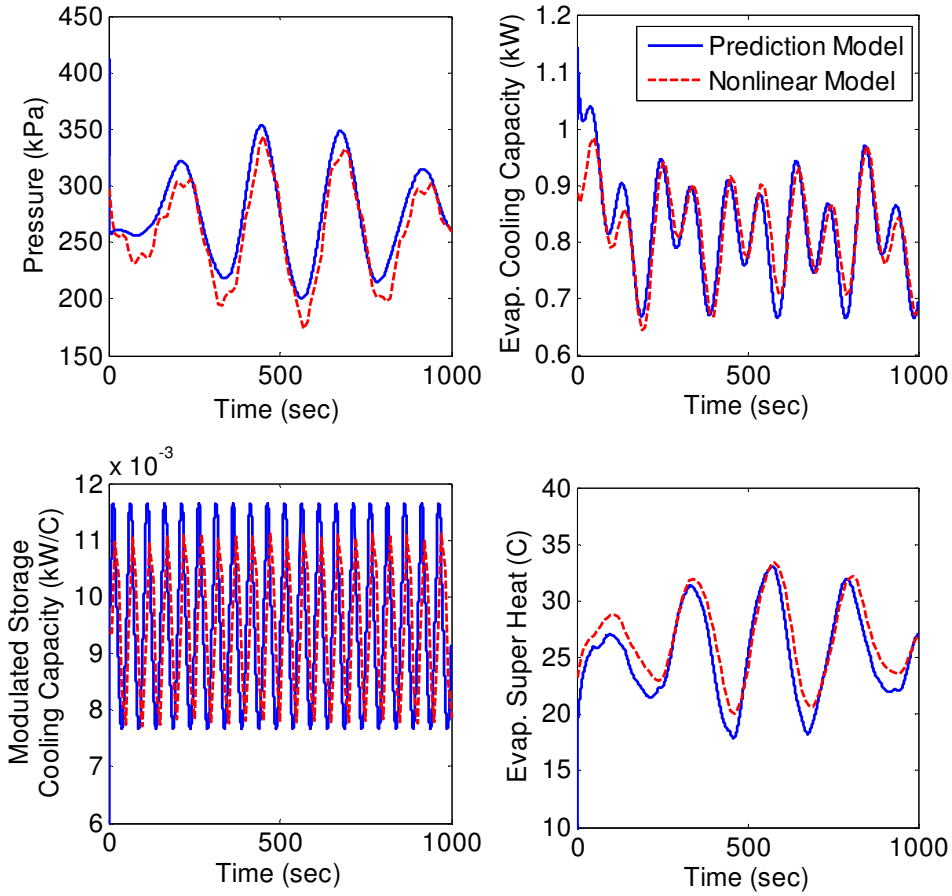

Figure 4.3: Prediction and nonlinear model comparison for case 1: 20% deviations about operating point near system ID nominal inputs

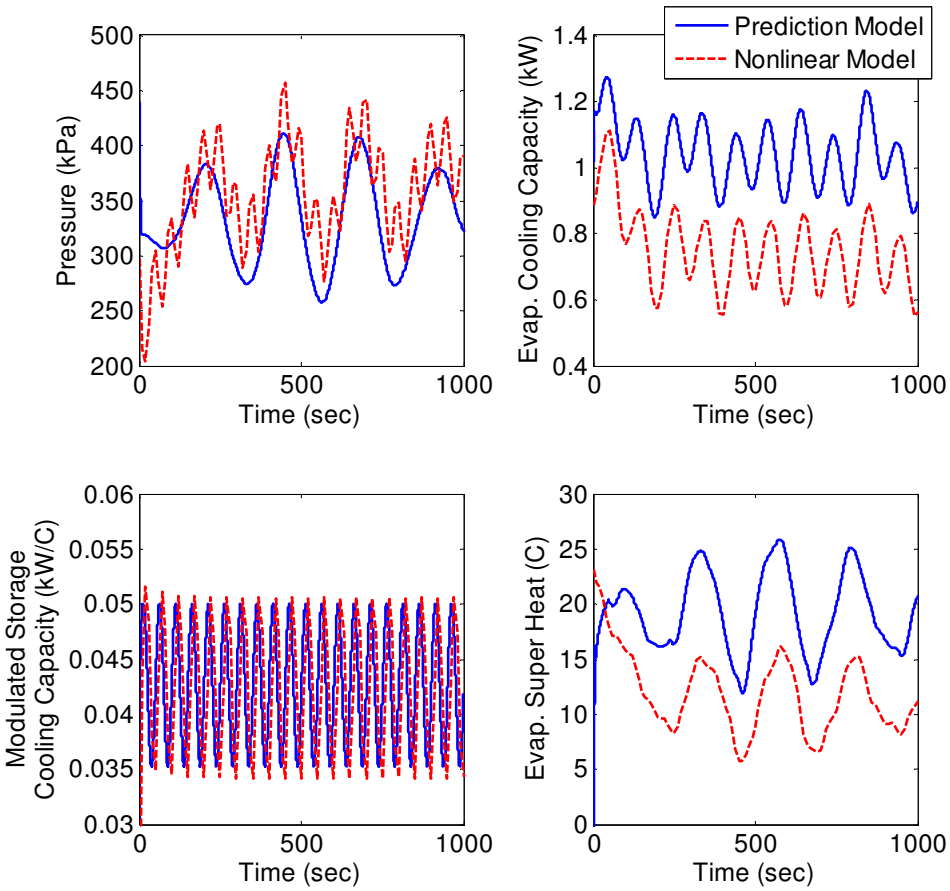

Figure 4.4: Prediction and nonlinear model comparison for case 2: 20% deviations about operating point away from system ID nominal inputs

4.2.2 Supervisory logic

Since the operation of the PVCC has been decomposed into two modes, supervisor logic is used to regulate switching between these modes. The purpose of this logic is to ensure that there is a sufficient temperature gradient between the TES and air entering the storage unit to meet the desired cooling demand. The supervisory logic is given by 4.20. When the variable Flag has a value of 1, the ON mode is used, and when it has a value of 0, the OFF mode is used. The default mode of operation is the OFF mode. This mode is maintained until the air mass flow rate command exceeds 90% of its maximum value. Once this condition occurs, the ON mode is active, and it is maintained until the outer wall temperature of the TES is below the melting

temperature of the storage medium. Once this condition occurs, the system switches back to the OFF mode.

$$Flag(t)=\begin{cases}1, & \text{If } u_{air} > 0.90 \cdot u_{air,\max} \\ 0, & \text{If } T_{wall,a} < T_m \\ Flag(t-1) & \text{Else}\end{cases} \qquad (4.20)$$

Here $u_{air,max}$ is the maximum air mass flow rate, and T_m is the melting temperature of the storage material. The MATLAB code used to define the supervisory logic in this study is given in Appendix J.

4.2.3 Objective functions

In each mode of operation, the objective functions are constructed such that the PVCC satisfies operational objectives while meeting a desired cooling capacity. In the OFF mode, the compressor is shut down and the storage unit is used to provide the desired cooling capacity. Therefore, only the fan on the TES is needed to meet the cooling demand. The objective function for this mode, given by Eq. 4.21, is composed of three terms. The first term, Eq. 4.22, penalizes deviation of the cooling capacity delivered by the TES from the desired cooling capacity. The second term, Eq. 4.23, penalizes using the compressor and evaporator to provide cooling by shutting down the compressor ($u_{\omega,des1} = 0$), closing the expansion valve ($u_{eev,des} = 0$), and isolating the TES from the rest of the system ($u_{IV,des} = 1$). Finally, the third cost term, Eq. 4.24, penalizes the integral of the cooling capacity tracking error. This term is needed to compensate for the modeling error introduced by the identified discrete linear prediction model.

$$J_{Off} = \sum_{j=1}^{n} \lambda_1 \cdot q_1(i) + q_2(i) + \zeta_1 \left(z_1(i) \right)^2 \qquad (4.21)$$

$$q_1(i) = \left(\frac{\gamma_{s,\mathrm{mod}}(i) - \dfrac{\gamma_{des}(i)}{\Delta T_{air,wall}}}{\gamma_{\max} / \Delta T_{air,wall}} \right)^2 \qquad (4.22)$$

$$q_2(i)=\left(\frac{u_\omega(i)-u_{\omega,des}}{u_{\omega,\max}}\right)^2+\left(\frac{u_{eev}(i)-u_{eev,des}}{u_{eev,\max}}\right)^2+\left(\frac{u_{IV}(i)-u_{IV,des}}{u_{IV,\max}}\right)^2 \qquad (4.23)$$

$$z_1(i)=z_1(0)+T_s\sum_{j}^{i-1}\gamma_{s,\mathrm{mod}}(j)-\frac{\gamma_{des}(j)}{\Delta T_{air,wall}} \qquad (4.24)$$

Here subscript *o* denotes a measurement made at the beginning of the prediction horizon, subscript *des* denotes the desired value, and subscript *max* denotes the maximum value of the respective variable.

The objective function for the ON mode is constructed in a similar manner, except the operational objectives have changed. Equations 4.25-4.30 describe the objective function. In this mode the evaporator is used as the primary means of meeting the cooling demand and the storage unit can be used to assist the evaporator. Equation 4.26 shows the tracking objective in which there is a penalty on the evaporator meeting the desired cooling demand and on the TES meeting the desired cooling demand minus the evaporator cooling capacity measured at the beginning of prediction horizon. In this way, the TES is used to meet the peak demands that exceed the evaporator's capacity. During this mode, a secondary goal is to operate the compressor under conditions which maximize its isentropic efficiency. The compressor's isentropic efficiency can be characterized by speed and pressure ratio, as shown in Fig. 4.5. Therefore, in order to maximize the compressor's isentropic efficiency, a penalty is placed on tracking a desired compressor speed and compressor inlet pressure (see Eq. 4.27). From Fig. 4.5, one can see that the efficiency is maximized for speeds between 1300 and 1700 RPM and pressure ratios between 3 and 3.5. The desired compressor speed was chosen to be 1400 RPM and the desired compressor inlet pressure is given by Eq. 4.28. Similar to the OFF mode, a penalty is also placed on the integral of the cooling demand tracking errors, shown in Eq. 4.29 and Eq. 4.30. Since a single state space model is used to predict the future response, it is straightforward to compute the Hessian matrix for each of the quadratic programming problems. The Hessian matrices along with their eigenvalues are given in Appendix D. From this calculation one will find that the Hessian is positive definite and therefore the optimization problem is strictly convex.

$$J_{on} = \sum_{i=1}^{n} \left[\lambda_2 \cdot q_3(i) + q_4(i) + \varsigma \cdot z_2(i) + \zeta_2 \cdot z_3(i) \right] \quad (4.25)$$

$$q_3(i) = \left(\frac{\gamma_{ev}(i) - \gamma_{des}(i)}{\gamma_{\max}} \right)^2 + \left(\frac{\gamma_{s,\mathrm{mod}}(i) - \frac{\left(\gamma_{des}(i) - \gamma_{e,o}\right)}{\Delta T_{air,wall}}}{\gamma_{\max} / \Delta T_{air,wall}} \right)^2 \quad (4.26)$$

$$q_4(i) = \left(\frac{u_\omega(i) - u_{\omega,des}}{u_{\omega,\max}} \right)^2 + \left(\frac{P_{comp,in}(i) - P_{comp,in,des}}{P_{comp,in,\max}} \right)^2 \quad (4.27)$$

$$P_{comp,in,des} = \frac{P_{comp,out,o}}{3.3} \quad (4.28)$$

$$z_2(i) = z_2(0) + T_s \sum_{j}^{i-1} \gamma_{ev}(j) - \gamma_{des}(j) \quad (4.29)$$

$$z_3(i) = z_3(0) + T_s \sum_{j}^{i-1} C_{s,\mathrm{mod}}(j) - \left(\frac{\gamma_{des}(j) - \gamma_{ev,old}}{\Delta T_{air,wall}} \right) \quad (4.30)$$

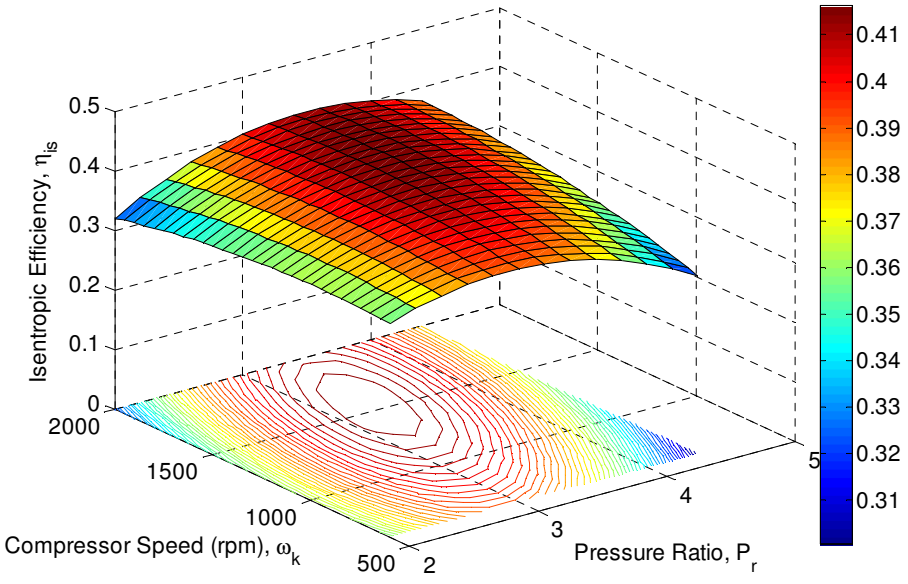

Figure 4.5: Compressor isentropic efficiency map; color bar indicates efficiency

For the PVCC, there exist upper and lower constraints on each of the input variables. The constraints on compressor speed, flow control valve command, and air mass flow rate, exist to ensure that actuator limitations are not exceeded by the controller. The upper constraint on expansion valve opening ensures that the pressure and temperature of the refrigerant drop enough to guarantee that only superheated vapor enters the compressor. In addition to these constraints, there are upper and lower limits placed on evaporator superheat to prevent liquid from entering the compressor and to ensure that the evaporator is operated in an efficient manner (if the superheat temperature is too high then there is poor heat transfer between the air and refrigerant in the evaporator). An upper bound is also placed on the evaporator cooling capacity to prevent overly taxing the PVCC system since the TES can be used to aid in meeting peak loads. These constraints are given by Eq. 4.31 through Eq. 4.36.

$$0 \text{ RPM} \le u_{\omega} \le 2000 \text{ RPM} \quad (4.31)$$

$$0 \le u_{IV} \le 1 \quad (4.32)$$

$$0\% \le u_{eev} \le 15\% \quad (4.33)$$

$$0 \text{ kg/s} \le u_{air} \le 0.09 \text{ kg/s} \quad (4.34)$$

$$5 \text{ C}^{\text{o}} \le T_{SH,ev} \le 15 \text{ C}^{\text{o}} \quad (4.35)$$

$$\gamma_{ev} \le 1.5 \text{ kW} \quad (4.36)$$

4.3 Simulation Results

To evaluate the effectiveness of the EMS, several simulations were run with the model predictive controller applied to the detailed nonlinear model of the PVCC. A 10 second update rate for the EMS was chosen because this is an order of magnitude faster than the time scale for cycling the compressor on. The following values were used in the objective function: $\lambda_1 = \lambda_2 = 1000$, $\varsigma = 0.01$, and $\zeta_1 = \zeta_2 = 0.1$. Similar to the hydraulic hybrid vehicle case study, these weights ensure that meeting the cooling demand is the primary objective and efficient compressor operation is a secondary goal. For the simulation study, two duty cycles were chosen which reflects the long time scale of refrigerated vehicle operation and the aggressive

transients that can occur when the truck door is opened, allowing ambient air to enter the truck cargo space. The first is a rural duty cycle for which there are fewer disturbances (door openings). In this duty cycle, there is a long period of low cooling capacity demand between two delivery events. The second duty cycle is an urban duty cycle in which there are frequent door openings and the storage unit is expected to have a larger impact. The duty cycle represents the frequent starts/stop one might encounter in a home food delivery truck. The duty cycles are shown in Fig. 4.6. When the refrigerated truck is traveling between delivery locations, the product is well insulated and so the desired cooling capacity is set to 0.85 kW. When the truck is making a delivery, there is a much greater thermal load on the system due to the open door. Therefore, to maintain the desired temperature inside the cargo space, the required cooling capacity is increased to 2.08 kW (the maximum attainable cooling capacity of the evaporator). These magnitudes were chosen because they highlight the limitations of sizing a traditional VCC system. The refrigeration system is sized for peak loading conditions but usually operates at conditions off peak demand. Later it will be demonstrated through simulation how the use of storage allows one to shave peak loads and normalize the operation of the compressor.

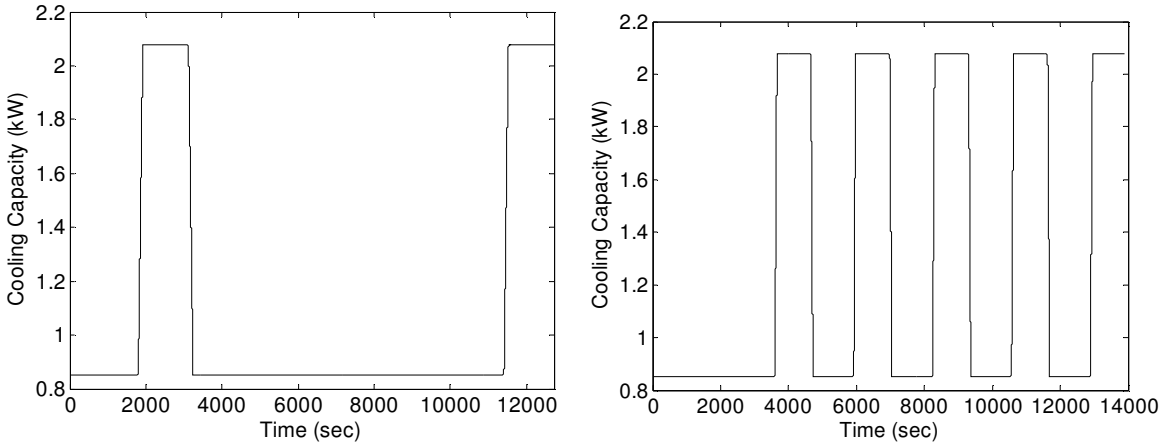

Figure 4.6: Duty cycles (from left to right: rural, urban)

We assume here that the refrigerated transport vehicle follows a prescribed trajectory wherein the time between delivery events can be predicted. This information could be extracted from data on previous deliveries over the same route. Therefore, the MPC is allowed to use future knowledge of the desired cooling capacity in the prediction horizon. To assess the value of including this information in the MPC, the simulation over the rural duty cycle was run with

prediction horizon lengths of 1, 5, 10, 15, and 20 steps. The compressor energy consumption and root mean square (RMS) tracking error for each simulation are shown in Fig. 4.7 and Fig 4.8. For very short prediction horizons, both the energy consumption and tracking error are large. Similarly, as the prediction horizon increases (> 5 steps) the energy consumption and tracking error increase. This is due to modeling error having an exacerbated effect for larger prediction horizons. Overall, the magnitude of change in energy consumption and tracking error are small between the $N = 1$ and $N = 5$ cases, suggesting that including more information about the future demand might not be worth the cost to acquire this information (larger processing power for solving the optimization problem, more memory for storing data, communication with weather monitors, etc.).

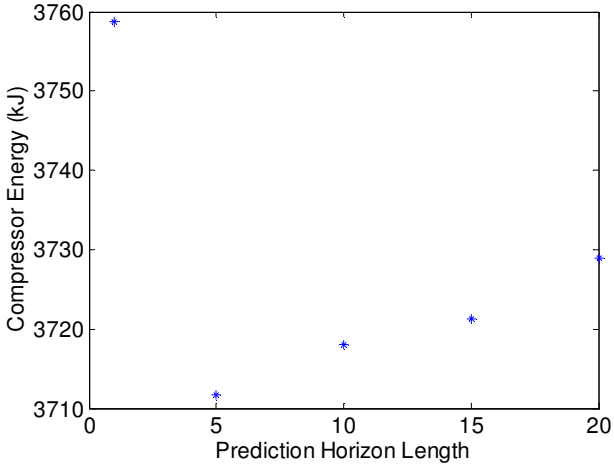

Figure 4.7: Energy consumed by the compressor for different prediction horizon lengths

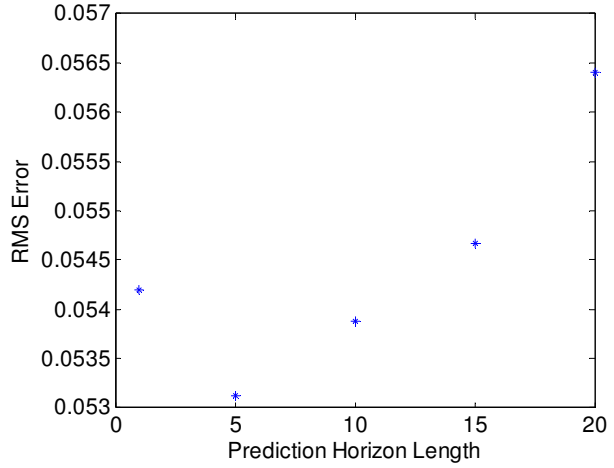

Figure 4.8: RMS tracking error for different prediction horizon lengths

Using a prediction horizon length of 5 steps, simulated PVCC response for the rural case is shown in Fig. 4.9. From these results one can see that the thermal management strategy was able to track the desired cooling capacity while observing constraints on the inputs and evaporator cooling capacity. The cycle starts with the thermal storage unit completely discharged and operating in the ON mode. During this mode, the evaporator and TES are used together to meet the cooling demand. Between delivery events, the storage unit is charged while the cooling demand is satisfied by the evaporator. Once the PCM is completely frozen, the system switches to the OFF mode and the compressor is cycled off. During this mode only the TES is used to meet the cooling demand and this operation continues until there is no longer a sufficient temperature gradient between the TES and the incoming air to meet the cooling demand. Note that for large cooling demands, when the evaporator and TES are used together, the evaporator cooling capacity exceeds the 1.5 kW upper limit. This slight excess is a result of the prediction model errors.

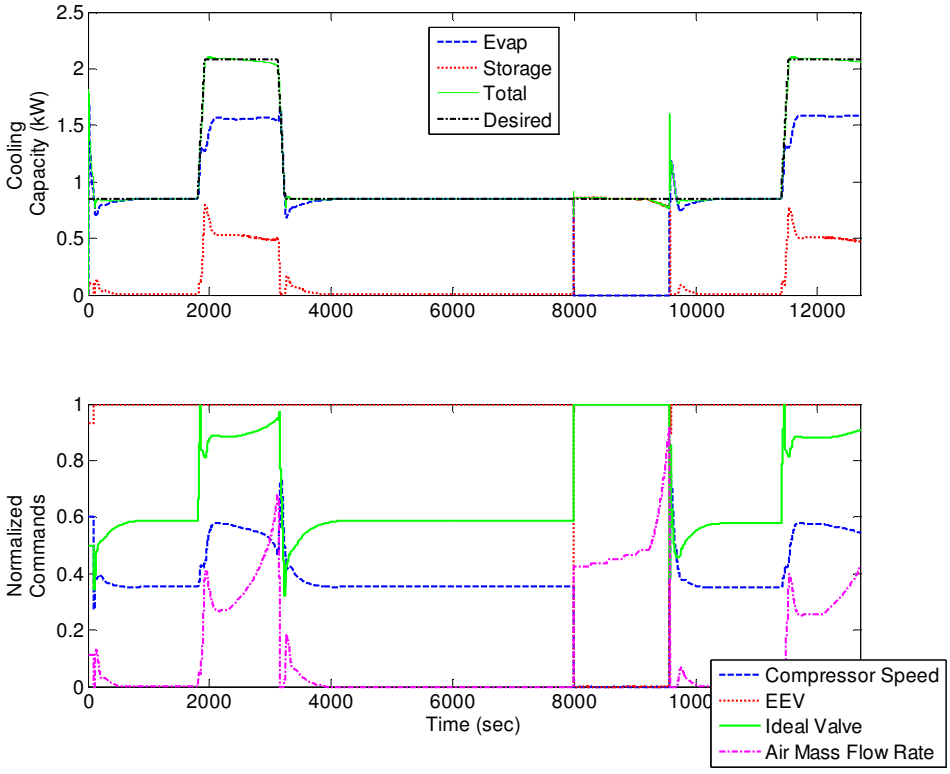

Figure 4.9: Simulated PVCC response for rural duty cycle with a prediction horizon of 5 steps

The charge/discharge cycle can also be seen in the response of the TES. For this simulation study, a 5-node model was used, and the change in specific enthalpy of each node over time is given in Fig. 4.10. One can see that in the OFF mode, as warm air travels across the TES the outer node melts first and then each subsequent node melts until the entire PCM is liquid. In the ON mode, the nodes freeze in the opposite order as refrigerant is pumped through the system.

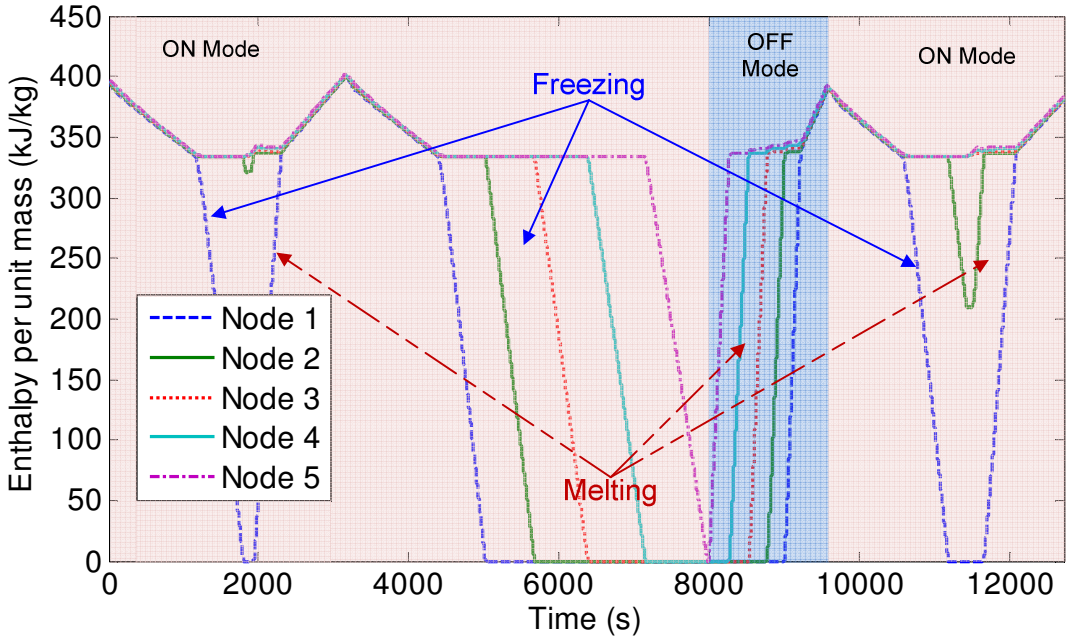

Figure 4.10: Simulated nodal enthalpies over rural duty cycle for a 5 node TES model, where each line is a separate node

Finally, during the ON mode, the compressor is also operated in regions of high isentropic efficiency. Due to the operational constraints, the compressor could not operate at its most efficient point. However, the TMS did find a balance between the compressor speed and inlet pressure which yielded high efficiency while respecting the constraints. From Fig. 4.11 one can see that the operating points of the compressor are most densely clustered around compressor speeds of 750 to 1250 RPM and pressure ratios of 2 to 3. This results in an average operating efficiency of 37.9%.

As a baseline for comparison, a conventional VCC was simulated over the same duty cycle. The component models for the PVCC and VCC are identical except the VCC does not

include a thermal storage unit. Optimal set points for the compressor speed and valve were found which minimize compressor power. To meet the 0.85 kW cooling demand, a valve opening of 1.15% and a compressor speed of 500 RPM are used (note that 500 RPM is the minimum allowed speed for continuous operation), and for the 2.08 kW cooling demand a valve opening of 15% and compressor speed of 2000 RPM are needed. In Fig. 4.11 one can see that for the VCC system, the compressor operates over a much larger range of the isentropic efficiency map. Between delivery events, the PVCC and VCC system operate at similar efficiencies and therefore it is not surprising that the average efficiency over the entire duty cycle is comparable (36.92% for the VCC case). However, during the delivery events the compressor efficiency decreased by 5.5%. As a result, the PVCC system was able to achieve a 14% decrease in compressor power consumption.

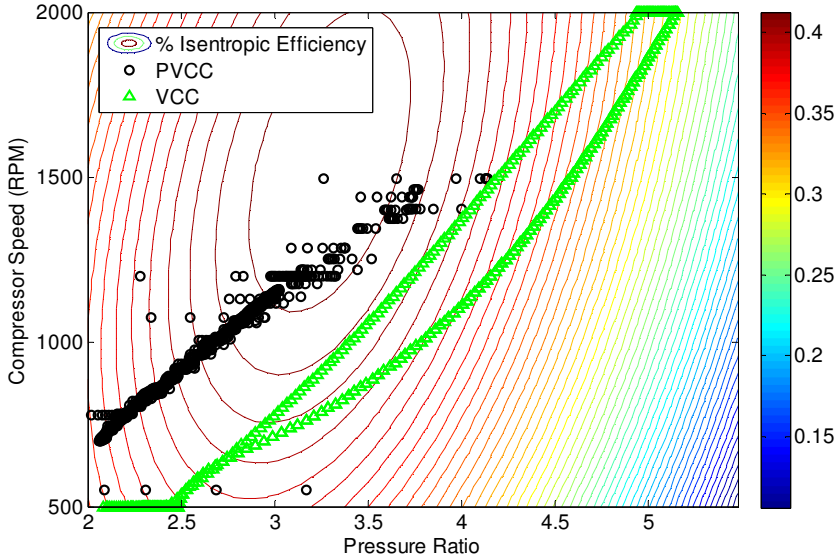

Figure 4.11: Compressor isentropic efficiency for PVCC and VCC, color bar indicates efficiency, markers indicate simulated compressor operating points

For the urban duty cycle, there are 5 delivery events separated by 20 minute intervals. The greater frequency of delivery events means that the storage is used to aid the compressor in meeting the cooling demand more often. Therefore, the system never reaches the OFF mode because the storage is never completely frozen. For this duty cycle, the ability to shave off peak demand and maintain more continuous compressor operation has a greater impact. The PVCC system achieved a 2.68% increase in average compressor isentropic efficiency and a 26%

reduction in compressor power use. Figure 4.12 shows the PVCC response and Fig. 4.13 shows the TES response. When comparing the rural and urban duty cycles, it is evident that for constant operation and an optimally sized conventional VCC system, inclusion of a thermal storage device would have little added benefit. However, in many mobile applications the demand is constantly changing as a result of external disturbances (weather, door openings, etc.) and therefore these systems often operate off design points. In this case, the thermal storage allows one to maintain highly efficient compressor operation in the face of thermal disturbances. Table 4.2 summarizes the operating efficiency and compressor energy consumption for the two cases.

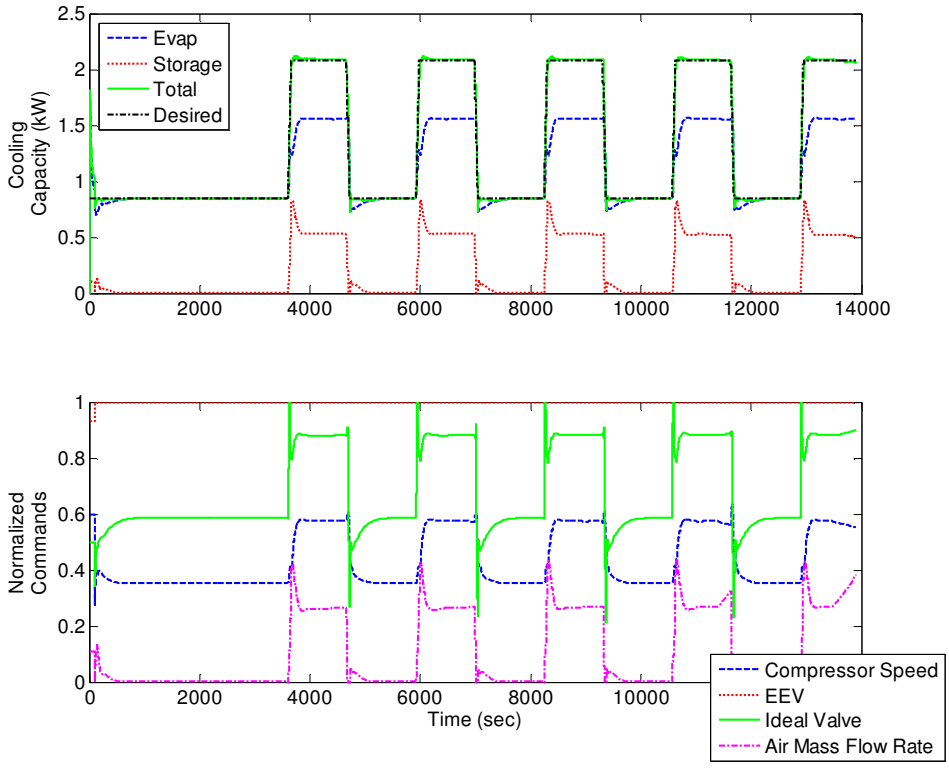

Figure 4.12: Simulated PVCC response for urban duty cycle with a prediction horizon of 5 steps

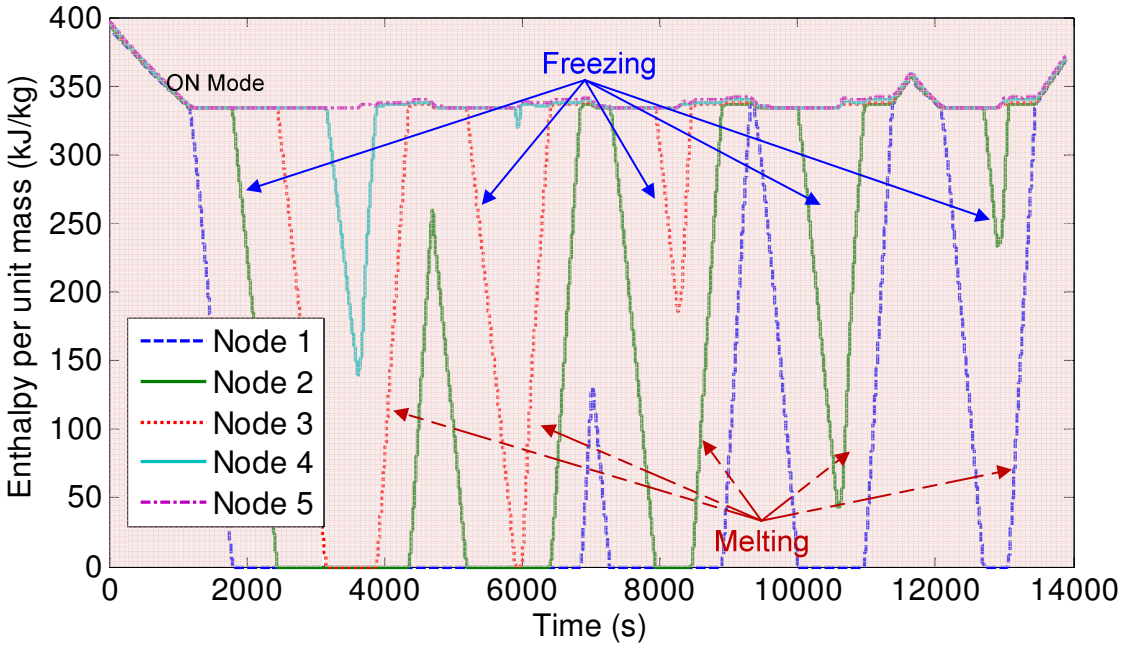

Figure 4.13: Simulated nodal enthalpies over urban duty cycle for a 5 node TES model, where each line is a separate node

Table 4.2: Compressor efficiency and energy consumption for PVCC vs. VCC

	Avg. Operating Efficiency	Energy Consumption
VCC (Rural)	36.92%	4337 kJ
VCC (Urban)	35.89%	6779 kJ
PVCC (Rural)	37.90%	3712 kJ
PVCC (Urban)	38.57%	4976 kJ

4.4 Concluding Remarks

Similar to the case study presented in Chapter 3, the thermal management study on the PVCC demonstrates the potential for energy storage coupled with intelligent control to improve system efficiency. For this study system identification is used to derive a discrete linear prediction model relating the inputs to the desired system outputs. Unlike the online linearization/discretization which was done in the hybrid vehicle study, this model introduces

significant model uncertainty which is mitigated by augmenting the MPC with integral action. For this case study, the MPC is also allowed to use exact information about the future cooling demand when solving the optimization problem. Through a simulation study, it was found that using this information did not significantly reduce energy consumption or improve tracking performance. This is not surprising since temperature control is a regulation problem and therefore the desired cooling capacity is constant for much of the duty cycle. Using an identified state space model for prediction and including look ahead in the cost evaluation are two modifications from the hybrid vehicle study which demonstrate the flexibility of the proposed EMS design method. Despite a radically different system architecture, slower system dynamics, and a different approach to predicting future response, the proposed EMS design method was able to show substantial improvement over a conventional system. In this case the primary benefit of including storage is the ability to level off peak loads and operate the compressor in a more constant manner. Through this load shaping a 14% decrease in energy consumption over a rural duty cycle was achieved. This reduction could be further increased by downsizing the compressor such that the most efficient operating point coincides with compressor speeds of 750 – 1250 RPM and pressure ratios of 2 – 3. It was also observed that the benefits of the PVVC system are more pronounced for a system which makes frequent stops. For the urban duty cycle, the PVCC system achieved a 26% decrease in compressor energy consumption.

Chapter 5 Conclusions

5.1 Summary of Research Contribution

Improving efficiency and reducing greenhouse gas emissions within the transportation sector is a critical societal need. To this end hybrid vehicles have been developed which offer an opportunity for improved efficiency through a combination of energy storage and energy management. These systems span a wide scale of power and weight as well as multiple energy domains: mechanical, thermal and electrical. Therefore a method for designing energy management strategies (EMS's) which could be applied to different architectures, energy domains, and applications (passenger vehicle, bus, delivery truck, etc.) was developed in this work. For this method to be generalizable to the entire class of hybrid vehicles, it needed to satisfy two criteria:

1. Computationally compact enough to be implemented in real-time with limited computing power
2. Allow for varying levels of duty cycle preview to be used

The proposed method uses a three step process to construct the EMS. First, decompose system operation into modes. Second, construct an optimization problem for each mode, and finally design a supervisory logic for regulating mode switching. This enables one to reduce the optimization problem within each mode to a quadratic programming problem. From an analysis of the actuators used in each mode, a set of quadratic cost terms associated with maximizing the efficiency of each actuator can be derived. These cost terms are combined with quadratic performance objectives to provide a convex quadratic objective function which can be solved online quickly with limited processing power. The model predictive control (MPC) framework is used to setup the optimization problem as an online receding horizon optimal controller. In this

way, the proposed methodology achieved the first objective. The ability of the EMS to be run in real-time was validated through hardware-in-the-loop experiments using the Augmented Earthmoving Vehicle Powertrain Simulator.

To satisfy the second objective, the EMS design method utilizes the fact that MPC is an online optimization method which can incorporate different prediction horizon lengths and use different previews of the duty cycle. This gives one a great deal of flexibility when choosing how to apply the EMS design method. Through the two case studies, it was demonstrated that the proposed method can be implemented with no duty cycle knowledge and short prediction horizon lengths for highly uncertain applications, like passenger vehicles. Conversely, if the duty cycle is well known, like a delivery cycle, the EMS can be implemented with complete duty cycle knowledge and long prediction horizon lengths. This flexibility enables one to apply the same approach for EMS design to applications with wildly varying levels of uncertainty.

The generalizability of this method was demonstrated through two case studies: a series hydraulic hybrid vehicle (SHHV) and a refrigerated delivery truck. The SHHV is a passenger vehicle which uses a hydrostatic transmission with a high pressure gas charged accumulator for energy storage. The refrigerated delivery truck uses a parallel vapor compression cycle (PVCC) system that includes thermal storage. These case studies employ different architectures, different energy domains, and different degrees of knowledge of the system and duty cycle. However, the same method for energy management was able to achieve energy savings for both. The hydraulic hybrid achieved an 11% decrease in fuel consumption over a non-hybrid vehicle powertrain for the Urban Dynamometer Driving Schedule. Similarly, the PVCC system achieved a 14% reduction in energy consumption for rural and a 26% reduction in energy consumption for urban duty cycles over a conventionally equipped delivery truck. In both cases, these improvements were achieved by applying the procedure outlined in Chapter 2 despite the differences in application and problem formulation.

Finally, the performance of the EMS design method was also benchmarked against two alternate methods for designing EMS's; rule-based and stochastic dynamic programming (SDP). Both of these techniques are offline optimization methods that use information about the system and duty cycle to minimize fuel consumption. A simulation study was conducted to determine how these three approaches are affected by variations in duty cycle and system parameters.

From this study it was observed that the proposed EMS design method is the most robust to these perturbations. This was achieved because the EMS was implemented without making assumptions on the duty cycle, unlike the rule-based and SDP approaches. However, the tradeoff for this flexibility is that the EMS was unable to achieve the fuel savings of the SDP solution. This study highlights the classic tradeoff between optimality and robustness which is a function of how much knowledge one has of their system. The greater the information one assumes, the more optimal the controller may perform but at the expense of adaptability.

5.2 Future Work

This work laid a rigorous framework for designing EMS's; including objective function selection and subsequent receding horizon optimization of energy use in hybrid vehicles. The general framework and tools laid out here can be used to explore other architectures, applications, and energy domains. Two directions for future consideration are how to couple this energy management framework with architecture design and how to improve system performance through optimization of the supervisory logic.

Effective energy management is only half of the challenge of fully realizing the potential of hybrid vehicles. Another critical component is optimizing the design of the system, including component sizing. Since the proposed method formulates the EMS from an analysis of individual component efficiencies, one has direct and immediate insight into how each component affects energy management. For example, the results of the refrigerated delivery truck study could be used to improve the design of the PVCC. Figure 4.11 clearly shows that to improve efficiency the compressor should be downsized such that the peak efficiency occurs for speeds around 1000 RPM and pressure ratios of 2.75. Coupling this EMS design method with architecture selection and sizing would allow one to rapidly evaluate many configurations, and hopefully converge to an optimal design and EMS.

Another direction for further development of this work is the design of the supervisory logic. In the present studies the supervisory logics were defined as a set of rules based primarily on the state of charge of the storage unit. These rules were motivated by an understanding of the system and its limitations. However, there is potential to improve these rules by using an optimization based approach to design, such as deterministic or stochastic dynamic

programming, and allowing the use of duty cycle preview. From the comparison of the rule-based, SDP, and MPC approaches to EMS design given in Chapter 3, it is evident that there is potential energy savings the current method does not realize. By using methods like SDP to leverage duty cycle preview in the design of the supervisory logic and MPC to perform optimization within each mode, it may be possible to combine the best features of each approach.

List of References

[1] S. C. Davis, S. W. Diegel, and R. G. Boundy, *Transportation Energy Data Book*, 31st ed. Oak Ridge National Laboratory, 2012.

[2] A. Sciarretta and L. Guzzella, "Control of Hybrid Electric Vehicles," *IEEE Control Systems Magazine*, pp. 60–70, 2007.

[3] K. Çağatay Bayindir, M. A. Gözüküçük, and A. Teke, "A Comprehensive Overview of Hybrid Electric Vehicle: Powertrain Configurations, Powertrain Control Techniques and Electronic Control Units," *Energy Conversion and Management*, vol. 52, pp. 1305–1313, Feb. 2011.

[4] R. Hall and J. J. Kargul, "Hydraulic Hybrid Promises Big Savings for UPS," *Hydraulics and Pneumatics*, pp. 42–46, 2006.

[5] U. Diego-Ayala, P. Martinez-Gonzalez, N. McGlashan, and K. R. Pullen, "The mechanical hybrid vehicle: an investigation of a flywheel-based vehicular regenerative energy capture system," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 222, pp. 2087–2101, Nov. 2008.

[6] R. Doucette and M. McCulloch, "A comparison of high-speed flywheels, batteries, and ultracapacitors on the bases of cost and fuel economy as the energy storage system in a fuel cell based hybrid electric vehicle," *Journal of Power Sources*, vol. 196, 2011.

[7] H. Liu and J. Jiang, "Flywheel energy storage—An upswing technology for energy sustainability," *Energy and Buildings*, vol. 39, pp. 599–604, May 2007.

[8] S. M. Lukic, J. Cao, R. C. Bansal, F. Rodriguez, and A. Emadi, "Energy Storage Systems for Automotive Applications," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2258–2267, 2008.

[9] A. Pedchenko and E. J. Barth, "Design and Validation of a High Energy Density Elastic Accumulator Using Polyurethane," in *ASME Dynamic Systems and Control Conference*, 2009, vol. i, pp. 283–290.

[10] J. Van de Ven, M. Olson, and P. Li, "Development of a hydro-mechanical hydraulic hybrid drive train with independent wheel torque control for an urban passenger vehicle," in *International Fluid Power Exposition*, 2008.

[11] Z. Filipi and Y. J. Kim, "Hydraulic Hybrid Propulsion for Heavy Vehicles: Combining the Simulation and Engine-In-the-Loop Techniques to Maximize the Fuel Economy and Emission Benefits," *Oil & Gas Science and Technology – Revue de l'Institut Français du Pétrole*, vol. 65, no. 1, pp. 155–178, Sep. 2009.

[12] E. Karden, S. Ploumen, B. Fricke, T. Miller, and K. Snyder, "Energy storage devices for future hybrid electric vehicles," *Journal of Power Sources*, vol. 168, no. 1, pp. 2–11, May 2007.

[13] A. Burke, "Batteries and ultracapacitors for electric, hybrid, and fuel cell vehicles," *Proceedings of the IEEE*, vol. 95, no. 4, 2007.

[14] H. A. Borhan, A. Vahidi, A. M. Phillips, M. L. Kuang, and I. V. Kolmanovsky, "Predictive Energy Management of a Power-Split Hybrid Electric Vehicle," in *American Control Conference*, 2009, pp. 3970–3976.

[15] S. Lu, K. Corzine, and M. Ferdowsi, "A New Method of Utilizing Ultra-Capacitor Energy Sources in Hybrid Electric Vehicles Over a Wide Speed Range," in *IEEE Applied Power Electronics Conference and Exposition*, 2007, pp. 222–228.

[16] W. Lhomme, P. Delarue, P. Barrade, A. Bouscayrol, and A. Rufer, "Design and Control of a Supercapacitor Storage System for Traction Applications," in *Industry Applications Conference*, 2005, vol. 3, pp. 2013–2020.

[17] W. Backé, "The present and future of fluid power," *Proceedings of the Institution of Mechanical Engineers, Part I, Journal of Systems and Control Engineering*, vol. 207, 1993.

[18] Y. Yan, G. Liu, and J. Chen, "Integrated modeling and optimization of a parallel hydraulic hybrid bus," *International Journal of Automotive Technology*, vol. 11, no. 1, pp. 97–104, 2010.

[19] B. Wu, C.-C. Lin, Z. Filipi, H. Peng, and D. Assanis, "Optimal Power Management for a Hydraulic Hybrid Delivery Truck," *Vehicle System Dynamics*, vol. 42, no. 1–2, pp. 23–40, Dec. 2004.

[20] R. Johri and Z. Filipi, "Low-Cost Pathway to Ultra Efficient City Car: Series Hydraulic Hybrid System with Optimized Supervisory Control," *SAE International Journal of Engines*, vol. 2, no. 2, pp. 505–520, 2010.

[21] K. Stelson and J. Meyer, "Optimization of a passenger hydraulic Hybrid vehicle to Improve Fuel Economy," *7th JFPS International Symposium on Fluid Power*, 2008.

[22] T. O. Deppen, A. G. Alleyne, K. A. Stelson, and J. J. Meyer, "Optimal Energy Use in a Light Weight Hydraulic Hybrid Passenger Vehicle," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 4, p. 041009, 2012.

[23] C. Lin, H. Peng, J. W. Grizzle, and J.-M. Kang, "Power Management Strategy for a Parallel Hybrid Electric Truck," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 839–849, 2003.

[24] J. Meyer, K. Stelson, A. G. Alleyne, and T. Deppen, "Power Management Strategy for a Parallel Hydraulic Hybrid Passenger Vehicle Using Stochastic Dynamic Programming," in *7th International Fluid Power Conference*, 2010.

[25] C. Lin, H. Peng, and J. Grizzle, "A Stochastic Control Strategy for Hybrid Electric Vehicles," in *American Control Conference*, 2004, pp. 4710–4715.

[26] H. Borhan, C. Zhang, A. Vahidi, A. M. Phillips, M. L. Kuang, and S. Di Cairano, "Nonlinear model predictive control for power-split hybrid electric vehicles," in *IEEE Conference on Decision and Control*, 2010, pp. 4890–4895.

[27] Z. Filipi, L. Louca, B. Daran, C. Lin, U. Yildir, B. Wu, M. Kokkolaras, D. Assanis, H. Peng, P. Papalambros, J. Stein, D. Szkubiel, and R. Chapp, "Combined optimisation of design and power management of the hydraulic hybrid propulsion system for the 6× 6 medium truck," *International Journal of Heavy Vehicle Systems*, vol. 11, pp. 371–401, 2004.

[28] J. W. Grizzle, "Energy Management Strategy for a Parallel Hybrid Electric Truck," in *American Control Conference*, 2001, vol. 5, no. D, pp. 2878–2883.

[29] D. Zhou, C. Y. Zhao, and Y. Tian, "Review on thermal energy storage with phase change materials (PCMs) in building applications," *Applied Energy*, vol. 92, pp. 593–605, Apr. 2012.

[30] B. Zalba, J. M. Marín, L. F. Cabeza, and H. Mehling, "Review on thermal energy storage with phase change: materials, heat transfer analysis and applications," *Applied Thermal Engineering*, vol. 23, no. 3, pp. 251–283, Feb. 2003.

[31] S. A. Tassou, G. De-Lille, and Y. T. Ge, "Food transport refrigeration – Approaches to reduce energy consumption and environmental impacts of road transport," *Applied Thermal Engineering*, vol. 29, no. 8–9, pp. 1467–1477, Jun. 2009.

[32] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2005.

[33] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[34] L. Balbis, A. Ordys, M. Grimble, and Y. Pang, "Tutorial introduction to the modelling and control of hybrid systems," *International Journal of Modeling, Identification, and Control*, vol. 2, no. 4, 2007.

[35] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, Mar. 1999.

[36] M. Kozlov, S. Tarasov, and L. Khachiyan, "The Polynomial Solvability of Convex Quadratic Programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 5, pp. 223–228, 1980.

[37] J. B. Heywood, *Internal Combustion Engine Fundamentals*. McGraw-Hill, 1988.

[38] Y. Ma, A. Kelman, A. Daly, and F. Borrelli, "Predictive Control for Energy Efficient Buildings with Thermal Storage: Modeling, Simulation, and Experiments," *IEEE Control Systems Magazine*, vol. 32, pp. 44–64, 2012.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*, vol. 25, no. 3. Cambridge University Press, 2010, pp. 487–487.

[40] D. Liberzon, *Calculus of Variations and Optimal Control Theory*. Princeton University Press, 2011.

[41] N. Jain and A. G. Alleyne, "A framework for the optimization of integrated energy systems," *Applied Thermal Engineering*, vol. 48, pp. 495–505, Dec. 2012.

[42] "quadprog," 2013. [Online]. Available: http://www.mathworks.com/help/optim/ug/quadprog.html.

[43] D. Liberzon, *Switching in Systems and Control*. Springer, 2003.

[44] R. Zhang, A. Alleyne, and E. Prasetiawan, "Modeling and H[sub 2]/H[sub ∞] MIMO Control of an Earthmoving Vehicle Powertrain," *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, no. 4, p. 625, 2002.

[45] A. Montgomery and A. Alleyne, "Optimizing the efficiency of electro-hydraulic powertrains," in *Proceedings of International Mechanical Engineering Congress & Exposition*, 2006, pp. 1–9.

[46] E. Prasetiawan, R. Zhang, A. G. Alleyne, and T. Tsao, "Modeling and Control Design of a Powertrain Simulation Testbed for Earthmoving Vehicles," in *International Mechanical Engineering Congress and Exposition: The Fluid Power and Systems Technology Division*, 1999.

[47] R. Zhang and A. G. Alleyne, "Dynamic Emulation Using an Indirect Control Input," *Journal of Dynamic Systems, Measurement, and Control*, vol. 127, no. 1, p. 114, 2005.

[48] D. Carter and A. Alleyne, "Load Modeling and Emulation for an Earthmoving Vehicle Powertrain," in *American Control Conference*, 2003, pp. 4963–4968.

[49] A. Takasaki, T. Mizutani, K. Kitagawa, T. Yamahana, K. Odaka, T. Kuzuya, Y. Mizuno, and Y. Nishikawa, "Development of new hybrid transmission for 2009 Prius," in *EVS24 International Battery, Hybrid Fuel Cell Electric Vehical Symposium*, 2009, pp. 1–6.

[50] "ADVISOR 2004," 2004. [Online]. Available: www.avl.com.

[51] R. Bellman, *Dynamic Programming*. Princeton: Princeton University Press, 1957.

[52] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Norwell, MA: Kluwer Academic Publishers, 2003.

[53] "Refrigerated transport: what's new?," *International Journal of Refrigeration*, vol. 25, no. 5, pp. 501–503, Aug. 2002.

[54] B. Li and A. G. Alleyne, "A dynamic model of a vapor compression cycle with shut-down and start-up operations," *International Journal of Refrigeration*, vol. 33, no. 3, pp. 538–552, May 2010.

[55] T. L. McKinley and A. G. Alleyne, "An advanced nonlinear switched heat exchanger model for vapor compression cycles using the moving-boundary method," *International Journal of Refrigeration*, vol. 31, no. 7, pp. 1253–1264, Nov. 2008.

[56] Y. Dutil, D. R. Rousse, N. Ben Salah, S. Lassue, and L. Zalewski, "A review on phase-change materials: Mathematical modeling and simulations," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 112–130, Jan. 2011.

[57] F. Wang, G. Maidment, J. Missenden, and R. Tozer, "The novel use of phase change materials in refrigeration plant. Part 2: Dynamic simulation model for the combined system," *Applied Thermal Engineering*, vol. 27, no. 17–18, pp. 2902–2910, Dec. 2007.

[58] J. Fasl, "Modeling and Control of Hybrid Vapor Compression Cycles," University of Illinois at Urbana-Champaign.

[59] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, vol. 6th. John Wiley & Sons, 2007, p. 997.

[60] B. P. Rasmussen, "Dynamic modeling and advanced control of air conditioning and refrigeration systems," University of Illinois at Urbana-Champaign, 2005.

[61] L. Ljung, *System Identification Toolbox: For Use With Matlab*. Natic, MA: Math Works Inc., 2001.

Appendix A AEVPS Parameters

Below is a list of the physical parameters used in the dynamic model of the Augmented Earthmoving Vehicle Powertrain Simulator (AEVPS).

Table A.1: AEVPS system parameters

A	2 (m^2)	D_m	4.005 ($cm^3.rad^{-1}$)
b_e	0.29 ($N.m(rad.s^{-1})^{-1}$)	D_p	4.216 ($cm^3.rad^{-1}$)
b_L	0.5 ($N.m(rad.s^{-1})^{-1}$)	I_e	0.383 ($kg.m^2$)
b_m	0.0463 ($N.m(rad.s^{-1})^{-1}$)	I_m	0.0019 ($kg.m^2$)
b_w	1 ($N.m(rad.s^{-1})^{-1}$)	k	1.4
β_d	53.23 (MPa)	K_P	37.4 ($cm^3.rad^{-2}$)
β_u	266.13 (MPa)	P_{pr}	5.17 (MPa)
Cap	$1.89*10^4$ (cm^3)	ρ_{air}	1.2 ($kg.m^{-3}$)
ψ_d	9.259 ($cm^3(s.MPa)^{-1}$)	r_w	0.31 (m)
C_{drag}	0.4	V_d	1854 (cm^3)
κ_e	$2.45*10^{-4}$ ($Nm(rad.s^{-1})^{-2}$)	V_u	2785 (cm^3)
ψ_u	0.7 ($cm^3(s.MPa)^{-1}$)		

Appendix B AEVPS MPC Simulation Study

A simulation study was conducted in which the energy management strategy described in Chapter 2 was applied to the Augmented Earthmoving Vehicle Powertrain Simulator model and was commanded to track the Urban Dynamometer Driving Schedule. For all simulations the update rate of the model predictive control was set to 1 Hz with a prediction horizon of 5 seconds and the parameters of the supervisory logic were: $\tau_{threshold}$ equals 30 Nm, $P_{threshold}$ equals 6.5 MPa, and v equals 1.9 V. The update rate and prediction horizon length were chosen such that the optimization problem could be solved in real-time, under the hardware constraints, while achieving a balance between the prediction step size and prediction length. Furthermore, a step size of 1 second and a prediction horizon of 5 steps were found to be successful in previous studies [14]. In this simulation study the effect of relative weighting of different component efficiencies within the objective function and the length of the dwell time were evaluated with respect to fuel consumption. Expanded versions of the objective functions presented in Chapter 3 were used in which the engine, pump, and valve efficiencies were each explicitly considered. Equations B.1 through B.6 give the objective functions used in this study.

$$J_1 = \sum_{j=1}^{N}\sum_{i=1}^{j}\left[\underbrace{\lambda_1\left(\frac{\omega_m(i)-\omega_{m,des}}{\omega_{m,\max}}\right)^2}_{\text{Motor Tracking}} + \lambda_2 \cdot \underbrace{C_1}_{\substack{\text{Engine}\\ \text{Efficiency}}} + \lambda_3 \cdot \underbrace{C_2}_{\substack{\text{Pump}\\ \text{Efficiency}}} + \lambda_4 \cdot \underbrace{C_3}_{\substack{\text{Valve}\\ \text{Efficency}}}\right] \qquad (B.1)$$

$$C_1 = \left(\frac{\omega_e(i)-\omega_{e,des2}}{\omega_{e,\max}}\right)^2 + \left(\frac{u_e(i)-u_{e,des2}}{u_{e,\max}}\right)^2 \qquad (B.2)$$

$$C_2 = \left(\frac{\omega_e(i) - \omega_{e,des1}}{\omega_{e,\max}} \right)^2 + \left(\frac{u_e(i) - u_{e,des1}}{u_{e,\max}} \right)^2 \quad \text{(B.3)}$$

$$C_3 = \left(\frac{P_u(i) - P_{u,des}}{P_{u,\max}} \right)^2 \quad \text{(B.4)}$$

$$J_2 = \sum_{j=1}^{N} \sum_{i=1}^{j} \left[\lambda_5 \underbrace{\left(\frac{\omega_m(i) - \omega_{m,des}}{\omega_{m,\max}} \right)^2}_{\text{Motor Tracking}} + \lambda_6 \cdot \underbrace{C_4}_{\substack{\text{Engine} \\ \text{Operation}}} \right] \quad \text{(B.5)}$$

$$C_4 = \left(\frac{\omega_e(i) - \omega_{e,des2}}{\omega_{e,\max}} \right)^2 + \left(\frac{u_e(i) - u_{e,des2}}{u_{e,\max}} \right)^2 \quad \text{(B.6)}$$

The λ_2 term seeks to optimize engine efficiency by placing the engine on the "sweet spot" of the engine efficiency map, see Fig. 3.4. This corresponds to a desired engine speed ($\omega_{e,des2}$) of 109 rad/sec (58% of max emulated engine speed) and a desired engine torque ($u_{e,des2}$) of 91.8 Nm (55% of max emulated engine power). The other cost terms of the objective function are the same as those described in Section 3.2.3. For the OFF mode, motor speed tracking had the highest priority with $\lambda_5 = 1000$ and $\lambda_6 = 1$. During the ON mode, every combination of the efficiency term weightings (λ_2, λ_3, and λ_4) was evaluated in increments of 0.1, such that they satisfied the constraints given by Eq. B.7.

$$\lambda_2 + \lambda_3 + \lambda_4 = 1 \quad \text{(B.7)}$$

Equation B.7 is used to ensure that the magnitude of the efficiency cost within the objective function is always small compared to the tracking cost which is given a much larger weight ($\lambda_1 = 1000$). In addition, each sweep of efficiency term weightings was evaluated at different dwell times ranging from 10 to 60 seconds in increments of 10 seconds. The results from one such study are summarized in Fig. B.1 and Table B.1. Each corner of the triangle in Fig. B.1 represents a case in which only one actuator is considered in the objective function. To ensure that the objective function is well posed, the case for $\lambda_2 = 0$, $\lambda_3 = 0$, and $\lambda_4 = 1$ was approximated with $\lambda_2 = 0$, $\lambda_3 = 0.01$, and $\lambda_4 = 0.99$. This ensures that the objective function has

cost terms for 3 of the 7 variables characterizing the system (4 states, 3 inputs) with 4 linear constraints from the prediction model. Note that for the simulation results presented in Chapter 3 the fuel consumption rate is lower bounded by $0.43 g.s^{-1}$ (fuel consumption under idle conditions). This bound was not imposed in this simulation study, meaning when the throttle command is zero, the fuel consumption is zero. However, the same trends for objective function weighting were observed and the same combination of weights yielded the lowest overall consumption for both methods of computing fuel consumption.

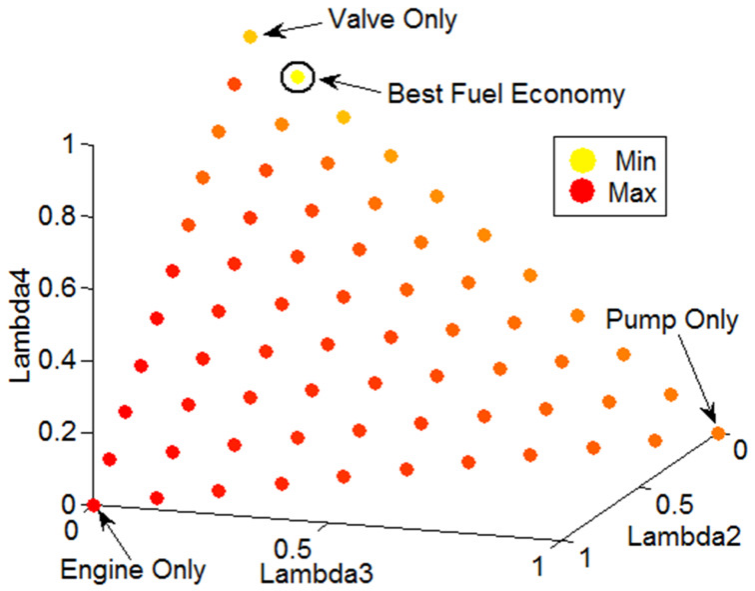

Figure B.1: Efficiency term weighting sweep for dwell time of 10 seconds, max fuel consumption: 1.5 kg, min fuel consumption: 1.1 kg

Table B.1: Fuel consumptions for different objective function weights

	Fuel Consumption
Engine Only	1.51
Pump Only	1.32
Valve Only	1.20
Best Case	1.10

Contrary to conventional wisdom which would have one believe that optimizing engine efficiency would yield maximum powertrain efficiency, these results demonstrate that focusing

on the pump and valve efficiencies will yield superior overall efficiency. From Fig. B.1 one can see that the $\lambda_2 = 0$, $\lambda_3 = 0.1$, and $\lambda_4 = 0.9$ case yielded the smallest fuel consumption. By comparison, the "engine only" case consumed 35% more fuel. The "valve only" case ($\lambda_2 = 0$, $\lambda_3 = 0.01$, and $\lambda_4 = 0.99$) achieved the second lowest consumption but the tighter tracking of upstream pressure forced the pump to operate at lower displacements and therefore overall efficiency was reduced.

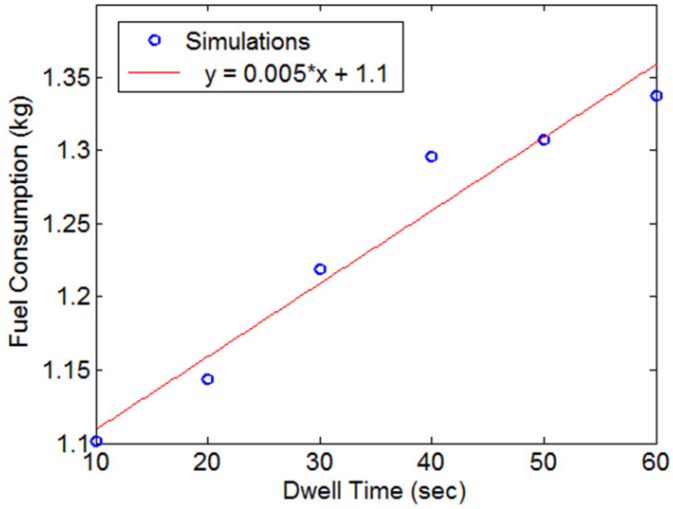

Figure B.2: Simulated fuel consumptions for different dwell times ($\lambda_2 = 0$, $\lambda_3 = 0.1$, $\lambda_4 = 0.9$)

The consumption distribution shown in Fig. B.1 was found to be consistent for all dwell times. However, with increasing dwell time there was an increase in fuel consumption. Results for the lowest fuel consumption case are shown in Fig. B.2. The increase in fuel consumption is a consequence of decreases in engine efficiency and pump displacement and an increase in average accumulator pressure during the ON mode. For the 60 second dwell time case, there was a 4% decrease in average engine efficiency, an 18% decrease in average pump displacement, and a 6% increase in average accumulator pressure compared to the 10 second dwell time case. The increase in the average accumulator pressure leads to greater losses across the valve. From Fig. B.2 one can see that there is a nearly linear increase in fuel consumption with increasing dwell time. Therefore, one may be able to achieve significant reduction in unnecessary vehicle wear and tear with a modest increase in fuel consumption.

Appendix C PVCC System ID

Below are the state space matrices used to define the prediction model for the parallel vapor compression cycle system. These matrices were derived using system ID applied to a nonlinear model of this system.

$$A = \begin{bmatrix} 0.6758 & 0.0092 & -0.0161 & 0.0472 & 0.0230 \\ 0.0190 & 0.0217 & 0.0710 & 0.0633 & 0.1248 \\ 0.0126 & 0.0427 & 0.2014 & -0.1445 & 0.4158 \\ -0.0456 & 0.0588 & 0.1654 & 0.5619 & 0.0980 \\ -0.0100 & 0.0910 & 0.3167 & 0.2048 & 0.5677 \end{bmatrix} \quad (C.1)$$

$$B = \begin{bmatrix} 0.0001 & 0.0093 & 0.5646 & -1.5405 \\ 0.0001 & -0.0007 & -0.1696 & 384.8474 \\ 0.0020 & -0.0674 & -0.0499 & 4.5734 \\ 0.0000 & -0.0006 & 0.2701 & -32.6867 \\ -0.0013 & 0.0320 & 0.0337 & -64.8587 \end{bmatrix} \quad (C.2)$$

$$C = \begin{bmatrix} 42.4777 & -10.5127 & -108.0660 & -60.9211 & 6.4975 \\ 0.9536 & 0.0031 & 0.0646 & -0.0263 & 0.0143 \\ 0.0001 & 0.0023 & -0.0000 & 0.0001 & 0.0001 \\ -8.9621 & 0.7357 & 2.9827 & 18.1416 & -4.6270 \end{bmatrix} \quad (C.3)$$

Appendix D Thermal Management Hessian Matrix

For the thermal management the Hessian matrix for the ON and OFF mode objective functions can be computed using the system matrices given in Appendix C. The Hessian matrices for the $N = 5$ case without any weighting or normalization factors are given below.

Hessian for ON mode columns 1:5:

1.05E+00	2.50E-05	-1.04E-03	-9.13E-04	-7.69E-04
2.50E-05	1.05E+00	9.85E-06	-1.05E-03	-9.15E-04
-1.04E-03	9.85E-06	1.05E+00	-1.05E-05	-1.06E-03
-9.13E-04	-1.05E-03	-1.05E-05	1.05E+00	-1.24E-05
-7.69E-04	-9.15E-04	-1.06E-03	-1.24E-05	1.05E+00
-1.79E+00	-1.94E-01	-1.23E-01	-9.81E-02	-7.93E-02
8.99E-03	-1.79E+00	-1.95E-01	-1.25E-01	-9.84E-02
4.29E-02	7.27E-03	-1.79E+00	-1.97E-01	-1.25E-01
3.53E-02	4.08E-02	4.82E-03	-1.80E+00	-1.98E-01
2.71E-02	3.22E-02	3.71E-02	5.81E-04	-1.80E+00
-3.23E+00	-3.01E+00	-2.00E+00	-1.29E+00	-8.23E-01
1.70E-01	-3.26E+00	-3.04E+00	-2.03E+00	-1.31E+00
1.87E-01	1.34E-01	-3.30E+00	-3.09E+00	-2.05E+00
1.24E-01	1.43E-01	8.33E-02	-3.36E+00	-3.10E+00
5.71E-02	6.67E-02	7.61E-02	8.05E-03	-3.37E+00
6.93E+02	-1.22E+02	-1.12E+02	-9.74E+01	-8.38E+01
5.80E+00	6.92E+02	-1.23E+02	-1.13E+02	-9.74E+01
-1.03E+01	4.37E+00	6.90E+02	-1.25E+02	-1.13E+02
-1.04E+01	-1.20E+01	2.37E+00	6.88E+02	-1.25E+02
-1.03E+01	-1.23E+01	-1.42E+01	-1.98E-01	6.88E+02

Hessian for ON mode columns 6:10:

-1.79E+00	8.99E-03	4.29E-02	3.53E-02	2.71E-02
-1.94E-01	-1.79E+00	7.27E-03	4.08E-02	3.22E-02
-1.23E-01	-1.95E-01	-1.79E+00	4.82E-03	3.71E-02
-9.81E-02	-1.25E-01	-1.97E-01	-1.80E+00	5.81E-04
-7.93E-02	-9.84E-02	-1.25E-01	-1.98E-01	-1.80E+00
6.44E+01	7.92E+00	5.04E+00	3.80E+00	2.80E+00
7.92E+00	6.42E+01	7.72E+00	4.80E+00	3.47E+00
5.04E+00	7.72E+00	6.40E+01	7.44E+00	4.40E+00
3.80E+00	4.80E+00	7.44E+00	6.37E+01	6.94E+00
2.80E+00	3.47E+00	4.40E+00	6.94E+00	6.29E+01
1.49E+02	1.29E+02	8.45E+01	5.26E+01	3.04E+01
3.43E+01	1.44E+02	1.25E+02	7.99E+01	4.74E+01
2.28E+01	2.93E+01	1.39E+02	1.19E+02	7.30E+01
1.44E+01	1.70E+01	2.29E+01	1.32E+02	1.10E+02
6.41E+00	7.58E+00	9.19E+00	1.37E+01	1.19E+02
-2.30E+04	5.20E+03	4.54E+03	3.73E+03	2.93E+03
-1.99E+03	-2.31E+04	5.05E+03	4.34E+03	3.41E+03
-1.27E+03	-2.13E+03	-2.33E+04	4.81E+03	3.97E+03
-1.13E+03	-1.44E+03	-2.35E+03	-2.36E+04	4.38E+03
-1.07E+03	-1.32E+03	-1.68E+03	-2.65E+03	-2.41E+04

Hessian for ON mode columns 11:15:

-3.23E+00	1.70E-01	1.87E-01	1.24E-01	5.71E-02
-3.01E+00	-3.26E+00	1.34E-01	1.43E-01	6.67E-02
-2.00E+00	-3.04E+00	-3.30E+00	8.33E-02	7.61E-02
-1.29E+00	-2.03E+00	-3.09E+00	-3.36E+00	8.05E-03
-8.23E-01	-1.31E+00	-2.05E+00	-3.10E+00	-3.37E+00
1.49E+02	3.43E+01	2.28E+01	1.44E+01	6.41E+00
1.29E+02	1.44E+02	2.93E+01	1.70E+01	7.58E+00
8.45E+01	1.25E+02	1.39E+02	2.29E+01	9.19E+00
5.26E+01	7.99E+01	1.19E+02	1.32E+02	1.37E+01
3.04E+01	4.74E+01	7.30E+01	1.10E+02	1.19E+02
1.18E+03	9.10E+02	6.03E+02	3.42E+02	1.33E+02
9.10E+02	9.53E+02	6.92E+02	3.99E+02	1.59E+02
6.03E+02	6.92E+02	7.39E+02	4.83E+02	1.98E+02
3.42E+02	3.99E+02	4.83E+02	5.21E+02	2.53E+02
1.33E+02	1.59E+02	1.98E+02	2.53E+02	2.51E+02
-3.03E+04	2.03E+04	1.60E+04	1.11E+04	5.34E+03
-3.29E+04	-3.12E+04	1.85E+04	1.31E+04	6.28E+03
-2.30E+04	-3.42E+04	-3.34E+04	1.49E+04	7.37E+03
-1.60E+04	-2.46E+04	-3.69E+04	-3.78E+04	8.18E+03
-1.14E+04	-1.79E+04	-2.77E+04	-4.18E+04	-4.53E+04

Hessian for ON mode columns 16:20:

6.93E+02	5.80E+00	-1.03E+01	-1.04E+01	-1.03E+01
-1.22E+02	6.92E+02	4.37E+00	-1.20E+01	-1.23E+01
-1.12E+02	-1.23E+02	6.90E+02	2.37E+00	-1.42E+01
-9.74E+01	-1.13E+02	-1.25E+02	6.88E+02	-1.98E-01
-8.38E+01	-9.74E+01	-1.13E+02	-1.25E+02	6.88E+02
-2.30E+04	-1.99E+03	-1.27E+03	-1.13E+03	-1.07E+03
5.20E+03	-2.31E+04	-2.13E+03	-1.44E+03	-1.32E+03
4.54E+03	5.05E+03	-2.33E+04	-2.35E+03	-1.68E+03
3.73E+03	4.34E+03	4.81E+03	-2.36E+04	-2.65E+03
2.93E+03	3.41E+03	3.97E+03	4.38E+03	-2.41E+04
-3.03E+04	-3.29E+04	-2.30E+04	-1.60E+04	-1.14E+04
2.03E+04	-3.12E+04	-3.42E+04	-2.46E+04	-1.79E+04
1.60E+04	1.85E+04	-3.34E+04	-3.69E+04	-2.77E+04
1.11E+04	1.31E+04	1.49E+04	-3.78E+04	-4.18E+04
5.34E+03	6.28E+03	7.37E+03	8.18E+03	-4.53E+04
1.01E+07	-1.02E+06	-1.10E+06	-1.10E+06	-1.12E+06
-1.02E+06	9.96E+06	-1.18E+06	-1.28E+06	-1.30E+06
-1.10E+06	-1.18E+06	9.77E+06	-1.40E+06	-1.52E+06
-1.10E+06	-1.28E+06	-1.40E+06	9.52E+06	-1.68E+06
-1.12E+06	-1.30E+06	-1.52E+06	-1.68E+06	9.21E+06

Hessian for OFF mode columns 1:5:

1.00E+00	1.47E-12	9.02E-13	4.19E-13	3.99E-14
1.47E-12	1.00E+00	6.91E-13	3.29E-13	3.15E-14
9.02E-13	6.91E-13	1.00E+00	2.39E-13	2.30E-14
4.19E-13	3.29E-13	2.39E-13	1.00E+00	1.56E-14
3.99E-14	3.15E-14	2.30E-14	1.56E-14	1.00E+00
3.96E-11	4.01E-11	3.05E-11	1.65E-11	1.61E-12
-2.05E-11	-5.55E-13	8.42E-12	7.38E-12	7.76E-13
-5.60E-11	-3.96E-11	-1.57E-11	-2.66E-12	-1.52E-13
-6.88E-11	-5.20E-11	-3.65E-11	-1.34E-11	-1.17E-12
-5.00E-11	-3.94E-11	-2.88E-11	-1.96E-11	-2.02E-12
4.13E-09	2.34E-09	1.18E-09	4.51E-10	4.10E-11
5.35E-09	3.10E-09	1.54E-09	5.84E-10	5.29E-11
5.51E-09	4.03E-09	2.06E-09	7.74E-10	6.99E-11
4.96E-09	3.76E-09	2.65E-09	1.05E-09	9.44E-11
3.29E-09	2.59E-09	1.90E-09	1.29E-09	1.33E-10
-2.36E-05	-1.51E-05	-8.47E-06	-3.61E-06	-3.37E-07
-2.34E-05	-1.52E-05	-8.50E-06	-3.62E-06	-3.38E-07
-2.01E-05	-1.49E-05	-8.54E-06	-3.64E-06	-3.39E-07
-1.53E-05	-1.17E-05	-8.27E-06	-3.66E-06	-3.41E-07
-8.58E-06	-6.76E-06	-4.95E-06	-3.36E-06	-3.46E-07

Hessian for OFF mode columns 6:10:

3.96E-11	-2.05E-11	-5.60E-11	-6.88E-11	-5.00E-11
4.01E-11	-5.55E-13	-3.96E-11	-5.20E-11	-3.94E-11
3.05E-11	8.42E-12	-1.57E-11	-3.65E-11	-2.88E-11
1.65E-11	7.38E-12	-2.66E-12	-1.34E-11	-1.96E-11
1.61E-12	7.76E-13	-1.52E-13	-1.17E-12	-2.02E-12
1.00E+00	2.30E-09	-5.30E-10	-2.16E-09	-2.02E-09
2.30E-09	1.00E+00	1.52E-09	-3.72E-10	-9.72E-10
-5.30E-10	1.52E-09	1.00E+00	1.59E-09	1.90E-10
-2.16E-09	-3.72E-10	1.59E-09	1.00E+00	1.47E-09
-2.02E-09	-9.72E-10	1.90E-10	1.47E-09	1.00E+00
-1.77E-07	-1.56E-07	-1.31E-07	-9.68E-08	-5.14E-08
-1.94E-08	-2.18E-07	-1.76E-07	-1.27E-07	-6.63E-08
1.03E-07	-7.28E-08	-2.44E-07	-1.70E-07	-8.75E-08
1.59E-07	3.29E-08	-1.06E-07	-2.37E-07	-1.18E-07
1.33E-07	6.39E-08	-1.25E-08	-9.65E-08	-1.66E-07
2.19E-04	5.51E-04	7.09E-04	6.72E-04	4.22E-04
-2.17E-04	5.58E-04	7.12E-04	6.75E-04	4.23E-04
-4.73E-04	1.23E-04	7.20E-04	6.78E-04	4.25E-04
-5.10E-04	-1.31E-04	2.86E-04	6.86E-04	4.27E-04
-3.47E-04	-1.67E-04	3.26E-05	2.52E-04	4.34E-04

Hessian for OFF mode columns 11:15:

4.13E-09	5.35E-09	5.51E-09	4.96E-09	3.29E-09
2.34E-09	3.10E-09	4.03E-09	3.76E-09	2.59E-09
1.18E-09	1.54E-09	2.06E-09	2.65E-09	1.90E-09
4.51E-10	5.84E-10	7.74E-10	1.05E-09	1.29E-09
4.10E-11	5.29E-11	6.99E-11	9.44E-11	1.33E-10
-1.77E-07	-1.94E-08	1.03E-07	1.59E-07	1.33E-07
-1.56E-07	-2.18E-07	-7.28E-08	3.29E-08	6.39E-08
-1.31E-07	-1.76E-07	-2.44E-07	-1.06E-07	-1.25E-08
-9.68E-08	-1.27E-07	-1.70E-07	-2.37E-07	-9.65E-08
-5.14E-08	-6.63E-08	-8.75E-08	-1.18E-07	-1.66E-07
1.00E+00	1.57E-05	1.07E-05	6.81E-06	3.38E-06
1.57E-05	1.00E+00	1.43E-05	8.92E-06	4.36E-06
1.07E-05	1.43E-05	1.00E+00	1.20E-05	5.75E-06
6.81E-06	8.92E-06	1.20E-05	1.00E+00	7.78E-06
3.38E-06	4.36E-06	5.75E-06	7.78E-06	1.00E+00
-8.36E-02	-7.43E-02	-6.29E-02	-4.79E-02	-2.77E-02
-5.53E-02	-7.50E-02	-6.31E-02	-4.81E-02	-2.78E-02
-3.51E-02	-4.66E-02	-6.37E-02	-4.83E-02	-2.80E-02
-2.02E-02	-2.64E-02	-3.53E-02	-4.88E-02	-2.81E-02
-8.81E-03	-1.14E-02	-1.50E-02	-2.03E-02	-2.85E-02

Hessian for OFF mode columns 16:20:

-2.36E-05	-2.34E-05	-2.01E-05	-1.53E-05	-8.58E-06
-1.51E-05	-1.52E-05	-1.49E-05	-1.17E-05	-6.76E-06
-8.47E-06	-8.50E-06	-8.54E-06	-8.27E-06	-4.95E-06
-3.61E-06	-3.62E-06	-3.64E-06	-3.66E-06	-3.36E-06
-3.37E-07	-3.38E-07	-3.39E-07	-3.41E-07	-3.46E-07
2.19E-04	-2.17E-04	-4.73E-04	-5.10E-04	-3.47E-04
5.51E-04	5.58E-04	1.23E-04	-1.31E-04	-1.67E-04
7.09E-04	7.12E-04	7.20E-04	2.86E-04	3.26E-05
6.72E-04	6.75E-04	6.78E-04	6.86E-04	2.52E-04
4.22E-04	4.23E-04	4.25E-04	4.27E-04	4.34E-04
-8.36E-02	-5.53E-02	-3.51E-02	-2.02E-02	-8.81E-03
-7.43E-02	-7.50E-02	-4.66E-02	-2.64E-02	-1.14E-02
-6.29E-02	-6.31E-02	-6.37E-02	-3.53E-02	-1.50E-02
-4.79E-02	-4.81E-02	-4.83E-02	-4.88E-02	-2.03E-02
-2.77E-02	-2.78E-02	-2.80E-02	-2.81E-02	-2.85E-02
3.62E+02	2.89E+02	2.17E+02	1.45E+02	7.24E+01
2.89E+02	2.91E+02	2.18E+02	1.45E+02	7.26E+01
2.17E+02	2.18E+02	2.20E+02	1.46E+02	7.29E+01
1.45E+02	1.45E+02	1.46E+02	1.47E+02	7.33E+01
7.24E+01	7.26E+01	7.29E+01	7.33E+01	7.44E+01

Using the above matrices, one can calculate the eigenvalues and confirm that the Hessians are positive definite.

Eigenvalues for the ON mode Hessian:

1.04E-04
1.38E-04
2.36E-04
6.85E-04
1.33E-02
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
2.97E+00
5.03E+00
1.34E+01
7.02E+01
1.31E+03
4.54E+06
1.08E+07
1.10E+07
1.11E+07
1.11E+07

Eigenvalues for the OFF mode Hessian:

1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
1.00E+00
2.09E+01
2.69E+01
4.39E+01
1.08E+02
8.95E+02

Appendix E MPC Guide

In this appendix a discussion is presented of how the quadratic efficiency objectives presented in this work are transformed using a linear discrete prediction model into a function of just the control sequence. This implementation guide only considers quadratic objective functions so all cost terms will be composed of proportional and squared terms of the outputs and inputs. The motivation for considering only quadratic objective functions is that this limits the complexity of the problem and for the linear/discrete prediction model, one is guaranteed to have a convex optimization problem and under some additional constraints a strictly convex problem. For this work, each of the quadratic efficiency objectives can be expressed as linear combinations of the four elements shown in Eq. (E.1) – (E.4).

$$c \cdot R(i) \cdot y_j(i) \qquad (E.1)$$

$$c \cdot y_j(i)\, y_k(i) \qquad (E.2)$$

$$c \cdot u_j(i)\, y_k(i) \qquad (E.3)$$

$$c \cdot u_j(i) \qquad (E.4)$$

Here c is a constant coefficient, R is a vector of values dependent on the time index (typically the reference trajectory), y is an output, u is an input, subscripts j and k are the indices of the variable (i.e. u_k is the k^{th} input and y_j is the j^{th} output), and i is the discrete time index. Below is a derivation of how each of the above terms can be expressed as a function of the control sequence and MATLAB code for generating each cost term. Throughout this discussion, the control sequence over the prediction horizon will be denoted by V, given by Eq. (E.5) where

N is the length of the prediction horizon. In addition, there are many terms throughout these derivations which take the form of Eq. (E.6). These terms should be interpreted as the summation from $i = 1$ to N of the r^{th} element of the j^{th} row of Z raised to the i^{th} power.

$$V = \left[u_1(0), u_1(1), \cdots, u_1(N-1), u_2(0), \cdots, u_2(N-1), \cdots, u_m(0), \cdots, u_m(N-1)\right]^T \quad \text{(E.5)}$$

$$\sum_{i=1}^{N} \left(Z^i\right)_j (r) \quad \text{(E.6)}$$

$$J = \sum_{i=1}^{N} c \cdot u_j(i-1)$$

Element:

$$J = \sum_{i=1}^{N} c \cdot u_j(i-1)$$

$$= c \cdot \left(u_j(0) + u_j(1) + u_j(2) + \cdots + u_j(N-1)\right)$$

Collect all terms which are proportional to $u(\cdot)$

$$F = \left[0^{1\times N(j-1)}, 1^{1\times N}, 0^{1\times N(m-j)}\right]$$

$$\rightarrow J = c \cdot F \cdot V$$

```
function[H,F]=MPC_H_F_for_c_ui(B,c,State1,n)
% Computes the H and F matrices associated with a cost term, c*u_state1(i),
% that results from expanding a quadratic objective function.  This term is
% summed from i = 0 to n-1.

[y,m] = size(B);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n
    F(i+(State1-1)*n) = 1;
end

F = c*F;
end
```

Element: $\sum_{i=1}^{N} c \cdot u_j(i-1) u_k(i-1)$

$$J = \sum_{i=1}^{N} c \cdot u_j(i-1) u_k(i-1)$$
$$= c \cdot \left(u_j(0) \cdot u_k(0) + u_j(1) \cdot u_k(1) + u_j(2) \cdot u_k(2) + \cdots + u_j(N-1) \cdot u_k(N-1) \right)$$

Collect all terms which include $u(\cdot) \cdot u(\cdot)$

assume $j > k$

$$H = \begin{bmatrix} \begin{bmatrix} & \\ & 0^{N(j-1)\times N(j-1)} \\ & \end{bmatrix} & & \begin{bmatrix} 0^{N(k-1)\times N} \\ 0.5 \cdot I^{N\times N} \\ 0^{N(j-k-1)\times N} \end{bmatrix} & \begin{bmatrix} \\ 0^{N(j-1)\times N(m-j)} \\ \end{bmatrix} \\ \begin{bmatrix} 0^{N\times N(k-1)} & 0.5 \cdot I^{N\times N} & {}^{N\times N(j-k-1)} \end{bmatrix} & & \begin{bmatrix} 0^{N(m-j+1)\times N(m-j+1)} \end{bmatrix} & \\ \begin{bmatrix} & \\ & 0^{N(m-j)\times N(j-1)} \\ & \end{bmatrix} & & & \end{bmatrix}$$

if $j = k$

$$H = \begin{bmatrix} 0^{N(j-1)\times N(j-1)} & 0^{N(j-1)\times N} & 0^{N(j-1)\times N(m-j)} \\ 0^{N\times N(j-1)} & I^{N\times N} & 0^{N\times N(m-j)} \\ 0^{N(m-j)\times N(j-1)} & 0^{N(m-j)\times N} & 0^{N(m-j)\times N(m-j)} \end{bmatrix}$$

$$\rightarrow J = c \cdot V^T \cdot H \cdot V$$

```
function[H,F]=MPC_H_F_for_c_ui_uj(B,c,State1,State2,n)
% Computes the H and F matrices associated with a cost term,
% c*u_state1(i)*u_state2(i), that results from expanding a quadratic
% objective function.  This term is summed from i = 0 to n-1.

[y,m] = size(B);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for k=1:n;
    H(k+(State1-1)*n,k+(State2-1)*n)= H(k+(State1-1)*n,k+(State2-1)*n)+1/2;
    H(k+(State2-1)*n,k+(State1-1)*n)= H(k+(State2-1)*n,k+(State1-1)*n)+1/2;
end

H = c*H;
end
```

$$J = \sum_{i=1}^{N} c \cdot y_j(i)$$

Element:

$$J = \sum_{i=1}^{N} c \cdot y_j(i)$$

$$= c\Big(\Big[C \cdot \big(A \cdot X(0) + G \cdot L + B \cdot U(0)\big)\Big]_j$$

$$+\Big[C \cdot \big(A \cdot \big(A \cdot X(0) + G \cdot L + B \cdot U(0)\big) + B \cdot U(1)\big)\Big]_j$$

$$+\Big[C \cdot \big(A \cdot \big(A \cdot \big(A \cdot X(0) + G \cdot L + B \cdot U(0)\big) + B \cdot U(1)\big) + B \cdot U(2)\big)\Big]_j$$

$$\vdots$$

$$+\Big[C \cdot \big(A^N \cdot X(0) + A^{N-1} \cdot G \cdot L + A^{N-1} \cdot B \cdot U(0) + A^{N-2} \cdot B \cdot U(1) + \cdots + B \cdot U(N-1)\big)\Big]_j\Big)$$

Collect all terms which are proportional to $u(\cdot)$

$$f_j(r) = \left[\sum_{i=1}^{N} \left(C \cdot A^{i-1} \cdot B\right)_j(r), \quad \sum_{i=1}^{N-1} \left(C \cdot A^{i-1} \cdot B\right)_j(r), \quad \cdots, \quad (C \cdot B)_j(r)\right]$$

$$F = \left[f_j(1), \quad f_j(2), \quad \cdots, \quad f_j(m)\right]$$

$$\rightarrow J \approx c \cdot F \cdot V$$

```
function[H,F]=MPC_H_F_for_c_yi(A,B,C,c,R,State,n)
% Computes the H and F matrices associated with a cost term, c*R(i)*y(i),
that
% results from expanding a quadratic objective function.  This term is summed
% from i = 1 to n.

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for i=1:m
```

```
    for j=1:n
        for k=1:n+1-j
            F(k+(i-1)*n)=F(k+(i-1)*n)+R(k+j-1)*Z(State,i+m*(j-1));
        end
    end
end

F = c*F;
end
```

$$J=\sum_{i=1}^{N} c\cdot y_j(i)\,y_k(i)$$

Element:

$$J=\sum_{i=1}^{N} c\cdot y_j(i)\,y_k(i)$$

$$\begin{aligned}
&=c\Big(\Big[C\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)\Big]_j\cdot\Big[C\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)\Big]_k\\
&+\Big[C\cdot\big(A\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)+B\cdot U(1)\big)\Big]_j\cdot\Big[C\cdot\big(A\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)+B\cdot U(1)\big)\Big]_k\\
&+\Big[C\cdot\big(A\cdot\big(A\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)+B\cdot U(1)\big)+B\cdot U(2)\big)\Big]_j\\
&\quad\cdot\Big[C\cdot\big(A\cdot\big(A\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)+B\cdot U(1)\big)+B\cdot U(2)\big)\Big]_k\\
&\vdots\\
&+\Big[C\cdot\big(A^N\cdot X(0)+A^{N-1}\cdot G\cdot L+A^{N-1}\cdot B\cdot U(0)+A^{N-2}\cdot B\cdot U(1)+\cdots+B\cdot U(N-1)\big)\Big]_j\\
&\quad\cdot\Big[C\cdot\big(A^N\cdot X(0)+A^{N-1}\cdot G\cdot L+A^{N-1}\cdot B\cdot U(0)+A^{N-2}\cdot B\cdot U(1)+\cdots+B\cdot U(N-1)\big)\Big]_k\Big)
\end{aligned}$$

Collect all terms which include $u(\cdot)\cdot u(\cdot)$

$$Q_{j,k}=\big(C\cdot A^{i+z-1}\cdot B\big)_j^T\big(C\cdot A^{i-1}\cdot B\big)_k+\big(C\cdot A^{i+z-1}\cdot B\big)_k^T\big(C\cdot A^{i-1}\cdot B\big)_j$$

$$h_{j,k}(r,p)=\begin{bmatrix}
\frac{1}{2}\sum_{i=1}^{N}Q_{j,k}(r,p)\Big|_{z=0} & \frac{1}{2}\sum_{i=1}^{N-1}Q_{j,k}(r,p)\Big|_{z=1} & \cdots & \frac{1}{2}\sum_{i=1}^{2}Q_{j,k}(r,p)\Big|_{z=N-2} & \frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=N-1}\\
\frac{1}{2}\sum_{i=1}^{N-1}Q_{j,k}(r,p)\Big|_{z=1} & \frac{1}{2}\sum_{i=1}^{N-1}Q_{j,k}(r,p)\Big|_{z=0} & \cdots & \frac{1}{2}\sum_{i=1}^{2}Q_{j,k}(r,p)\Big|_{z=N-3} & \frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=N-2}\\
\vdots & \vdots & \ddots & \vdots & \vdots\\
\frac{1}{2}\sum_{i=1}^{2}Q_{j,k}(r,p)\Big|_{z=N-2} & \frac{1}{2}\sum_{i=1}^{2}Q_{j,k}(r,p)\Big|_{z=N-3} & \cdots & \frac{1}{2}\sum_{i=1}^{2}Q_{j,k}(r,p)\Big|_{z=0} & \frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=1}\\
\frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=N-1} & \frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=N-2} & \cdots & \frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=1} & \frac{1}{2}Q_{j,k}(r,p)\Big|_{i=1,z=0}
\end{bmatrix}$$

$$H=\begin{bmatrix}
h_{j,k}(1,1) & \cdots & h_{j,k}(1,m)\\
\vdots & \ddots & \vdots\\
h_{j,k}(m,1) & \cdots & h_{j,k}(m,m)
\end{bmatrix}$$

Collect all terms which are proportional to $u(\cdot)$

$$w_{j,k} = \left(C \cdot A^{i} \cdot X(0) + C \cdot A^{i-1} \cdot G \cdot L\right)_j \cdot \left(C \cdot A^{i-1} \cdot B\right)_k + \left(C \cdot A^{i} \cdot X(0) + C \cdot A^{i-1} \cdot G \cdot L\right)_k \cdot \left(C \cdot A^{i-1} \cdot B\right)_j$$

$$f_{j,k}(r)\left[\sum_{i=1}^{N} w_{j,k}(r), \quad \sum_{i=1}^{N-1} w_{j,k}(r), \quad \cdots, \quad w_{j,k}\Big|_{i=0}(r)\right]$$

$$F = F = \left[f_{j,k}(1), \quad f_{j,k}(2), \quad \cdots, \quad f_{j,k}(m)\right]$$

$$\rightarrow J \approx c \cdot \left(V^{T} \cdot H \cdot V + F \cdot V\right)$$

```
function[H,F]=MPC_H_F_for_c_yi_yj(A,B,C,Xo,c,State1,State2,n)
% Computes the H and F matrices associated with a cost term,
% Integral(c*y1(i)*y2(i)), that results from expanding a quadratic objective
% function.  This term is summed from i = 1 to n.

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end


for j=1:m
    for k=1:n
        for l=1:n+1-k
            q=C*A^(l+k-1);
            for i = 1:r
                F(l+(j-1)*n)=F(l+(j-1)*n)
                  +q(State1,i)*Xo(i)*Z(State2,j+m*(k-1))
                  +Z(State1,j+m*(k-1))*q(State2,i)*Xo(i);
            end

            for a=1:m
                H(l+(j-1)*n,l+(a-1)*n) = H(l+(j-1)*n,l+(a-1)*n)
                  +(1/2)*(Z(State1,j+m*(k-1))*Z(State2,a+m*(k-1))
                  +Z(State2,j+m*(k-1))*Z(State1,a+m*(k-1)));

                if l >=2
                    for o=1:l-1
```

```
                        H(o+(j-1)*n,l+(a-1)*n)= H(o+(j-1)*n,l+(a-1)*n)
                          +(1/2)*(Z(State1,j+m*(l+k-1-o))*Z(State2,a+m*(k-1))
                          +Z(State2,j+m*(l+k-1-o))*Z(State1,a+m*(k-1)));
                        H(l+(j-1)*n,o+(a-1)*n)= H(l+(j-1)*n,o+(a-1)*n)
                          +(1/2)*(Z(State1,j+m*(k-1))*Z(State2,a+m*(l+k-1-o))
                          +Z(State2,j+m*(k-1))*Z(State1,a+m*(l+k-1-o)));
                    end
                end
            end
        end
    end
end


F = c*F;
H = c*H;
End
```

$$J=\sum_{i=1}^{N} c\cdot u_j(i-1)\,y_k(i-1)$$

Element:

$$J=\sum_{i=1}^{N} c\cdot u_j(i-1)\,y_k(i-1)$$
$$=c\Big(\Big[\big(U(0)\big)\Big]_j\cdot\Big[C\cdot X(0)\Big]_k$$
$$+\Big[\big(U(1)\big)\Big]_j\cdot\Big[C\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)\Big]_k$$
$$+\Big[\big(U(2)\big)\Big]_j\cdot\Big[C\cdot\big(A\cdot\big(A\cdot X(0)+G\cdot L+B\cdot U(0)\big)+B\cdot U(1)\big)\Big]_k$$
$$\vdots$$
$$+\Big[U(N-1)\Big]_j\cdot\Big[C\cdot\big(A^{N-1}\cdot X(0)+A^{N-2}\cdot G\cdot L+A^{N-2}\cdot B\cdot U(0)+A^{N-3}\cdot B\cdot U(1)+\cdots+B\cdot U(N-2)\big)\Big]_k\Big)$$

Collect all terms which include $u(\cdot)\cdot u(\cdot)$

$$Q_{j,k} = \left[0^{1\times j-1}, 1, 0^{1\times m-j}\right]^T \cdot \left(C\cdot A^{i-1}\cdot B\right)_k$$

$$h_{j,k}(r,p) = \begin{bmatrix} 0 & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=1} & \cdots & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=N-2} & \frac{1}{2}Q_{j,k}(r,p)\big|_{i,z=N-1} \\ \frac{1}{2}Q_{j,k}(r,p)\big|_{i=1} & 0 & \cdots & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=N-3} & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=0,z=N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{2}Q_{j,k}(r,p)\big|_{i=N-2} & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=N-3} & \cdots & 0 & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=1} \\ \frac{1}{2}Q_{j,k}(r,p)\big|_{i=N-1} & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=N-2} & \cdots & \frac{1}{2}Q_{j,k}(r,p)\big|_{i=1} & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} h_{j,k}(1,1) & \cdots & h_{j,k}(1,m) \\ \vdots & \ddots & \vdots \\ h_{j,k}(m,1) & \cdots & h_{j,k}(m,m) \end{bmatrix}$$

Collect all terms which are proportional to $u(\cdot)$

$$f_k = \left(C\cdot A^{i-1}\cdot X(0) + C\cdot A^{i-2}\cdot G\cdot L\right)_k$$

$$F = \left[0^{1\times N(j-1)}, \left(C\cdot X(0)\right)_k, f_k\big|_{i=2}, f_k\big|_{i=3}, \cdots, f_k\big|_{i=N}, 0^{1\times N(m-j)}\right]$$

From the above derivations one can see that after all terms which are independent of the input vector are removed, each term of the objective function results in an F vector, an H matrix, or a combination of the two. Therefore, the final objective function is simply the summation of these terms as shown in Eq. (E.7).

$$J = V^T\cdot\left(\sum_{i=1} H_i\right)\cdot V + \left(\sum_{i=1} F_i\right)\cdot V \qquad \text{(E.7)}$$

Here H_i and F_i are the i^{th} H matrix and F vector resulting from the decomposition of the objective function.

Appendix F AEVPS Operation Guide

The Augmented Earthmoving Vehicle Powertrain Simulator (AEVPS) is a complex hydro-mechanical powertrain with energy storage and computer control. Below is a discussion of how to operate this system and setup MATLAB based control of the system. The structure and major components of this experimental system were described in Section 3.1. Here a more detailed description of the operation is presented. Specifically, instructions are given for how to turn the system on/off, how the electrical connections are configured, and how to setup computer control. In addition to the electric motor, variable displacement pump, gas charged accumulator, valve manifold, and three load units, the AEVPS also has two computers for controlling the system and manual valves for changing the physical architecture. One computer, denoted $Comp_{eng}$, is used exclusively for engine control. $Comp_{eng}$ uses a model of an internal combustion engine to predict the engine speed based on throttle command and loading. This engine speed is then used as a reference for the electric motor. This computer uses WinCon to generate a real time executable program from Simulink models. The second computer, $Comp_{pow}$, is a dSPACE box which is used to run the energy management strategy and vehicle load emulation. Real time executable programs are generated from a Simulink model and downloaded onto the dSPACE box. Figure F.1 shows the dSPACE box.

Figure F.1: dSPACE box

The AEVPS can be setup in two different physical configurations through manipulation of a pair of manual valves. **It is very important that these valves be setup properly before turning on the hardware**. Failure to check the valve configuration could result in an incomplete hydraulic circuit which may lead to over pressurization and damage of components. In the configuration shown on the left of Fig. F.2, the gas charged accumulator is isolated from the circuit and there is no storage. In the configuration shown on the right Fig. F.2, the gas charged accumulator is part of the circuit and energy storage is possible. The manual valve directly below the accumulator goes directly to the tank and is used to drain the accumulator between experiments. During operation it should be closed (handle is perpendicular to the drain hose). To open the valve, move the handle so that it is parallel with the drain hose. Fig. F.3 shows the drain valve configurations. Properly positioning the manual valves is the only physical setup required for the AEVPS. However, it is always good practice to open and close the drain valve before an experiment, in case the accumulator has pressurized fluid within it.

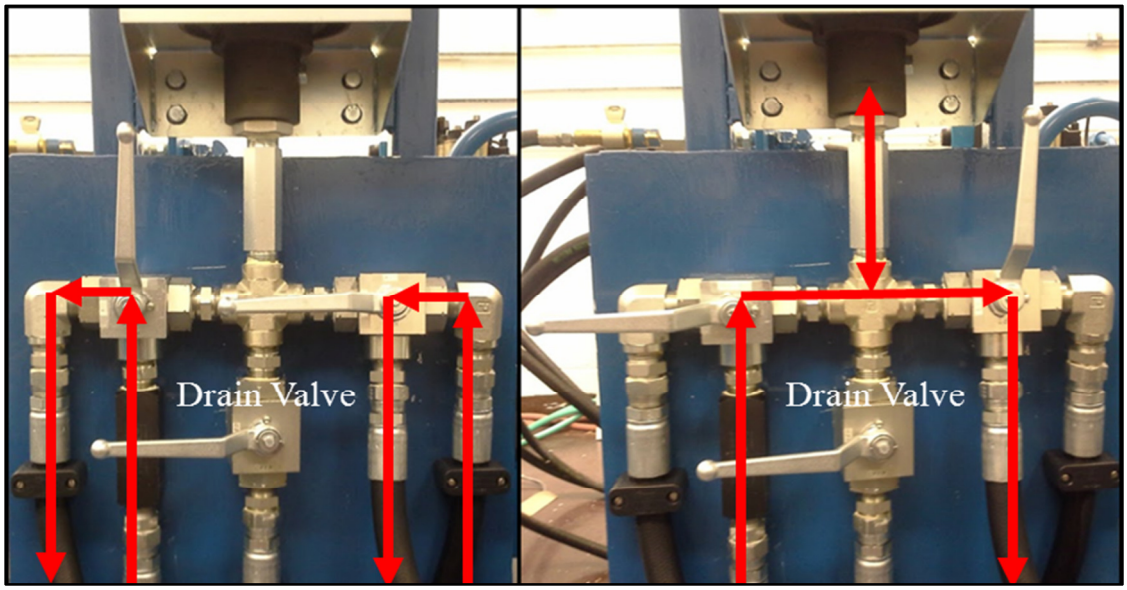

Figure F.2: Manual valve bank; left: no accumulator, right: accumulator

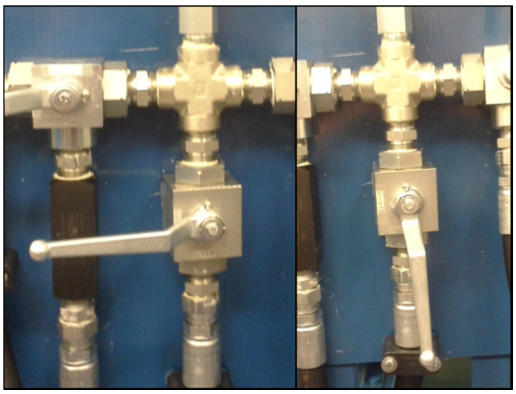

Figure F.3: Drain valve; left is closed, right is open

To turn on the system, there are a series of components which must be turned on in sequence. Note that these steps assume that the computers are already on and the real time executable controllers have already been generated. Instructions for using the computers and setting up the controllers are provided later. The following sequence is used to turn on the AEVPS.

1. Turn the control cabinet on by turning the large lever clockwise to the "On" position. See Fig. F.4 for an image of the lever.
2. Switch up the "Drive Enable" toggle switch, shown in Fig. F.5.
3. Turn on the amplifier shown in Fig. F.6.

Once these steps are complete the AEVPS is ready to receive inputs from the computers. When activating computer control, it is recommended that the program running on $Comp_{eng}$ be started first and then $Comp_{pow}$. To shut down the AEVPS, the same steps should be run in reverse.

Figure F.4: Control cabinet and power lever

Figure F.5: "Drive Enable" toggle switch on control cabinet

Figure F.6: Amplifier

Between the AEVPS, $Comp_{eng}$, and $Comp_{pow}$ there are many electrical connections for transferring inputs and measurements. Figures F.7 - F.10 show schematics of the input and output connections of each component. Table F.1 is a list of each input to the AEVPS and its associated number. Table F.2 is a list of each measurement and its associated number.

Table F.1: AEVPS Input signals

1	AC motor speed	16	Pressure relief valve command for load unit 2
9	Displacement Command	19	Valve command for load unit 3
12	Valve command for load unit 1	20	Pressure relief valve command for load unit 3
13	Pressure relief valve command for load unit 1	31	Engine load estimate
15	Valve command for load unit 2		

Table F.2: AEVPS measurement signals

2	AC motor torque	11	Pressure downstream of valve for load unit 1
3/4	AC motor speed	14	Hydraulic motor speed for load unit 1
5	Pressure of load unit 1	17	Pressure downstream of valve for load unit 2
6	Pressure of load unit 2	18	Hydraulic motor speed for load unit 2
7	Pressure of load unit 3	21	Pressure downstream of valve for load unit 3
8	Pump displacement	22	Hydraulic motor speed for load unit 3
10	Upstream pressure		

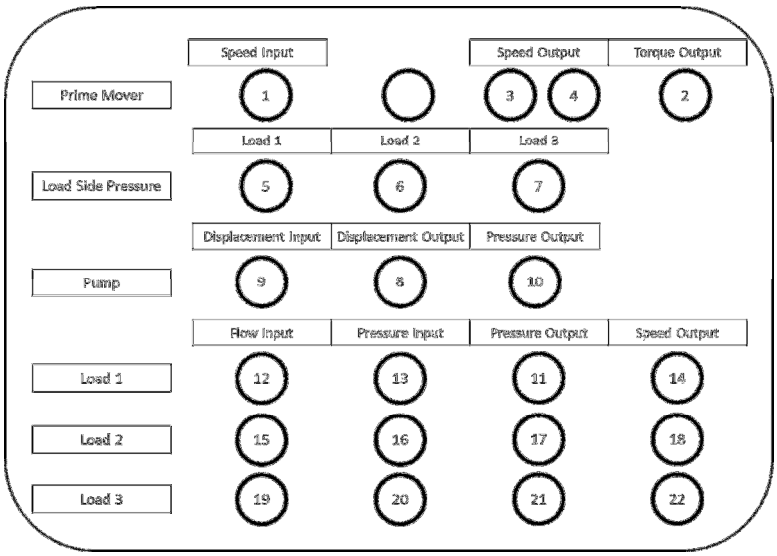

Figure F.7: AEVPS electric connection schematic

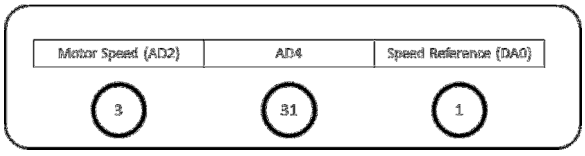

Figure F.8: $Comp_{eng}$ electric connection schematic

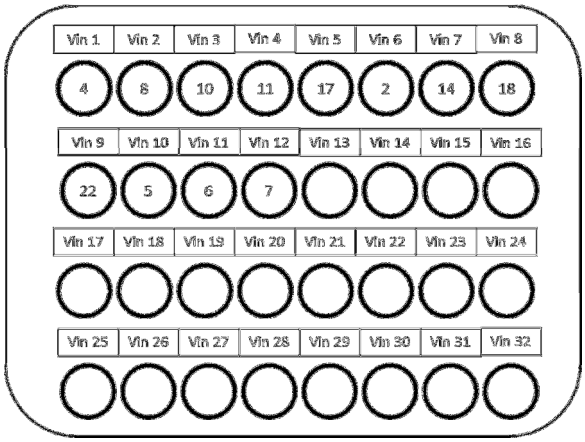

Figure F.9: $Comp_{pow}$ electric input connection schematic

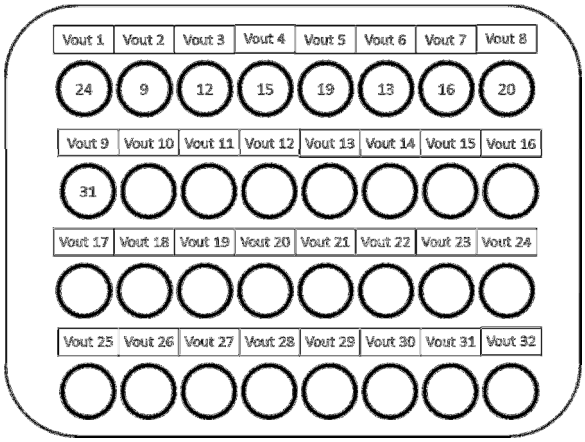

Figure F.10: $Comp_{pow}$ electric output connection schematic

The final component of properly using the AEVPS is to setup the control strategies. As discussed previously, $Comp_{eng}$ is used for engine emulation. The Simulink model "AEVPS_Eng_Emulation", which can be accessed from the desktop, contains an engine model with the appropriate input/output blocks. Figure F.11 shows the Simulink block diagram. This model takes in a throttle command from $Comp_{pow}$, the AC motor speed measurement, and an estimate of the load torque on the emulated engine. The output of this model is the desired speed of the AC motor. To produce a real time executable program, click on "Build" from the WinCon dropdown menu, see Fig. F.12. Once this program is produced, the WinCon toolbar shown in Fig. F.13 will appear. From this toolbar one can open plot windows of signals and then save that recorded data for offline analysis. The signals of most interest are "Speed (rad/sec)" which is the AC motor speed measurement, "Speed_sim (rad/sec)" is the reference speed from the engine model, and "Torque (N-m) is the estimated load torque from $Comp_{pow}$. The green "START" button is used to launch the engine emulation.

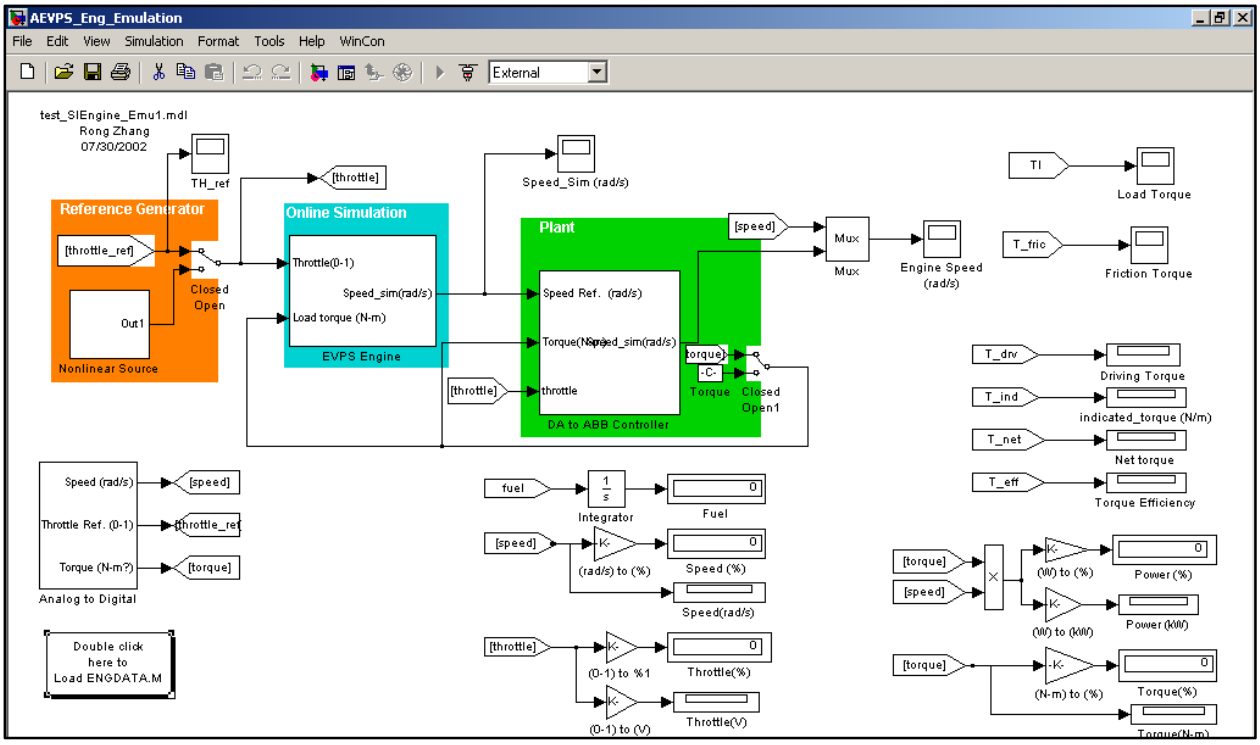

Figure F.11: Simulink model used for engine emulation

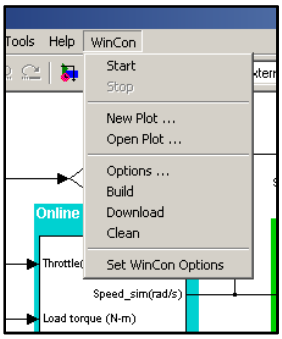

Figure F.12: WinCon dropdown menu

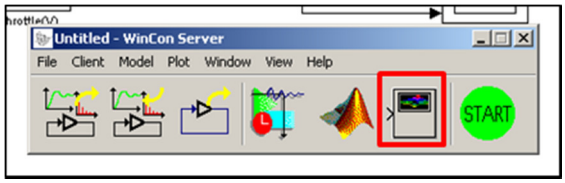

Figure F.13: WinCon Server toolbar, the plot icon is outlined

The powertrain control and load emulation are handled by $Comp_{pow}$, the dSPACE box. To use $Comp_{pow}$ one must first build a Simulink model and download it using the ControlDesk

software. ControlDesk is the software interface for managing the dSPACE box. Before turning on the computer connected to the dSPACE box, the box must be on. Once the computer is on, launch ControlDesk from the start menu, see Fig. F.14, and then open MATLAB. Navigate the MATLAB working directory to the folder "AEVPS_Start" located on the desktop. In this folder there are two files "AEVPS_basic_interface.mdl" and "Vehicle_Parameters.m", The m-file defines parameters for the load emulation and should be executed before running any program in which load emulation is needed. Figure F.15 shows the Simulink model titled "AEVPS_basic_interface". This model can be used as the bases for any controller interacting with the AEVPS. Simply, add the controller to this model and route signals to the appropriate inputs and outputs. The dSPACE I/O blocks are configured for the connections shown in Fig. F.9 and Fig. F.10. One can add or remove inputs and outputs through the dSPACE I/O GUI. The GUI can be accessed by double clicking on the "Outputs" and "Inputs" Simulink blocks. Figure F.16 shows the Simulink blocks and GUI's for the dSPACE I/O's. To download a model to dSPACE access the "Tools" dropdown menu, select "Real-Time Workshop", and then "Build Model…". Figure F.17 shows the menu sequence.

From Fig. F.15 one can see that a manual switch is included in this model. This provides a software means of shutting down the AEVPS. This signal gives a throttle and displacement command of zero and opens the valve connected to load unit 1 (if other load units are needed copy the appropriate block for their inputs as well). The valve is opened to allow the accumulator to drain. It is recommended that this switch be used to shut down the AEVPS and drain the accumulator before turning off the physical components. If the physical components are turned off, without draining the accumulator, then the manual drain valve can be used to empty the accumulator.

Figure F.14: ControlDesk launch from start menu

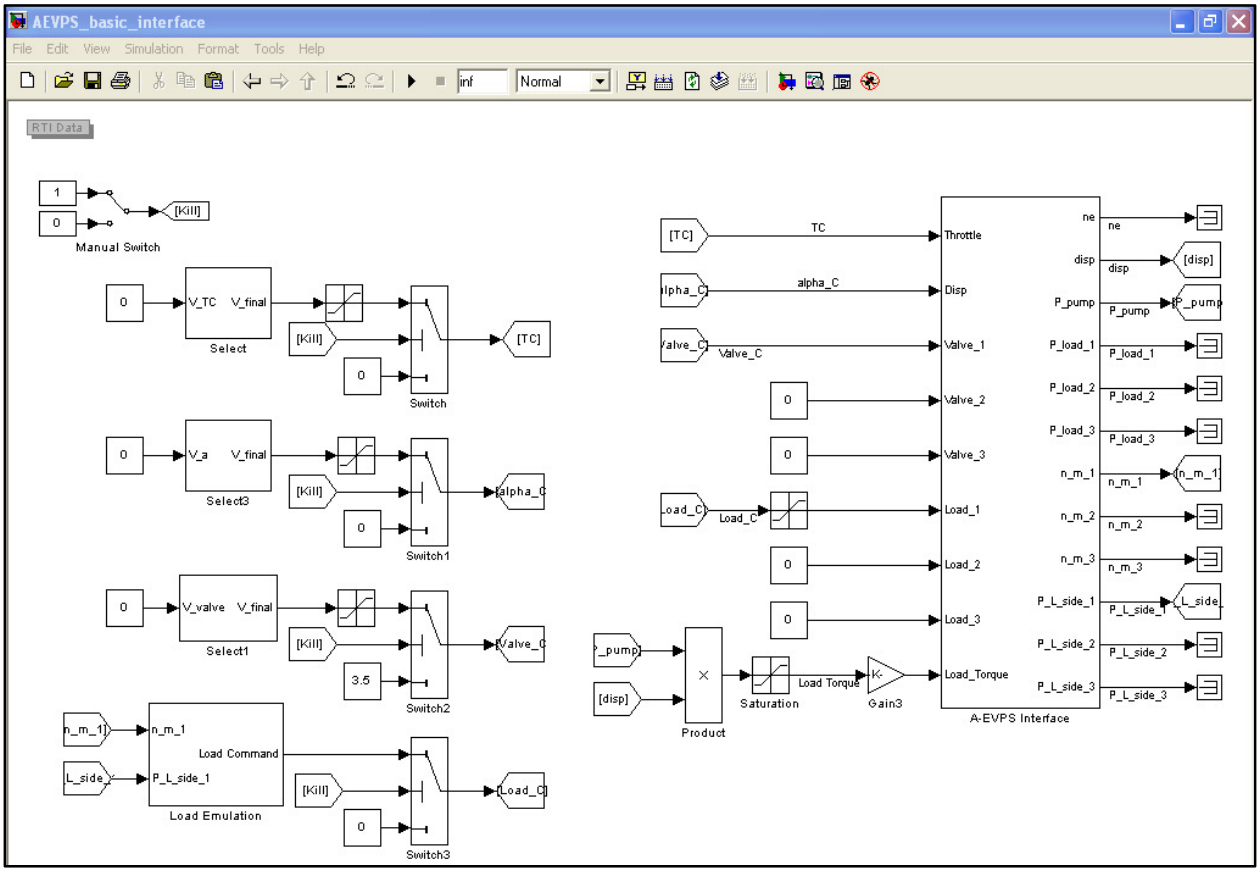

Figure F.15: Simulink diagram with dSPACE I/O connections and load emulation

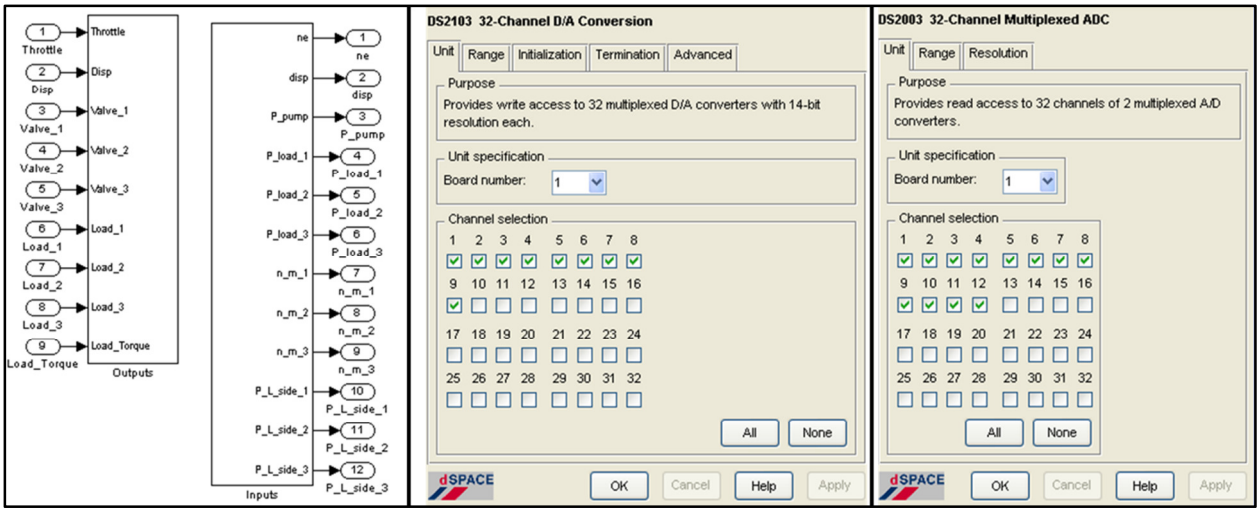

Figure F.16: dSPACE I/O blocks and GUI, left: Simulink blocks, center: Outputs GUI, right: Inputs GUI

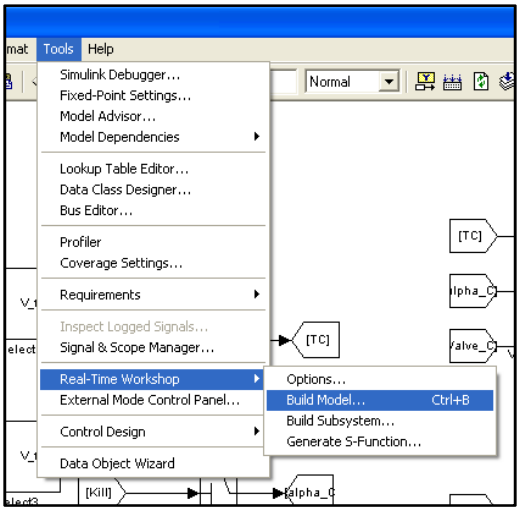

Figure F.17: $Comp_{pow}$ build dropdown menu

Once the program is built and downloaded to the dSPACE box, the program is managed through the ControlDesk interface. This interface is shown in Fig. F.18 along with a list of important features. If a program is running on the dSPACE box, there will be a green triangle next to the ds1005 icon. To view/save data coming into the dSPACE box and enable manual inputs, a layout must first be created. Once a new layout is started, the first tool which should be added is the "CaptureSettings" tool. This can be accessed through the "Data Acquisition" tab. This tool allows one to specify the time length over which data is captured, the downsampling of

the data, and which variables are captured. To configure these settings, one must enter animation mode using the mode selection buttons. The default mode is edit mode which is used to edit the layout. Animation mode is used when collecting data so one can see the graphs evolve in real time. To configure the "CaptureSettings" tool, choose the model from the dropdown menu.

Another common component of a layout is the "PlotterArray" . This tool is used to plot data. Once the tool is on the layout, variable tags can be dragged and dropped onto the tool to plot them. One "PlotterArray" can be used to plot multiple variables. To stop a program that is running on the dSPACE box, simply click the red square. Finally, when shutting down $comp_{pow}$ remember to turn of the computer first, and then the dSPACE box.

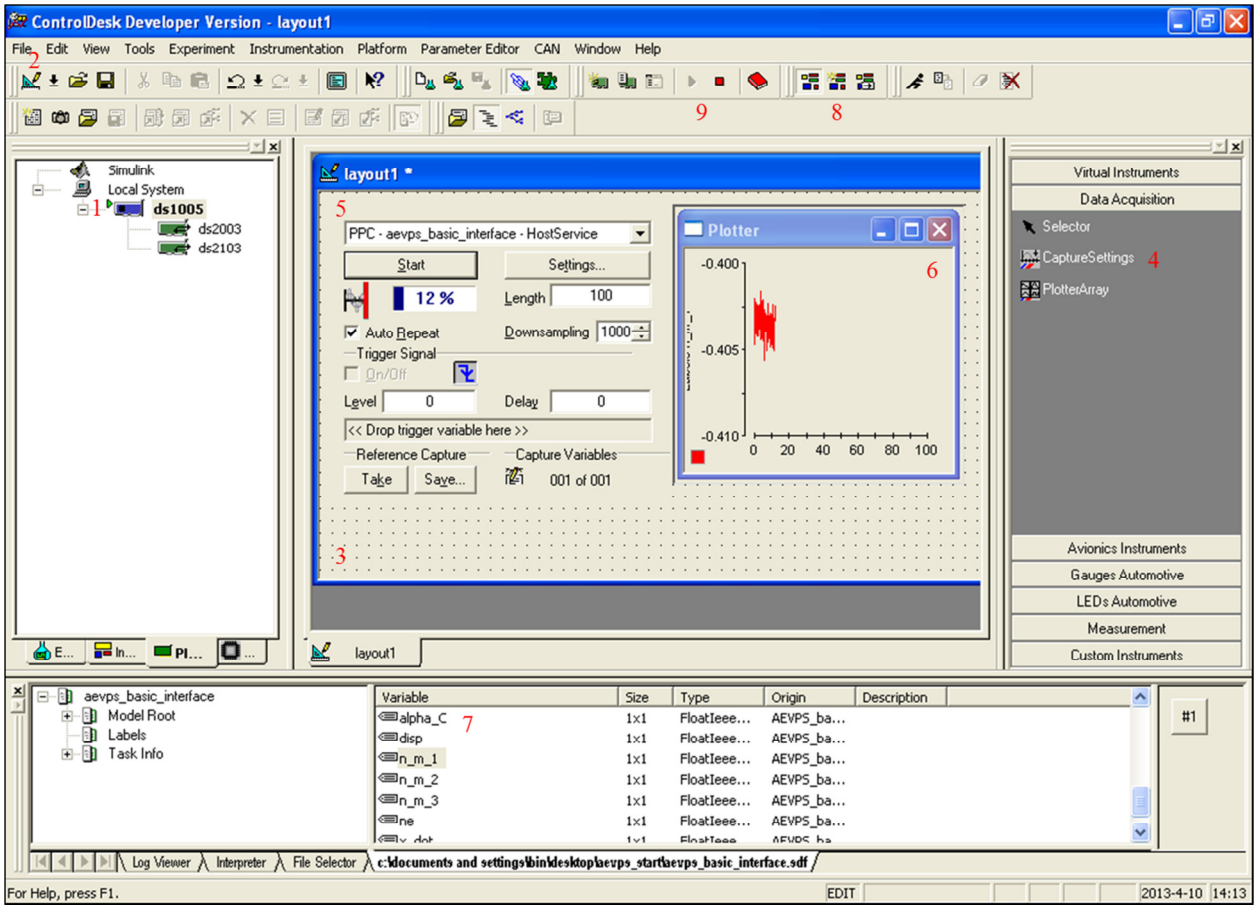

Figure F.18: ControlDesk GUI

1. Indicator that program is running on dSPACE box
2. Add new layout button

3. Layout where graphical interface for sending inputs and collecting data
4. Data acquisition tools
5. CaptureSettings tool
6. PlotterArray tool
7. Variable tags
8. Mode selection (Edit, Test, Animation)
9. Stop/Play buttons

Appendix G Hydraulic Hybrid EMS Code

Below is the MATLAB code used for the embedded MATLAB block used to define the MPC in the hybrid vehicle study. Note that for this system, the states are the outputs and the references values are constant throughout the prediction horizon. This code also includes code for a Newton's method solver of the optimization problem.

```
function [u1,u2,u3] = fcn(n_m_des,C,a_o,uv_o,n_e_o,P_u_o,P_d_o,a11,a12,a13,a14,a21,
                          a22,a23,a24,a31,a32,a33,a34,a41,a42,a43,a44,b11,b12,b13,
                          b21,b22,b23,b31,b32,b33,b41,b42,b43, n, Lambda1, Lambda2,
                          Plow, Phigh, Lambda3, Lambda4, n_m_max, n_e_max, T_e_max,
                          pu_max)
%#eml
K_P = 37.4;

T_e_o = K_P*a_o*P_u_o+(121/(1185*0.8))*((2+0.048*((220/188.5)*(30/pi)*n_e_o)
        +1.0336e-05*((220/188.5)*(30/pi)*n_e_o)^2)*.014*53.05)+0.2407*n_e_o;

if T_e_o >= 120
    T_e_o = 120;
End

if C == 1
    n_e_des1 = 109.4-n_e_o;
    T_des1 = 91.793-T_e_o;
    n_e_des2 = 76-n_e_o;
    T_des2 = 96.8-T_e_o;
    pu_des = 7-P_u_o;
else
    n_e_des1 = 60-n_e_o;
    T_des1 = 0-T_e_o;
    n_e_des2 = 60-n_e_o;
    T_des2 = 0-T_e_o;
    pu_des = 7-P_u_o;
    Lambda2 = 1;
    Lambda3 = 0;
    Lambda4 = 0;
End

%% Initialization and declaration %%
QP1 = [-T_e_o*ones(n,1);-a_o*ones(n,1);-uv_o*ones(n,1)];
QP2 = [(121-T_e_o)*ones(n,1);(0.314-a_o)*ones(n,1);(5-uv_o)*ones(n,1)];
A=[a11,a12,a13,a14;a21,a22,a23,a24;a31,a32,a33,a34;a41,a42,a43,a44];
B=[b11,b12,b13;b21,b22,b23;b31,b32,b33;b41,b42,b43];
```

```
y,m] = size(B);
Xo=zeros(y,1);
[Za,Zb]=Za_Zb_pressure_bounds_obj(A,B,Xo,Plow-P_u_o,Phigh-P_u_o,n);
IC = zeros(m*n,1);
U=zeros(m*n,1);
H=zeros(m*n,m*n);
F=zeros(1,m*n);
EG = zeros(m*n,1);
%% End %%

%% Execution %%
[H,F,P_star]=H_F_obj_AEVPS_full(A,B,Xo,n_m_des,n_e_des1,n_e_des2,T_des1,T_des2,
                                pu_des,Lambda1,Lambda2,Lambda3,Lambda4,n_m_max,
                                n_e_max,T_e_max,pu_max,n);
%% Newton's Method%%
[U]=Newton_Opt_Quad_boundries_AEVPS(2*H,F,IC,QP1,QP2,Za,Zb,P_star,2,0.01,0.5,
                                    1e-3,100,100,100);
%% End %%

%% Output %%
u1 =U(1)+T_e_o;
u2 =U(n+1)+a_o;
u3=U(2*n+1)+uv_o;
%% End %%
end

function[Za,Zb]=Za_Zb_pressure_bounds_obj(A,B,Xo,Plow,Phigh,n)
% Computes the Za and Zb matrices for the Pressure bounded between Plow and Phigh.

[Za1,Zb1]=MPC_Constraint_for_xi_greater_than_c(A,B,Xo,Plow,2,n);
[Za2,Zb2]=MPC_Constraint_for_xi_less_than_c(A,B,Xo,Phigh,2,n);

Za = [Za1;Za2];
Zb = [Zb1;Zb2];
end

function[Za,Zb]=MPC_Constraint_for_xi_greater_than_c(A,B,Xo,c,State,n)
% Computes the Za and Zb matrices associated with a constraint term, x_state(i)<c.
% This term is summed from i = 1 to n

[y,m] = size(B);

C = zeros(y,m*n);

Za = zeros(n,m*n);
Zb = -c*ones(n,1);

for i=1:n;
    for j=1:y;
        for k=1:m;
            Q = (A^(i-1))*B;
            C(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    for k=1:n
        for l=1:n+1-k
            Za(l+k-1,l+(j-1)*n) = -C(State,j+(k-1)*m);
        end
```

```
    end
end

for i=1:n
    for j=1:y
        q=A^(i);
        Zb(i) = Zb(i)+q(State,j)*Xo(j);
    end
end
end

function[Za,Zb]=MPC_Constraint_for_xi_less_than_c(A,B,Xo,c,State,n)
% Computes the Za and Zb matrices associated with a constraint term, x_state(i)<c.
% This term is summed from i = 1 to n

[y,m] = size(B);

C = zeros(y,m*n);

Za = zeros(n,m*n);
Zb = c*ones(n,1);

for i=1:n;
    for j=1:y;
        for k=1:m;
            Q = (A^(i-1))*B;
            C(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    for k=1:n
        for l=1:n+1-k
            Za(l+k-1,l+(j-1)*n) = C(State,j+(k-1)*m);
        end
    end
end

for i=1:n
    for j=1:y
        q=A^(i);
        Zb(i) = Zb(i)-q(State,j)*Xo(j);
    end
end
end

function[H,F,P_star]=H_F_obj_AEVPS_full(A,B,Xo,n_m_des,n_e_des1,n_e_des2,T_des1,
                                        T_des2,pu_des,Lambda1,Lambda2,Lambda3,
                                        Lambda4,n_m_max,n_e_max,T_e_max,pu_max,n)
% Computes the H and F matrices for the following objective function:
% J = Lambda1*((n_m-n_m_des)/n_m_max)^2
%     +Lambda2*((n_e-n_e_des1)/n_e_max)^2+Lambda2*((T_eng-T_des1)/T_e_max)^2
%     +Lambda3*((n_e-n_e_des1)/n_e_max)^2+Lambda2*((T_eng-T_des2)/T_e_max)^2
%     +Lambda4*((uv-uv_des)/uv_max)^2
% Note all terms are with respect to delta variables, delta_x = x-x_o
%P_star is minimum cost associated with H F matrices for quadratic
%objective function.  Assume J=0 is achievable
%%
P_star = 0;
%%
```

```
[H1,F1]=MPC_H_F_for_c_xi_xj_Integ(A,B,Xo,1,4,4,n);
[H2,F2]=MPC_H_F_for_c_xi_Integ(A,B,-2*n_m_des,4,n);
%%
[H3,F3]=MPC_H_F_for_c_xi_xj_Integ(A,B,Xo,1,1,1,n);
[H4,F4]=MPC_H_F_for_c_xi_Integ(A,B,-2*n_e_des1,1,n);
%%
[H5,F5]=MPC_H_F_for_c_ui_uj_Integ(B,1,1,1,n);
[H6,F6]=MPC_H_F_for_c_ui_Integ(B,-2*T_des1,1,n);
%%
[H7,F7]=MPC_H_F_for_c_xi_xj_Integ(A,B,Xo,1,1,1,n);
[H8,F8]=MPC_H_F_for_c_xi_Integ(A,B,-2*n_e_des2,1,n);
%%
[H9,F9]=MPC_H_F_for_c_ui_uj_Integ(B,1,1,1,n);
[H10,F10]=MPC_H_F_for_c_ui_Integ(B,-2*T_des2,1,n);
%%
[H11,F11]=MPC_H_F_for_c_xi_xj_Integ(A,B,Xo,1,2,2,n);
[H12,F12]=MPC_H_F_for_c_xi_Integ(A,B,-2*pu_des,2,n);
%%
H = Lambda1*(1/n_m_max^2)*(H1+H2)+Lambda2*(1/n_e_max^2)*(H3+H4)
    +Lambda2*(1/T_e_max^2)*(H5+H6)+Lambda3*(1/n_e_max^2)*(H7+H8)
    +Lambda3*(1/T_e_max^2)*(H9+H10)+Lambda4*(1/pu_max^2)*(H11+H12);

F = Lambda1*(1/n_m_max^2)*(F1+F2)+Lambda2*(1/n_e_max^2)*(F3+F4)
    +Lambda2*(1/T_e_max^2)*(F5+F6)+Lambda3*(1/n_e_max^2)*(F7+F8)
    +Lambda3*(1/T_e_max^2)*(F9+F10)+Lambda4*(1/pu_max^2)*(F11+F12);
%%
for i=1:n
    Ao=A^(n);
    P_star = P_star-Lambda1*(Ao(4,:)*Xo-n_m_des)^2-Lambda2*(Ao(1,:)*Xo-n_e_des1)^2
             -Lambda2*T_des1^2-Lambda3*(Ao(1,:)*Xo-n_e_des2)^2-Lambda3*T_des2^2
             -Lambda4*(Ao(2,:)*Xo-pu_des)^2;
end
end

function[H,F]=MPC_H_F_for_c_xi_xj_Integ(A,B,Xo,c,State1,State2,n)
% Computes the H and F matrices associated with a cost term,
% Integral(c*x_state1(i)*x_state2(i)), that results from expanding a
% quadratic objective function.  This term is summed from i = 1 to n.

[y,m] = size(B);

C = zeros(y,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:y;
        for k=1:m;
            Q = (A^(i-1))*B;
            C(j,k+(i-1)*m) = Q(j,k);
        end
    end
end


for j=1:m
    for k=1:n
        for l=1:n+1-k
            q=A^(l+k-1);
            for i = 1:y
```

```
                F(l+(j-1)*n)=F(l+(j-1)*n)
                            +(n+2-k-l)*q(State1,i)*Xo(i)*C(State2,j+m*(k-1))
                            +(n+2-k-l)*C(State1,j+m*(k-1))*q(State2,i)*Xo(i);
            end

            for a=1:m
                H(l+(j-1)*n,l+(a-1)*n) = H(l+(j-1)*n,l+(a-1)*n)+(n+2-k-l)*(1/2)*
    (C(State1,j+m*(k-1))*C(State2,a+m*(k-1))+C(State2,j+m*(k-1))*C(State1,a+m*(k-1)));

                if l >=2
                    for o=1:l-1
                        H(o+(j-1)*n,l+(a-1)*n)= H(o+(j-1)*n,l+(a-1)*n)
                                                +(n+2-k-l)*(1/2)*
(C(State1,j+m*(l+k-1-o))*C(State2,a+m*(k-1))+C(State2,j+m*(l+k-1-o))*C(State1,a+m*(k-
1)));
                        H(l+(j-1)*n,o+(a-1)*n)= H(l+(j-1)*n,o+(a-1)*n)
                                                +(n+2-k-l)*(1/2)*
(C(State1,j+m*(k-1))*C(State2,a+m*(l+k-1-o))+C(State2,j+m*(k-1))*C(State1,a+m*(l+k-1-
o)));
                    end
                end
            end
        end
    end
end


F = c*F;
H = c*H;
end

function[H,F]=MPC_H_F_for_c_xi_Integ(A,B,c,State,n)
% Computes the H and F matrices associated with a cost term, Integral(C*x_state(i)),
% that results from expanding a quadratic objective function.  This term is summed
% from i = 1 to n.

[y,m] = size(B);

C = zeros(y,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:y;
        for k=1:m;
            Q = (A^(i-1))*B;
            C(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for i=1:m
    for j=1:n
        for k=1:n+1-j
            F(k+(i-1)*n)=F(k+(i-1)*n)+(n+2-j-k)*C(State,i+m*(j-1));
        end
    end
end

F = c*F;
```

```
end

function[H,F]=MPC_H_F_for_c_ui_uj_Integ(B,c,State1,State2,n)
% Computes the H and F matrices associated with a cost term, %
Integral(c*u_state1(i)*u_state2(i)), that results from expanding a quadratic objective
% function.  This term is summed from i = 0 to n-1.

[y,m] = size(B);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for k=1:n;
    H(k+(State1-1)*n,k+(State2-1)*n)= H(k+(State1-1)*n,k+(State2-1)*n)+(n+1-k)*1/2;
H(k+(State2-1)*n,k+(State1-1)*n)= H(k+(State2-1)*n,k+(State1-1)*n)+(n+1-k)*1/2;
end

H = c*H;
end

function[H,F]=MPC_H_F_for_c_ui_Integ(B,c,State1,n)
% Computes the H and F matrices associated with a cost term, Integral(c*u_state1(i)),
% that results from expanding a quadratic objective function.  This term is summed
% from i = 0 to n-1.

[y,m] = size(B);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n
    F(i+(State1-1)*n) = n+1-i;
end

F = c*F;
end

function[X]=Newton_Opt_Quad_boundries_AEVPS(H,F,Xi,Low,High,C,UB,P_star,mu,alpha,beta,
eps,max_itr_dual,max_itr_primal,max_itr_line)
%Newtons method based  optimization for a quadratic objective function of the form
% 1/2*x'*H*x+F*x subject to Low <= X <= High and C*X <= UB where C is a matrix and UB
% is a vector (implimented as barrier function)
X = Xi;
m = 2*length(X)+length(UB);
q = m/(0.5*X'*H*X+F*X-P_star);

for k=1:max_itr_dual
    for i=1:max_itr_primal
        grad1_F = q*(H*X+F');
        grad2_F = q*H;
        for j=1:length(X)
            grad1_F(j) = grad1_F(j)-((1/(X(j)-Low(j))+1/(X(j)-High(j))));
            grad2_F(j,j) = grad2_F(j,j)+((1/(X(j)-Low(j))^2+1/(X(j)-High(j))^2));
            for n=1:length(UB)
                grad1_F(j) = grad1_F(j)-C(n,j)/(C(n,:)*X-UB(n));
                for l=1:length(X)
                    grad2_F(j,l) = grad2_F(j,l)+C(n,j)*C(n,l)/(C(n,:)*X-UB(n))^2;
                end
            end
        end
        N_step = -grad2_F\grad1_F;
```

```
        N_dec_2 = -grad1_F'*N_step;
        if N_dec_2/2 <= eps
            break
        end
        t = 1;
        A = q*(0.5*(X+t*N_step)'*H*(X+t*N_step)+F*(X+t*N_step));
        B = q*(0.5*X'*H*X+F*X)+alpha*t*grad1_F'*N_step;
        for j=1:length(X)
            A = A-(real(log(complex(X(j)+t*N_step(j)-Low(j))))+real(log(complex(-
(X(j)+t*N_step(j)-High(j))))));
            B = B-(real(log(complex(X(j)-Low(j))))+real(log(complex(-(X(j)-
High(j))))));
        end
        for n=1:length(UB)
            A = A-real(log(complex(-(C(n,:)*(X+t*N_step)-UB(n)))));
            B = B-real(log(complex(-(C(n,:)*X-UB(n)))));
        end
        for i_line=1:max_itr_line
            if A < B
                break
            end
            t = beta*t;
            A = q*(0.5*(X+t*N_step)'*H*(X+t*N_step)+F*(X+t*N_step));
            B = q*(0.5*X'*H*X+F*X)+alpha*t*grad1_F'*N_step;
            for j=1:length(X)
                A = A-(real(log(complex(X(j)+t*N_step(j)-Low(j))))+real(log(complex(-
(X(j)+t*N_step(j)-High(j))))));
                B = B-(real(log(complex(X(j)-Low(j)))+log(complex(-(X(j)-High(j))))));
            end
            for n=1:length(UB)
                A = A-real(log(complex(-(C(n,:)*(X+t*N_step)-UB(n)))));
                B = B-real(log(complex(-(C(n,:)*X-UB(n)))));
            end
        end
        X = X+t*N_step;
    end
    if m/q < eps
        break
    end
    q = mu*q;
end
end
```

Appendix H Hydraulic Hybrid Logic Code

Below is the MATLAB code used to define the supervisory logic in the hybrid vehicle study. This code is implemented as an embedded MATLAB block.

```
function y  = fcn(Teng,n_m_des,Pu,Pd,Time, T_dwell_mpc)
%#eml

%% Initialization and decleration %%
persistent Flag1 t_trigger % Makes Matlab retain the value of the variables between
                           % function calls.  The variables are still local
if isempty (Flag1)
    Flag1 = 0;
end
if isempty (t_trigger)
    t_trigger = 0;
end
Delta_P = Pu - Pd;
    if Delta_P < 0
        Delta_P = 0;
    end
%% End %%

%% Execution %%
if (Teng > 30 || Pu <= 6.5) && Flag1 == 0
    Flag1 = 1;
    t_trigger = Time;
end

if Flag1 == 1 && Time <= t_trigger+T_dwell_mpc
    C = 1;
else if Flag1 == 1 && n_m_des <= 58*sqrt(Delta_P)/4.216;
        C = 2;
        Flag1 = 0;
        t_trigger = 0;
    else if Flag1 == 1
            C = 1;
        else
            C = 2;
            Flag1 = 0;
            t_trigger = 0;
        end
    end
end
```

```
%% End %%

%% Output %%
y = C;
%% End %%
```

Appendix I Thermal Study EMS Code

Below is the MATLAB code used for the MPC in the hybrid vehicle study. This code uses the 'quadprog' command to solve the optimization problem. This code also allows for a time varying desired cooling capacity and includes cost on the integral of the tracking error. This code is implemented as an embedded MATLAB block.

```
function [u1,u2,u3,u4] =
fcn(Time,Pert,P_low_d,kspeed_d,EEV_d,IV_d,E_rate,S_rate,T_lower,T_upper,
CoolCap_E_old,Diff_temp,Delta_x1,Delta_x2,Delta_x3,Delta_x4,Delta_x5, error_e,
error_s1, error_s2, A_long_on, B_long_on, C_long_on, kspeed_n_on, EEVC_n_on, IVC_n_on,
m_air_s_n_on, n, CoolCap_max, kspeed_max, Lambda1, Lambda2, Lambda3, P_low_max,
A_long_off, B_long_off, C_long_off, Mode, kspeed_n_off, EEVC_n_off, IVC_n_off,
m_air_s_n_off, EEV_max, Lambda4, Beta_on, Ts, Beta_off, CoolCap_P,
Cool_Cap_s_mod_n_off, Cool_Cap_e_n_on)

%#eml

eml.extrinsic('quadprog','optimset')
Time = floor(Time);
CoolCap_D = zeros(1,n);
CoolCap_D(1) = CoolCap_P(Time);
for i = 1:n-1
    CoolCap_D(i+1) = CoolCap_P(Time+i*Ts)+Pert*(rand(1)-0.5)*CoolCap_P(Time+i*Ts);
end

%% Initialization and declaration %%

if Mode <= 0.5

    %OFF Mode%
    QP1 = [(0-kspeed_n_off)*ones(n,1);(0-EEVC_n_off)*ones(n,1);(0-
IVC_n_off)*ones(n,1);(0.0-m_air_s_n_off)*ones(n,1)];
    QP2 = [(2000-kspeed_n_off)*ones(n,1);(15-EEVC_n_off)*ones(n,1);(1-
IVC_n_off)*ones(n,1);(0.09-m_air_s_n_off)*ones(n,1)];

Else

    %ON Mode%
    QP1 = [(0-kspeed_n_on)*ones(n,1);(0-EEVC_n_on)*ones(n,1);(0-
IVC_n_on)*ones(n,1);(0.0-m_air_s_n_on)*ones(n,1)];
    QP2 = [(2000-kspeed_n_on)*ones(n,1);(15-EEVC_n_on)*ones(n,1);(1-
IVC_n_on)*ones(n,1);(0.09-m_air_s_n_on)*ones(n,1)];
```

```
End

[y,m] = size(B_long_on);
Xo=[Delta_x1;Delta_x2;Delta_x3;Delta_x4;Delta_x5];
IC = zeros(m*n,1);
U=zeros(m*n,1);
H=zeros(m*n,m*n);
F=zeros(1,m*n);
%% End %%

%% Execution %%

if Mode <= 0.5

    %OFF Mode%

    CoolCap_d_s = (CoolCap_D./Diff_temp)-Cool_Cap_s_mod_n_off;

[H,F]=H_F_obj_OFF(A_long_off,B_long_off,C_long_off,Xo,error_s2,CoolCap_d_s,kspeed_d,EE
V_d,IV_d,Diff_temp,Lambda1,Lambda2,Lambda3,Lambda4,Beta_off,CoolCap_max,EEV_max,kspeed
_max,Ts,n);
    [Za_sH,Zb_sH] =
T_Super_Heat_const(A_long_off,B_long_off,C_long_off,Xo,T_lower,T_upper,n);
    [Za_C,Zb_C] =
Storage_rate_const(A_long_off,B_long_off,C_long_off,Xo,E_rate,S_rate,n);
    [U] =
quadprog(2*H,F,[Za_C],[Zb_C],[],[],QP1,QP2,IC,optimset('LargeScale','off','MaxIter',20
00));

Else

    %ON Mode%

    CoolCap_d_e = CoolCap_D-Cool_Cap_e_n_on;
    CoolCap_d_s = ((CoolCap_D-CoolCap_E_old)./Diff_temp)-Cool_Cap_s_mod_n_off;
    CoolCap_d = CoolCap_D-Cool_Cap_s_mod_n_off*Diff_temp-Cool_Cap_e_n_on;

[H,F]=H_F_obj_ON(A_long_on,B_long_on,C_long_on,Xo,error_e,error_s1,CoolCap_d,CoolCap_d
_e,CoolCap_d_s,kspeed_d,P_low_d*ones(1,n),m_air_s_n_on,Diff_temp,Lambda1,Lambda2,Lambd
a3,Lambda4,Beta_on,CoolCap_max,P_low_max,kspeed_max,Ts,n);
    [Za_sH,Zb_sH] =
T_Super_Heat_const(A_long_on,B_long_on,C_long_on,Xo,T_lower,T_upper,n);
    [Za_C,Zb_C] =
Storage_rate_const(A_long_on,B_long_on,C_long_on,Xo,E_rate,S_rate,n);
    [U] =
quadprog(2*H,F,[Za_sH;Za_C],[Zb_sH;Zb_C],[],[],QP1,QP2,IC,optimset('LargeScale','off',
'MaxIter',2000));

End

%% End %%

%% Output %%

if Mode <= 0.5

    %OFF Mode%

    u1 =U(1)+kspeed_n_off;
    u2 =U(n+1)+EEVC_n_off;
    u3=U(2*n+1)+IVC_n_off;
```

```
    u4=U(3*n+1)+m_air_s_n_off;
else
    %ON Mode%

    u1 =U(1)+kspeed_n_on;
    u2 =U(n+1)+EEVC_n_on;
    u3=U(2*n+1)+IVC_n_on;
    u4=U(3*n+1)+m_air_s_n_on;

end

%% End %%

end

function[H,F]=H_F_obj_ON(A,B,C,Xo,Zo1,Zo2,CoolCap_d,CoolCap_d_e,CoolCap_d_s,kspeed_d,P
_low_d,m_air_s_n_on,Diff_temp,Lambda1,Lambda2,Lambda3,Lambda4,Beta,CoolCap_max,P_low_m
ax,kspeed_max,Ts,n)

% Computes the H and F matricies for the following objective function:
% J = Lambda1*((CoolCap_e-CoolCap_d_e)/CoolCap_max)^2
%     +Lambda2*((P_low - P_low_d)/P_low_max)^2
%     +Lambda3*((kspeed - kspeed_d)/kspeed_max)^2
% Note all terms are with respect to delta variables, delta_x = x-x_o
%%

[H1,F1]=MPC_H_F_for_c_yi_yj(A,B,C,Xo,1,2,2,n);
[H2,F2]=MPC_H_F_for_c_yi(A,B,C,-2,CoolCap_d_e,2,n);
[H3,F3]=MPC_H_F_for_c_yi_yj(A,B,C,Xo,1,3,3,n);
[H4,F4]=MPC_H_F_for_c_yi(A,B,C,-2,CoolCap_d_s,3,n);
%%
[H6,F6]=MPC_H_F_for_c_yi_yj(A,B,C,Xo,1,1,1,n);
[H7,F7]=MPC_H_F_for_c_yi(A,B,C,-2,P_low_d,1,n);
%%
[H8,F8]=MPC_H_F_for_c_ui_uj(B,1,1,1,n);
[H9,F9]=MPC_H_F_for_c_ui(B,-2*kspeed_d,1,n);
%%
[H10,F10]=MPC_H_F_for_c_ui_uj(B,1,4,4,n);
[H11,F11]=MPC_H_F_for_c_ui(B,-2*(0-m_air_s_n_on),4,n);
%%
[H12,F12]=MPC_H_F_for_Integ_error(A,B,C,Xo,Zo1,Ts,1,CoolCap_d_e,2,n);
[H13,F13]=MPC_H_F_for_Integ_error(A,B,C,Xo,Zo2,Ts,1,CoolCap_d_s,3,n);
%%
H =
Lambda1*(1/CoolCap_max^2)*(H1+H2)+Lambda1*(1/(CoolCap_max/Diff_temp)^2)*(H3+H4)+Lambda
2*(1/P_low_max^2)*(H6+H7)+Lambda3*(1/kspeed_max^2)*(H8+H9)+Beta*H12+10*Beta*H13;
F =
Lambda1*(1/CoolCap_max^2)*(F1+F2)+Lambda1*(1/(CoolCap_max/Diff_temp)^2)*(F3+F4)+Lambda
2*(1/P_low_max^2)*(F6+F7)+Lambda3*(1/kspeed_max^2)*(F8+F9)+Beta*F12+10*Beta*F13;

end

function[H,F]=H_F_obj_OFF(A,B,C,Xo,Zo,CoolCap_d_s,kspeed_d,EEV_d,IV_d,Diff_temp,Lambda
1,Lambda2,Lambda3,Lambda4,Beta,CoolCap_max,EEV_max,kspeed_max,Ts,n)

% Computes the H and F matricies for the following objective function:
% J = Lambda1*((CoolCap_s-CoolCap_d_s)/CoolCap_max)^2
%     +Lambda2*((EEV - EEV_d)/EEV_max)^2
%     +Lambda3*((kspeed - kspeed_d)/kspeed_max)^2
% Note all terms are with respect to delta variables, delta_x = x-x_o
%%
```

```
[H1,F1]=MPC_H_F_for_c_yi_yj(A,B,C,Xo,1,3,3,n);
[H2,F2]=MPC_H_F_for_c_yi(A,B,C,-2,(CoolCap_d_s),3,n);
%%
[H3,F3]=MPC_H_F_for_c_ui_uj(B,1,2,2,n);
[H4,F4]=MPC_H_F_for_c_ui(B,-2*EEV_d,2,n);
%%
[H5,F5]=MPC_H_F_for_c_ui_uj(B,1,1,1,n);
[H6,F6]=MPC_H_F_for_c_ui(B,-2*kspeed_d,1,n);
%%
[H7,F7]=MPC_H_F_for_c_ui_uj(B,1,3,3,n);
[H8,F8]=MPC_H_F_for_c_ui(B,-2*IV_d,3,n);
%%
[H9,F9]=MPC_H_F_for_Integ_error(A,B,C,Xo,Zo,Ts,1,CoolCap_d_s,3,n);
%%
H =
Lambda1*(1/(CoolCap_max/Diff_temp)^2)*(H1+H2)+Lambda2*(1/EEV_max^2)*(H3+H4)+Lambda3*(1
/kspeed_max^2)*(H5+H6)+Lambda4*(1/1^2)*(H7+H8)+10*Beta*H9;
F =
Lambda1*(1/(CoolCap_max/Diff_temp)^2)*(F1+F2)+Lambda2*(1/EEV_max^2)*(F3+F4)+Lambda3*(1
/kspeed_max^2)*(F5+F6)+Lambda4*(1/1^2)*(F7+F8)+10*Beta*F9;
end

function[H,F]=MPC_H_F_for_c_yi_yj(A,B,C,Xo,c,State1,State2,n)

% Computes the H and F matrices associated with a cost term, c*y1(i)*y2(i), that
% results from expanding a quadratic objective function.  This term is
% summed from i = 1 to n.

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end


for j=1:m
    for k=1:n
        for l=1:n+1-k
            q=C*A^(l+k-1);
            for i = 1:r
                F(l+(j-1)*n)=F(l+(j-1)*n)+q(State1,i)*Xo(i)*Z(State2,j+m*(k-
1))+Z(State1,j+m*(k-1))*q(State2,i)*Xo(i);
            end

            for a=1:m
                H(l+(j-1)*n,l+(a-1)*n) = H(l+(j-1)*n,l+(a-
1)*n)+(1/2)*(Z(State1,j+m*(k-1))*Z(State2,a+m*(k-1))+Z(State2,j+m*(k-
1))*Z(State1,a+m*(k-1)));
```

```
                if l >=2
                    for o=1:l-1
                        H(o+(j-1)*n,l+(a-1)*n)= H(o+(j-1)*n,l+(a-
1)*n)+(1/2)*(Z(State1,j+m*(l+k-1-o))*Z(State2,a+m*(k-1))+Z(State2,j+m*(l+k-1-
o))*Z(State1,a+m*(k-1)));
                        H(l+(j-1)*n,o+(a-1)*n)= H(l+(j-1)*n,o+(a-
1)*n)+(1/2)*(Z(State1,j+m*(k-1))*Z(State2,a+m*(l+k-1-o))+Z(State2,j+m*(k-
1))*Z(State1,a+m*(l+k-1-o)));
                    end
                end
            end
        end
    end
end


F = c*F;
H = c*H;
end

function[H,F]=MPC_H_F_for_c_yi(A,B,C,c,R,State,n)

% Computes the H and F matrices associated with a cost term, c*R(i)*y(i), that
% results from expanding a quadratic objective function.  This term is
% summed from i = 1 to n.

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for i=1:m
    for j=1:n
        for k=1:n+1-j
            F(k+(i-1)*n)=F(k+(i-1)*n)+R(k+j-1)*Z(State,i+m*(j-1));
        end
    end
end

F = c*F;
end

function[H,F]=MPC_H_F_for_c_ui_uj(B,c,State1,State2,n)

% Computes the H and F matrices associated with a cost term,
% c*u_state1(i)*u_state2(i), that results from expanding a quadratic objective
% function.  This term is summed from i = 0 to n-1.

[y,m] = size(B);
```

```
F=zeros(1,m*n);
H=zeros(m*n,m*n);

for k=1:n;
    H(k+(State1-1)*n,k+(State2-1)*n)= H(k+(State1-1)*n,k+(State2-1)*n)+1/2;
    H(k+(State2-1)*n,k+(State1-1)*n)= H(k+(State2-1)*n,k+(State1-1)*n)+1/2;
end

H = c*H;
end

function[H,F]=MPC_H_F_for_c_ui(B,c,State1,n)

% Computes the H and F matrices associated with a cost term, c*u_state1(i), that
% results from expanding a quadratic objective function.  This term is
% summed from i = 0 to n-1.

[y,m] = size(B);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n
    F(i+(State1-1)*n) = 1;
end

F = c*F;
end

function [H,F]=MPC_H_F_for_Integ_error(A,B,C,Xo,Zo,Ts,c,R,State,n)

% Computes the H and F matrices associated with a cost term,
% Integral(y_state(i)-R), with a time step of Ts.  This term is
% summed from i = 1 to n.
[H1,F1] = MPC_H_F_for_c_xi_SumSum(A,B,C,-2*Ts*c*Zo,State,n);
[H2,F2]=MPC_H_F_for_c_Sum_Sum_yi_Sum_R(A,B,C,-2*Ts^2*c,State,R,n);
[H3,F3]=MPC_H_F_for_c_xi_xj_Sum_square_Sum(A,B,C,Xo,Ts^2*c^2,State,n);

H = H1+H2+H3;
F = F1+F2+F3;
end

function[H,F]=MPC_H_F_for_c_xi_SumSum(A,B,C,c,State,n)

% Computes the H and F matrices associated with a cost term, C*Sum(Sum(y_state(i))).
% This term is summed from i = 1 to n.

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
```

```
        end
    end
end

for i=1:m
    for j=1:n
        for k=1:n+1-j
            F(k+(i-1)*n)=F(k+(i-1)*n)+(n+2-j-k)*Z(State,i+m*(j-1)); % Added (n+2-j-k)*
        end
    end
end

F = c*F;
end

function[H,F]=MPC_H_F_for_c_Sum_Sum_yi_Sum_R(A,B,C,c,State1,R,n)

% Computes the H and F matrices associated with a cost term,
% C*Sum(Sum(y_state1(i))*Sum(R(i))).  This term is summed from i = 1 to n.

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);
Q = zeros(r,m);
F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    Q = Q+C*(A^(i-1))*B;
    for j=1:r;
        for k=1:m;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    q = 0;
    for k=1:n
        for i = 1:k-1
            q = q+R(i);
        end
        for l=1:n+1-k
            %q = sum(R(1:l+k-1));
            q = q + R(l+k-1);
            F(l+(j-1)*n)=F(l+(j-1)*n)+Z(State1,j+m*(k-1))*q;
        end
        q = 0;
    end
end

F = c*F;
end

function[H,F]=MPC_H_F_for_c_xi_xj_Sum_square_Sum(A,B,C,Xo,c,State,n)

% Computes the H and F matrices associated with a cost term,
% C*Sum((Sum(x_state(i)))^2).  This term is summed from i = 1 to n.

[y,m] = size(B);
```

```
[r,p] = size(C);

Z = zeros(r,m*n);
Q = zeros(r,m);
F=zeros(1,m*n);
H=zeros(m*n,m*n);

for i=1:n;
    Q = Q+C*(A^(i-1))*B;
    for j=1:r;
        for k=1:m;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end


for j=1:m
    q = zeros(r,p);
    for k=1:n
        q = q+C*A^(k-1);
        for l=1:n+1-k
            q = q+C*A^(l+k-1);
            for i = 1:p
                F(l+(j-1)*n)=F(l+(j-1)*n)+2*q(State,i)*Xo(i)*Z(State,j+m*(k-1));
            end

            for a=1:m
                H(l+(j-1)*n,l+(a-1)*n) = H(l+(j-1)*n,l+(a-1)*n)+Z(State,j+m*(k-
1))*Z(State,a+m*(k-1));

                if l >=2
                    for o=1:l-1
                        H(o+(j-1)*n,l+(a-1)*n)= H(o+(j-1)*n,l+(a-
1)*n)+Z(State,j+m*(l+k-1-o))*Z(State,a+m*(k-1));
                        H(l+(j-1)*n,o+(a-1)*n)= H(l+(j-1)*n,o+(a-1)*n)+Z(State,j+m*(k-
1))*Z(State,a+m*(l+k-1-o));
                    end
                end
            end
        end
        q = zeros(r,p);
        for b = 1:k
            q = q+C*A^(b-1);
        end
    end
end


F = c*F;
H = c*H;
end

function[Za,Zb]=MPC_Constraint_for_yi_equal_to_c(A,B,C,Xo,c,State,n)

% Computes the Za and Zb matrices associated with a constraint term, x_state(i)=c.
% This term is summed from i = 1 to n

[y,m] = size(B);
[r,p] = size(C);
```

```
Z = zeros(r,m*n);

Za = zeros(n,m*n);
Zb = c*ones(n,1);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    for k=1:n
        for l=1:n+1-k
            Za(l+k-1,l+(j-1)*n) = Z(State,j+(k-1)*m);
        end
    end
end

for i=1:n
    for j=1:r
        q=C*A^(i);
        Zb(i) = Zb(i)-q(State,j)*Xo(j);
    end
end
end

function[Za,Zb]=MPC_Constraint_for_yi_plus_yi_equal_to_c(A,B,C,Xo,c,c1,c2,State1,State
2,n)

% Computes the Za and Zb matrices associated with a constraint term,
% c1*x_state1(i)+c2*x_state2(i)=c.  This term is summed from i = 1 to n

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

Za = zeros(n,m*n);
Zb = c*ones(n,1);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    for k=1:n
        for l=1:n+1-k
            Za(l+k-1,l+(j-1)*n) = c1*Z(State1,j+(k-1)*m)+c2*Z(State2,j+(k-1)*m);
        end
    end
end
```

```
for i=1:n
    for j=1:r
        q=C*A^(i);
        Zb(i) = Zb(i)-c1*q(State1,j)*Xo(j)-c2*q(State2,j)*Xo(j);
    end
end
end

function[Za,Zb]=MPC_Constraint_for_yi_less_than_c(A,B,C,Xo,c,State,n)

% Computes the Za and Zb matrices associated with a constraint term, x_state(i)<=c.
% This term is summed from i = 1 to n

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

Za = zeros(n,m*n);
Zb = c*ones(n,1);

for i=1:n;
    for j=1:r;
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    for k=1:n
        for l=1:n+1-k
            Za(l+k-1,l+(j-1)*n) = Z(State,j+(k-1)*m);
        end
    end
end

for i=1:n
    for j=1:r
        q=C*A^(i);
        Zb(i) = Zb(i)-q(State,j)*Xo(j);
    end
end
end

function[Za,Zb]=MPC_Constraint_for_yi_greater_than_c(A,B,C,Xo,c,State,n)

% Computes theZa and Zb matrices associated with a constraint term, x_state(i)>c.
% This term is summed from i = 1 to n

[y,m] = size(B);
[r,p] = size(C);

Z = zeros(r,m*n);

Za = zeros(n,m*n);
Zb = -c*ones(n,1);

for i=1:n;
    for j=1:r;
```

```
        for k=1:m;
            Q = C*(A^(i-1))*B;
            Z(j,k+(i-1)*m) = Q(j,k);
        end
    end
end

for j=1:m
    for k=1:n
        for l=1:n+1-k
            Za(l+k-1,l+(j-1)*n) = -Z(State,j+(k-1)*m);
        end
    end
end

for i=1:n
    for j=1:r
        q=C*A^(i);
        Zb(i) = Zb(i)+q(State,j)*Xo(j);
    end
end
end

function[Za_comb,Zb_comb] = Storage_rate_const(A,B,C,Xo,E_rate,S_rate,n)
[Za1,Zb1]=MPC_Constraint_for_yi_less_than_c(A,B,C,Xo,E_rate,2,n);
Za_comb = Za1;
Zb_comb = Zb1;
end

function[Za_comb,Zb_comb] = T_Super_Heat_const(A,B,C,Xo,T_lower,T_upper,n)
[Za1,Zb1]=MPC_Constraint_for_yi_greater_than_c(A,B,C,Xo,T_lower,4,n);
[Za2,Zb2]=MPC_Constraint_for_yi_less_than_c(A,B,C,Xo,T_upper,4,n);
Za_comb = [Za1;Za2];
Zb_comb = [Zb1;Zb2];
end
```

Appendix J Thermal Study Logic Code

Below is the MATLAB code used to define the supervisory logic in the thermal hybrid study. This code is implemented as an embedded MATLAB block.

```
function m = fcn(T,M)
%#eml
persistent F
if isempty(F)
    F = 0;
end

if T>=25 || M>=0.9*0.09
    F = 1;
else if T < 0.0
        F = 0;
    end
end

if F == 0
    m = 0;
else
    m = 1;
end
```