8-2010

Ultracapacitor Heavy Hybrid Vehicle: Model Predictive Control Using Future Information to Improve Fuel Consumption

Seneca Schepmann
Clemson University, sschepm@clemson.edu

Ultracapacitor Heavy Hybrid Vehicle: Model Predictive Control Using Future Information to Improve Fuel Consumption

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Seneca Andrew Schepmann
August 2010

Accepted by:
Dr. Ardalan Vahidi, Committee Chair
Dr. John Wagner
Dr. Mohammed Daqaq

Abstract

This research is concerned with the improvement in the fuel economy of heavy transport vehicles through the use of high power ultracapacitors in a mild hybrid electric vehicle platform. Previous work has shown the potential for up to 15% improvement on a hybrid SUV platform, but preliminary simulations have shown the potential improvement for larger vehicles is much higher.

Based on vehicle modeling information from the high fidelity, forward-looking modeling and simulation program Powertrain Systems Analysis Toolkit (PSAT), a mild parallel heavy ultracapacitor hybrid electric vehicle model is developed and validated to known vehicle performance measures. The vehicle is hybridized using a 75kW motor and small energy storage ultracapacitor pack of 56 Farads at 145 Volts. Among all hybridizing energy storage technologies, ultracapacitors pack extraordinary power capability, cycle lifetime, and ruggedness and as such are well suited to reducing the large power transients of a heavy vehicle.

The control challenge is to effectively manage the very small energy buffer (a few hundred Watt-hours) the ultracapacitors provide to maximize the potential fuel economy. The optimal control technique of Dynamic Programming is first used on the vehicle model to obtain the ”best possible” fuel economy for the vehicle over the driving cycles. A variety of energy storage parameters are investigated to aid in determining the best ultracapacitor system characteristics and the resulting effects this has on the fuel economy.

On a real vehicle, the Dynamic Programming method is not very useful since it is computationally demanding and requires predetermined vehicle torque de-

mands to carry out the optimization. The Model Predictive Control (MPC) method is an optimization-based receding horizon control strategy which has shown potential as a powertrain control strategy in hybrid vehicles. An MPC strategy is developed for the hybrid vehicle based on an exponential decay torque prediction method which can achieve near-optimal fuel consumption even for very short prediction horizon lengths of a few seconds. A critical part of the MPC method which can greatly affect the overall control performance is that of the prediction model. The use of telematic based "future information" to aid in the MPC prediction method is also investigated. Three types of future information currently obtainable from vehicle telematic technologies are speed limits, traffic conditions, and traffic signals, all of which have been incorporated to improve the vehicle fuel economy.

Acknowledgments

To all of those who contributed to this research in some way, I thank you. It would not have been possible to complete this work without the help of friends and classmates; Dean, Ali, Chen, and Grant - you have all contributed to my successes in graduate school here at Clemson, and I don't know if I could have accomplished all of this without you.

I owe much to the professors and staff of the department, especially my committee advisors, who have helped advance my continual studies since I began here six years ago. The lessons learned both inside and outside the classroom, be it through help with homework, research, or simple advice on life, will stick with me for a long time to come.

Above all, I have to thank my advisor and friend Dr. Ardalan Vahidi for his continual efforts keeping me directed, motivated, and productive on my research tasks, both graduate and undergraduate. Dr. Vahidi has helped me in more ways than I can count, and without his kindness and dedication to me as a student this whole adventure would have never left the ground.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The improvement of vehicle efficiency, in particular the vehicle fuel economy, is a strong driving force for the development of new vehicle platforms, components, configurations, and related technologies. Many different technological areas are under research for reducing vehicle fuel consumption, including lightweight vehicle design, advanced engine technologies, shifting optimization methods, and powertrain hybridization.

A primary research and development focus of the U.S. Army is on the fuel efficiency of its vehicles. There exists a great need for improved vehicle efficiency due to the ever increasing energy and power demands that new technologies continually bring to the battlefield. The need for a substantial and continual supply of fuel for battle operations is not only a significant cost but also a strategic disadvantage. Military transport vehicles similar to the LMTV M1081 serve a variety of military needs, consume substantial amounts of fuel, and are a heavy vehicle platform suitable for hybridization as the target vehicle for this research. This research is concerned with the improvement in the fuel economy of heavy vehicles through the use of high power ultracapacitors in a mild hybrid electric vehicle platform. Previous work has shown the potential for up to 15% improvement on the smaller hybrid SUV platform, but preliminary simulations have shown the potential improvement for larger vehicles is much higher.

Initial work focuses on modeling the power and energy demands of the M1081 heavy vehicle platform and the selection of a suitable driving cycle. The power demands of the heavy vehicle are substantially higher than those seen in smaller vehicles and the high power capability of ultracapacitors is well suited to assisting the vehicle in meeting these high power demands. A broad overview of hybridizing technologies is undertaken, comparing currently available products in a variety of technological areas on their power and energy characteristics. This overview serves as a basis for the choice of ultracapacitors as the hybridizing energy storage technology. Ultracapacitors pack extraordinary power capability, cycle lifetime, and ruggedness, and as such are well suited to reducing the large power transients of a heavy military vehicle.

Based on vehicle modeling techniques and data from the MATLAB® based, high fidelity, forward-looking modeling and simulation program Powertrain Systems Analysis Toolkit (PSAT), a mild parallel heavy ultracapacitor hybrid electric vehicle model is developed and validated to known vehicle performance measures. The hybrid vehicle uses a 75kW motor and ultracapacitor pack of 56 Farads at 145 Volts. Simulations show a large potential improvement in fuel economy with the effective use of this relatively small amount of energy storage.

The control challenge explored here is to effectively manage the very small energy buffer (a few hundred Watt-hours) of ultracapacitors in order to maximize the potential fuel economy. First, the optimal control technique of Dynamic Programming (DP) is used with the vehicle model to obtain the "best possible" fuel economy for the vehicle over the driving cycles. Since the DP method provides a consistently implementable control method and comparable results, a variety of vehicle component variations are carried out at this time. Changes studied here include the influence of the capacitance, resistance, and voltage of the ultracapacitor pack on the resulting maximum fuel economy, as well as other factors like the vehicle mass.

On a real vehicle, the Dynamic Programming technique is not very useful since it requires predetermined vehicle torque demands to carry out the optimization calculations. The Model Predictive Control (MPC) method is an optimization-based receding-horizon control strategy which has shown potential as a power-

train control strategy in hybrid vehicles. An MPC strategy is developed for the heavy hybrid vehicle based on the same vehicle model and DP cost function using an exponential decay prediction method which can achieve near-optimal fuel consumption even for very short horizon lengths. The prediction model of MPC is critical to the overall control performance. A new expected vehicle demand-based method of tuning the exponential decay method is presented along with improved results over previous methods.

Modern vehicles are continually integrating new technologies like GPS navigation systems for the benefit and convenience of the driver. These telematic technologies have the potential to be used in aiding the vehicle control strategy to improve the fuel economy. In this research, realistic "future information" is used to aid in the MPC prediction method and improve the potential fuel economy of the vehicle. Examples of future information used here are speed limits, traffic conditions, and traffic signals along the desired route. The future information is able to improve the fuel economy above and beyond the promising results shown using the exponential decay prediction method.

The thesis organization is as follows: Chapter 2 provides the background information on hybrid vehicles, the vehicle driving demands, and the comparison of hybridizing energy storage technologies currently available in the market. Chapter 3 details the modeling development, validation, and vehicle simulations conducted in the PSAT software environment, as well as covering previous research done on ultracapacitor hybrid vehicles. Chapter 4 goes through the Dynamic Programming model, its validation and comparison to PSAT results, and the simulation results from the investigations in energy storage system sizing. Chapter 5 transitions into the forward looking Model Predictive Control structure and investigates the exponential decay prediction method. Chapter 6 incorporates the use of future information into the MPC prediction method to improve the fuel economy even further. The final chapter offers conclusions and possible future areas of exploration which may hold promise for further research.

Chapter 2

Hybrid Vehicle Background

In this chapter, general background material is presented which covers a variety of topics surrounding hybrid vehicle power systems. Current hybrid system designs and hybridizing technologies are covered first, with brief comparisons and discussion of the current technological limitations. This is followed by a simple modeling approach to determining the expected power and energy demands of a driving vehicle. The use of standard driving cycles and basic physics concepts allows for interesting insights into the power demands and associated sizing of hybrid vehicle components. Since this research is concerned primarily with energy storage and usage, a comparison of available energy storage technologies is covered as well, with an emphasis on batteries, ultracapacitors, and hydraulic hybrid systems. More information on these topics can be learned from sources in the complete list of references. Lastly, this chapter will cover previous research conducted by Clemson students in the area of model-predictive control methods (MPC) applied to hybrid vehicle control.

2.1 Components, Configurations, Driving Demands

By definition, a hybrid vehicle is any vehicle which makes use of more than one energy source for propulsion. This definition includes a variety of types of vehicles, many of which are currently based on the internal combustion engine (ICE) along with an auxiliary propulsion source. Many different purposes exist for combining more than one propulsion system into a single vehicle, including: improvement of drive train efficiency (fuel economy), performance improvement (acceleration), environmental concerns (reduction of emissions), reductions in maintenance (longer service intervals), as well as reduced cost of ownership. A variety of hybrid systems have been commercially developed to suit these various purposes, competing for a market share that has been continuously growing over the past decades. Within a hybrid vehicle platform, the multiple propulsion sources can be arranged in a variety of mechanical configurations, such as in "series", "parallel", or "power-split" hybrids. More details on the components required in a hybrid vehicle and the possible powertrain configurations are presented in the following sections.

2.1.1 Vehicle Components

In addition to the components required for a conventional, ICE-driven vehicle, a hybrid vehicle generally requires a few extra components to be integrated into the powertrain. Hybrids require an additional energy storage system, a conversion device (to provide mechanical energy to drive the wheels from this stored energy), as well as the control systems and electronics needed to integrate these systems with the rest of the vehicle.

Energy storage is usually accomplished solely by the fuel tank, but in a hybrid vehicle an additional reservoir of energy is needed. This energy storage system (ESS) generally contains electric energy in the form of batteries or ultracapacitors, mechanical energy in the form of compressed hydraulic fluids or kinetic energy of a flywheel, or chemical energy from a hydrogen fuel tank. In the same

manner as ignition timing control provides effective use of fuel in the ICE, some form of control system is required in order to control the usage of this secondary energy source. For electric hybrids, this can be any sort of motor control or power electronic device which commands the motor. In hydraulic hybrids, the control system consists of flow switches and hydraulic pump/motor controls.

While most current hybrid vehicles rely mostly on the ICE, the secondary propulsion source is usually designed for a specific benefit. This propulsion source, while generally smaller than the ICE, helps in providing peak load assistance, low speed torque, or in allowing for the capture of energy normally lost as heat in braking. This latter task, generally referred to as "regenerative braking", improves the hybrid vehicle's energy usage by capturing kinetic energy which is normally wasted in friction brakes and allowing it to be subsequently used for propulsion. Even high-profile racing circuits such as Formula One have been pushing towards hybrid vehicles with this capability; in fact, it is a new rule for the 2009 season that each Formula One car may employ some form of kinetic energy recovery system (KERS) [11]. This KERS system is designed to provide each vehicle with an added boost on every lap of up to 60kW using a maximum of 400kJ of energy [11]. Aside from this use of hybridization for increases in acceleration performance, the most common use of hybrid technologies in passenger vehicles is to provide increased fuel economy and lower emissions. Towards this aim, a variety of different hybrid configurations as well as hybridizing technologies have been developed over the past decades.

2.1.2 Vehicle Configurations

Within a hybrid vehicle platform, the multiple propulsion sources can be arranged in a wide variety of mechanical configurations. The three most common of these configurations are known as "parallel", "series" or "power-split" hybrids. Figures depicting these configurations show the three separate regions of power usage present in any vehicle: Energy Storage, Energy Transformation (from the stored energy to mechanical energy) and Energy Transmission to the wheels. Each of these regions has separate requirements and issues associated

with the implementation of a given hybrid technology. In general, a hybrid vehicle has more than one energy storage system (ESS) and more than one driving mode. Most hybrids currently available use an internal combustion engine with a battery pack and electric motor, so for simplicity, these terms are used in the diagrams. The later Section 2.2 discusses the different hybridizing technologies which are becoming more and more prevalent.

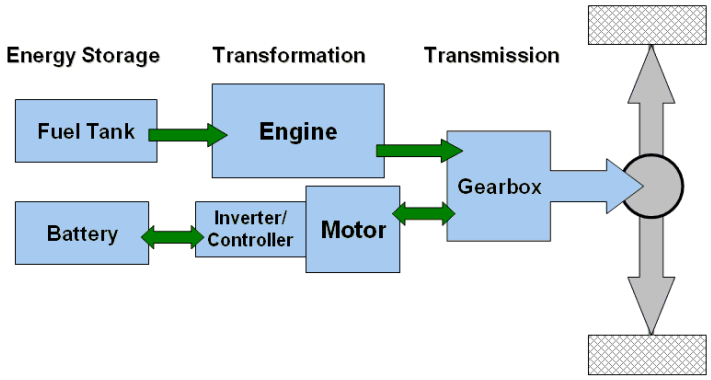


Figure 2.1: Schematic of Parallel Hybrid Vehicle

The parallel hybrid configuration is shown above in Figure 2.1. A parallel hybrid vehicle is so termed because of the two separate and parallel power sources which work together to propel the vehicle. Mechanical clutches can allow the engine or motor to separately power the vehicle and the motor placement can vary; it can come before or after the gearbox or even be directly coupled to the wheels. According to [19] parallel hybrids are usually lower power vehicles, such as passenger cars, where the hybridizing system is used to enhance performance. In contrast, a series hybrid is shown below in Figure 2.2.

A series hybrid makes use of the electric motor (or other device, such as a hydraulic pump/motor) to provide all of the vehicle propulsion demand. In this configuration, the IC engine provides only additional electric power or energy to increase acceleration or extend the range of the vehicle beyond that of the electric battery alone. [19] claims that series hybrids are generally higher power systems, sometimes even using a gas turbine between 150kW and 1000kW. A significant benefit of the series hybrid configuration is the added degree of freedom in be-

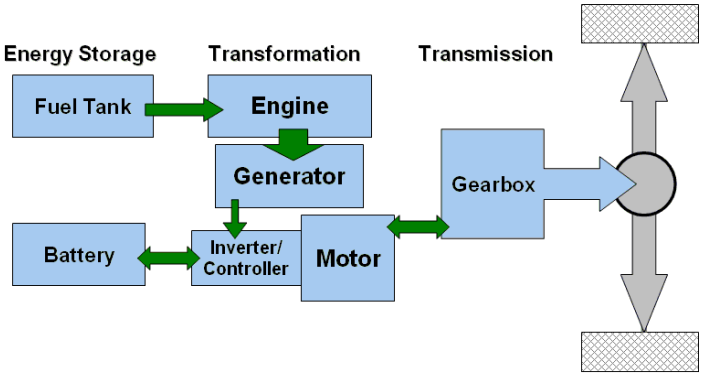


Figure 2.2: Schematic of Series Hybrid Vehicle

ing able to control the engine speed independent of the vehicle speed. With the appropriate controls methodologies, this hybrid configuration should be able to increase mileage by 50% and decrease emissions by 60% or more [19].

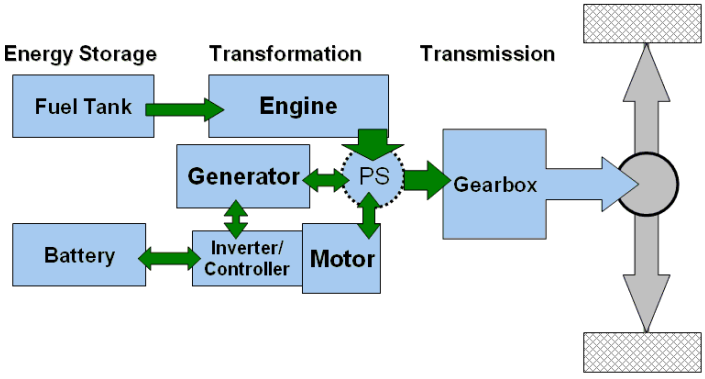


Figure 2.3: Schematic of Powersplit Hybrid Vehicle

A "power-split" hybrid configuration is shown in 2.3. This type of hybrid makes use of a power splitting device, usually a planetary gear mechanism, and has become a popular design due to its unique modes of operation as well as its implementation in the successful Toyota Prius. Toyota's hybrid system is able to function as a continuously variable transmission and provide smooth power delivery and very efficient operation [19]. One of the biggest benefits of this type of

hybrid configuration is the great potential reduction in vehicle emissions [19].

A variety of technologies exist which can be employed as the second power source for a hybrid vehicle. The main types of hybrid vehicles available in the market are those employing ICE's alongside batteries, ultracapacitors, or hydraulic hybrid systems. These technologies are of primary research interest as fossil fuel-saving technologies which are beginning the transition away from ICE-based vehicles into the future cars based in fuel cells or battery electric. But before an energy storage system can effectively be designed, it is critically important to understand the necessary power and energy requirements for the vehicle's mission.

2.1.3 Energy and Power Requirements

Every vehicle is designed with a specific set of requirements in mind, which can range from low speed neighborhood vehicles, average commuter cars, all the way up to the fastest "supercars" capable of speeds faster than most roads can accommodate. In order to understand these requirements, original equipment manufacturers (OEMs) set design goals for vehicle performance in a variety of categories. In hybrid vehicles, improvement of fuel economy is an important goal. To save time and money in building new prototype vehicles, simulation tools are widely used to help design and predict the fuel economy of these vehicles. A critical element needed to effectively simulate a vehicle for objective comparisons is the driving cycle, which defines a specific velocity versus time curve for the vehicle to follow [35]. The selection of this driving cycle depends on the anticipated vehicle use and is specific to the vehicle's purpose; the power and energy requirements of different driving cycles can vary widely for the same vehicle. In order to help define comparative baselines for vehicle consumption, a simple MATLAB® routine has been created which calculates a variety of power and energy requirements for a few standard driving cycles. Using the velocity versus time input from the driving cycle and basic information about the vehicle the program calculates torques and powers required to overcome rolling resistance, aerodynamic drag, inertia, and road grade. From these calculations, characteristic energy and power requirements are tabulated to help describe the vehicle

demands resulting from driving on this cycle. The following formulae are used to calculate these demands using a method similar to [23]:

$$F = \alpha_A v^2 + \alpha_I a + \alpha_R + \alpha_S \tag{2.1}$$

where F is the total driving force, v is the velocity, a is the acceleration, α_A is a multiplying coefficient for aerodynamic forces, α_I is the coefficient for inertial forces, α_R is the coefficient for rolling resistance, and α_S is the coefficient for road grade or slope forces. The α terms of Equation 2.1 are defined as follows:

$$\begin{aligned} \alpha_A &= 0.5\rho C_D A_f & \alpha_I &= 1.1m \\ \alpha_R &= (C_{R,1} + C_{R,2}\omega)\, mg\cos\theta & \alpha_S &= mg\sin\theta \end{aligned} \tag{2.2}$$

where ρ is the air density, C_D is the vehicle drag coefficient, A_f is the vehicle frontal area, m is the vehicle mass, $C_{R,1}$ and $C_{R,2}$ are rolling resistance coefficients, ω is the wheel rotational speed, g is the acceleration of gravity, and θ is the road slope angle. The definition of rolling resistance here uses a first order relationship in ω which is further detailed in [23]. Using defined vehicle parameters and knowledge of the driving cycle, these coefficients can be defined and used to calculate the torque and power requirements at the wheel as:

$$\begin{aligned} T_{dmd,wheel} &= Fr \\ P_{dmd,wheel} &= T_{dmd,wheel}\omega \end{aligned} \tag{2.3}$$

where T is total torque demand, P is total power demand and r is the wheel radius. Here a condition is enforced such that the torque and power demands are zero when the vehicle speed and acceleration are both zero, since the vehicle is stopped. This condition would have to be changed slightly for any driving cycle that has road slope changes, but at this time, the road is considered flat with zero slope. The same modeling relationships are employed throughout subsequent work, including the simulation studies and the controls investigations.

2.1.3.1 Sport Utility Vehicle Platform

The first vehicle considered on these cycles is that of an average sport utility vehicle (SUV) with the general specifications listed in Table 2.1.

Make	Model	Year	Engine Size	Power kW [hp]	Curb Weight kg [lbs]	Fuel Economy City / Highway
Ford	Explorer	2009	4.0L 6-Cylinder	157 [210]	2018 [4450]	13 / 19 mpg

Table 2.1: Baseline Passenger SUV Specifications

A wide range of driving scenarios will likely be encountered by a typical SUV which range from slow urban traffic to high-speed highway driving. This variety makes it difficult to capture the driving requirements in one simple cycle, so a variety of cycles were considered: New York City Traffic (low speed stop and go), Artemis Urban (moderate speed stop and go), UDDS (Urban Dynamometer Driving Schedule - moderate speed, more stop and go), EPA LA92 (low to high speed, more aggressive), Artemis Extra Urban (moderate and high speed), and Artemis Highway (highway driving at high speeds). For more details on these cycles, see the Appendix. The chosen cycles are all passenger vehicle cycles which have been developed by various agencies for the purposes of vehicle fuel consumption estimation. The detailed results of this MATLAB® routine run over the given cycles are presented in Table 2.2

Maximum Demanded Power[kW] or Energy[kJ]	NY City	— Urban	Artemis Cycles Extra Urban	— Highway	UDDS	EPA LA92
Propulsion Power	54.5	54.9	94.0	147	68.9	93.6
Regen Power	19.9	43.9	64.7	94.6	22.1	89.3
Avg Power (5 sec)	36.3	43.3	83.1	136	58.3	84.6
Avg Power (20 sec)	16.5	23.9	68.9	96.5	41.0	71.4
Avg Power (60 sec)	8.9	18.3	44.1	78.7	33.7	48.5
Energy (5 sec)	182	216	415	679	291	423
Energy (20 sec)	330	478	1378	1929	819	1427
Energy (60 sec)	534	1100	2644	4723	2019	2910

Table 2.2: Power and Energy Characteristics of Passenger Driving Cycles - Typical SUV

The table shown provides a large number of comparative numbers for these cycles which can be very helpful in designing a hybrid vehicle. When considering these cycles, the maximum power capabilities of the sport utility vehicle are sufficient to meet the highest demands calculated. However, this conclusion is only valid provided the engine is geared correctly to deliver enough power or torque at the needed road speed. Gearing considerations are generally carried out to give the vehicle adequate acceleration from stop as well as good fuel economy when driving at freeway speeds. In the hybrid vehicle, it is important that the regenerative braking system is sized to adequately capture the potentially wasted braking energy from driving. As is evident from the "Regen Power" row of the table, the power demands from braking can approach (and in some scenarios even surpass) that of the maximum propulsion power. While all of this power will go wasted in a typical vehicle, it is important to the fuel economy benefits of hybrids to consider the potential regenerative energy savings when sizing the vehicle propulsion systems.

In an ultracapacitor hybrid vehicle, where the electric hybrid system is designed to reduce peak torque demands in order to reduce fuel consumption, the amount of energy storage available is the limiting factor. An important consideration which is explored here is that of the energy demands of the cycles. Table 2.2 contains rows of values for the maximum energy required for any given 5, 20, or 60 second interval over the cycle. This energy quantity is measured in kJ and is included for future discussions relevant to the energy storage capabilities of ultracapacitors. In later modeling, the ultracapacitor module considered (165F at 48V) contains approximately 200kJ of energy. A minimum of three of these modules are used in simulations due to the voltage requirements of the motor/inverter, for a total energy storage around 600kJ. The effective use of this tiny energy storage should be able to improve fuel economy, however, the challenge comes with the implementation of an effective energy management control.

2.1.3.2 Military Transport Vehicle Platform

This research is primarily concerned with the use of ultracapacitors in larger vehicles, with the targeted vehicle being a 2.5-ton light military transport vehicle (LMTV), in particular the M1081 standard cargo truck. This vehicle's purpose is to transport goods and troops at relatively low speeds of up to around 50mph over a variety of terrains and conditions. The specifications for the M1081 are listed in Table 2.3 and were obtained from [13] and [2]. The fuel economy figure is approximate and assumes the vehicle is at its gross weight of approximately 9700kg, traveling at an average velocity of 25mph.

Model	Engine Size	Power kw [hp]	Curb Weight kg [lbs]	Fuel Economy
LMTV M1081	6.6L 6-Cyl Diesel	168 [225]	7900 [17500]	5.55 mpg

Table 2.3: Baseline Heavy Vehicle Specifications

The mission of this vehicle is substantially different from that of the typical passenger car, and as such the previously mentioned driving cycles are not well-suited to this vehicle. The cycles considered for this vehicle are those more appropriate for refuse trucks and transit buses which have lower speed with more stop and go driving. The cycles considered here are: Manhattan bus (low speed, stop and go), Artemis urban (moderate speed stop and go), CSHVR (City Suburban Heavy Vehicle Route - moderate speed stop and go), UDDS(moderate speed, more stop and go), and UDDS Truck (moderate speed, less stop and go). Again, more details of these cycles can be found in the Appendix. The results from running the same MATLAB® routine with the fully loaded M1081 vehicle over these cycles is presented in Table 2.4

The table values were generated for the worst case loading scenario, when the vehicle is fully loaded with cargo and has a gross vehicle weight of 9700kg. It is evident from the table values that the maximum power demands for the M1081 on these cycles are in excess of the vehicle's capability. Even the comparatively gentle low speed cycles that are more representative of the normal demands on a larger vehicle are potentially too much for the engine to meet. It is not expected in prac-

Maximum Demanded Power[kW] or Energy[kJ]	Manhattan Bus	Artemis-Urban	CSHVR	UDDS	UDDS Truck
Propulsion Power	198	275	204	383	353
Regen Power	79.7	205	62.2	125	99.5
Avg Power (5 sec)	154	217	165	220	297
Avg Power (20 sec)	59.8	130	143	167	215
Avg Power (60 sec)	33.2	99.4	115	154	182
Energy (5 sec)	773	1088	825	1100	1487
Energy (20 sec)	1196	2604	2875	3340	4300
Energy (60 sec)	1992	5963	6910	9288	11000

Table 2.4: Power and Energy Characteristics of Heavy Vehicle Driving Cycles - M1081

tice that a given vehicle will be able to perfectly follow the power demands placed on it in these cycles, however, the cycles are still used as guidelines for determining vehicle capabilities. The current generation of military transport vehicles have a slightly smaller engine than the original 205kW engine designed for the vehicle. This change was due to the need to meet EPA guidelines for emissions in the late 1990's and the vehicle is only capable of speeds up to around 55mph [13]. Newer generation vehicles will likely seek to increase the vehicle's power capability while at the same time desiring greater fuel efficiency. The UDDS and UDDS truck driving cycles are included in this comparison for this reason, since they are slightly more aggressive than the vehicle currently can handle.

When comparing the tabulated values for maximum power on each of these cycles, the M1081 is only capable of meeting the five second max power demands of two of the cycles. The regenerative power demands are quite significant as well in all the cycles and should be considered in the sizing of the hybrid system in order to allow for the effective capture of this regenerative energy. As noted previously, if the ultracapacitor hybrid system is desired to reduce fuel economy by providing a power boost, the limiting factor is the total energy storage. The base module considered in simulations contains approximately 600kJ of energy, a figure which is nearly sufficient to completely propel the vehicle during the five seconds of highest energy demand on the Manhattan Bus and CSHVR cycles. The vehicle design question to be answered is how much energy do we really need to provide a significant benefit to the fuel economy of the vehicle.

2.1.3.3 Characteristic Driving Cycle Selection

In choosing a simulation cycle that is most representative of this vehicle's mission, the cycle speed profile as well as the characteristic power demands have been considered. The cycle used extensively for simulations in the rest of the report will be that of the City Suburban Heavy Vehicle Route, or CSHVR for short. This cycle's top speed is 20m/s (44mph) and has the velocity profile depicted below in Figure 2.4.

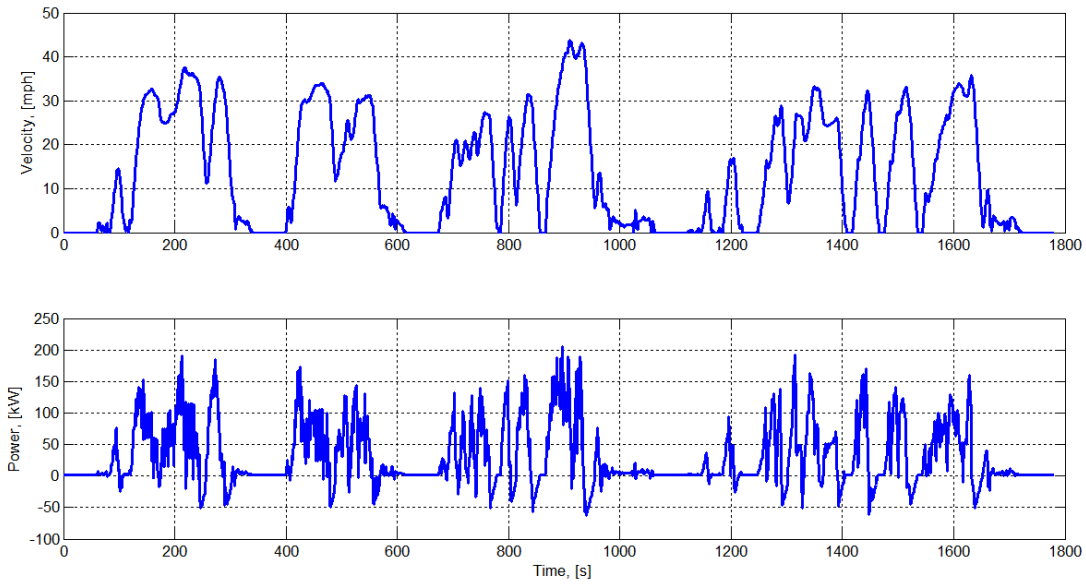


Figure 2.4: Velocity and Power Profile for CSHVR Cycle

In choosing this cycle as a representative cycle for this vehicle, a few different considerations were made. Considering the cycle velocity profile, as can be seen from Figure 2.4, the cycle is relatively low speed with portions of higher speed and numerous stop-and-go cycles. The CSHVR cycle was originally created to aid in the modeling of heavy vehicle emissions on varying suburban transport routes; it was created by a "concatenation of microtrips" which are typical of real truck use [41]. This will allow for the results from this study to be applied to both the M1081 transport vehicle platform as well as to similar sized non-military cargo vehicles.

Additionally, the cycle's power demands can be analyzed in a temporal fashion by plotting the gross power demand versus the amount of cycle time spent above that power level. This power spectrum provides a variety of interesting insights into the demands placed on the vehicle. The power spectrum for each of the five heavy vehicle cycles is shown below in Figure 2.5.

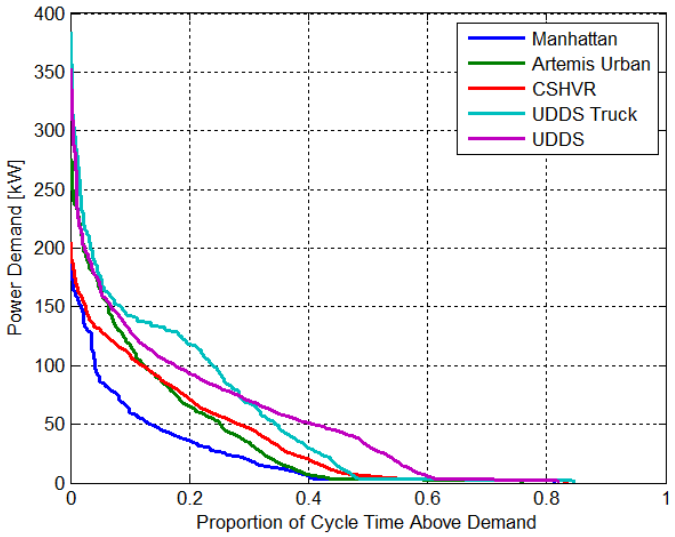


Figure 2.5: Cycle Power Spectra - Actual Power Values

The power spectra are scaled over total cycle time, so the proportion of the cycle time spent above a certain power level is plotted with the corresponding power level. If we want to compare these results across multiple cycles it is easier to show the same plot but scale the power values to the maximum power value for each cycle.

A variety of conclusions can be drawn from the plot when scaled in both time and power level, as is shown in Figure 2.6. First, the amount of time that the vehicle spends at idle is given by the proportion of the cycle spent down near zero power demand. More difficult to see from the figure is the amount of time spent in regeneration which is the fraction without a line. Secondly, the slope of the line in the middle section corresponds to the "difficulty" of the cycle, namely the amount of time the vehicle is demanding higher power. As can be seen from the figure, out of the cycles considered the CSHVR cycle spends the most time at

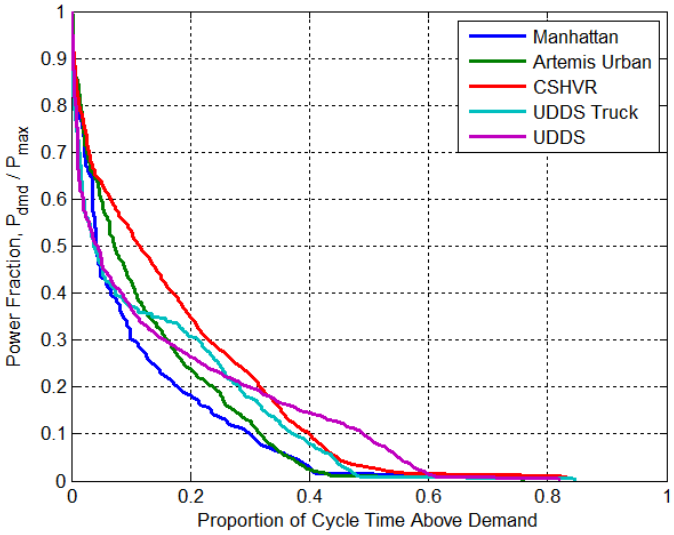


Figure 2.6: Cycle Power Spectra - Scaled Power Values

highest power levels. This would make it more difficult for the hybrid system to accommodate these peak demands and reduce the fuel consumption.

Since the previous section showed that the power demand values for the CSHVR cycle were not excessively high, the velocity profile is reasonable, and the character of the power spectrum of the vehicle demands over this driving cycle are also suitable, the CSHVR cycle has been chosen as the primary simulation cycle for the M1081 vehicle in this study.

2.2 Hybridizing Technologies

One of the fundamental obstacles to the mass production of electric vehicles, perhaps the most significant, is that of onboard energy storage. Comparison of different energy storage technologies can easily be accomplished through descriptive measures of energy and power, such as: specific energy (Whr/kg), specific power (W/kg), energy density (Whr/L) and power density (W/L). These four parameters can be used to compare among different technologies; however, each

technology will also have particular issues with its implementation, which can be critical to its successful use. The hybridizing technologies which will be discussed here include batteries, ultracapacitors, fuel cells, hydraulics, and mechanical energy storage systems. It is important for comparative purposes to note that the first few hybridizing technologies all rely on the electric motor for the conversion of stored energy to mechanical work, while the hydraulic hybrid makes use of a hydraulic pump/motor and flywheels may make use of a mechanical transmission and/or an electric motor, depending on design. A comparison of currently available products implementing each of these technologies is included in Section 2.3.

2.2.1 Batteries

A battery is fundamentally an electrochemical device consisting of two electrodes, an anode (-) and cathode (+), with a separator, terminals, an electrolyte and enclosure [14]. Battery types are typically separated by the electrolyte (solid, liquid, or gel) and battery chemistry. In rating a battery's capacity, both the cell voltage and Ampere-hour (Ah) rating are usually given, and the battery discharge is rated on a "C" scale, with 1C corresponding to a discharge current in amperes equal to the Ah rating of the battery. Additionally, the cycle life until the battery retains only 80% of its rated capacity is widely used as a comparative measure of how long different batteries will last. When assembled into a pack, cells in parallel will add capacity and power capability, while cells in series are used to increase the pack voltage. In a vehicle battery system, each cell in series should be equalized with other cells in order to prolong the life of the battery pack and maintain a balanced pack. This is normally a function of a device called a battery management system or BMS, the functions of which can vary from voltage balancing to full thermal management of the pack. While not discussed here, this critical component of any hybrid electric vehicle (HEV) has been the focus of much research and development efforts.

The most common batteries used in vehicle applications are lead acid, nickel metal hydride (NiMH), lithium ion (liquid or gel electrolyte), lithium polymer

(solid electrolyte), and high temperature systems [14]. Currently, many different companies are researching different types of battery chemistries in hopes of creating the best battery for vehicle applications.

Critical features for hybrid vehicle batteries are those of high energy capacity, high discharge power capability, stable discharge characteristics for consistent performance over that range, long cycle life, safe operation (with acceptable failure modes), and recyclability [14]. These quantities are all very subjective; however, after a brief discussion of battery chemistry and a survey of currently available technologies, we hope to have a better handle on what makes a good hybrid vehicle battery.

2.2.1.1 Lead Acid

While lead acid battery technology has been the most widely used in vehicle and utility power applications, new superior technologies are rapidly gaining acceptance in this area [14]. Current lead acid batteries are capable of reaching a specific energy of up to 45 Whr/kg at low discharge rates and specific power up to 245 W/kg for short bursts of current [14]. As is the case with all battery chemistries, increased power capabilities will come at the cost of lower energy, an important tradeoff for vehicle systems. Benefits of lead acid technology is its ability to deliver a short burst of high current, which is one reason for its widespread use as an automotive starting battery. Issues with using lead acid batteries for hybrids are numerous, most notably the comparatively low specific energy and weight, which can greatly reduce a vehicle's acceleration performance. Another critical issue is that of the lead battery's need to "form up" capacity at the beginning of its useful life, its self-discharge characteristics, and cycle life limitations [14]. To achieve the rated capacity of a lead acid battery, it first needs to be cycled a few times to activate the compounds which undergo the reversible chemical reaction. In addition, in order to discharge to its rated capacity, lead acid batteries need to be discharged at a very low rate, usually less than a tenth of its rated capacity per hour (C/10). Discharging faster than this rate will decrease the battery capacity and affect cycle life. If left charged, a lead acid battery can self-discharge

upwards of 15% of its capacity per month [14]. Additionally, lead acid batteries tend to have relatively short cycle life at significant discharge depths, between 300-500 cycles [14]. This, along with the significant weight of a lead acid pack makes it unlikely to be a candidate for any future electric or hybrid vehicles.

2.2.1.2 NiMH

NiMH batteries were developed as a replacement for nickel cadmium rechargeable batteries in the early 1990's [9]. Current NiMH cells are capable of 60 Whr/kg and 180 W/kg discharge power [9, 14]. In lab tests, one source describes NiMH cells which are able to achieve up to 110 Whr/kg specific energy or 2000 W/kg specific power, but commercial products are only capable of 80 Whr/kg with 2000 W/kg specific discharge power [20]. The higher continuous discharge capabilities of NiMH cells are due to low internal resistance and cycle life has been demonstrated at 1000 cycles [9, 14]. An additional benefit of the NiMH technology is the cell materials, which are more environmentally friendly than many other batteries which use toxic heavy metals [9]. While this technology has been available for over a decade, its introduction into the hybrid vehicle market has taken some time due to a variety of technical challenges. NiMH cells have a low cell voltage, usually around 1.2 volts, which means that significant numbers of cells in series are needed to achieve the 300 plus volts usually seen in vehicle packs. This cell chemistry is also more sensitive to heating effects and should have some form of thermal protection in vehicle battery systems [14]. An additional issue is that of self-discharge, where the NiMH batteries can lose 25% of their energy in a month [14]. This makes these cells seemingly less attractive for vehicle applications, however the NiMH technology has been used on the successful Toyota Prius hybrid vehicle platform for the past decade.

2.2.1.3 Lithium

Many of today's electronic devices now run on rechargeable lithium batteries. A wide variety of lithium battery chemistries exist, including "lithium ion"

chemistries such as cobalt, nickel, and manganese oxides, and "lithium polymer" chemistries such as vanadium oxide, iron phosphate, or other oxides [14]. Due to the lighter weight cell materials, lithium batteries also can store the same amount of energy in around half the size and weight of NiMH [14]. Lithium storage batteries are claimed to be capable of greater than 115Whr/kg specific energy and 850W/kg specific power according to [14]. The battery pack of the popular Tesla Motors Roadster is has an energy and power density of 120Whr/kg and 450W/kg, which is quite significant compared to other production vehicles [40]. In fact, currently available lithium cells have pushed this envelope even further, with specific energies around 175Whr/kg and specific power capability well over 2000W/kg [17]. Lithium cell voltages, depending on the particular chemistry, range between 3 to 3.7 volts, which is significantly higher than other chemistries [14]. This high voltage allows for fewer series connections and less complicated battery management compared with NiMH or lead-acid packs. Lithium batteries are also more environmentally friendly, depending on the particular chemistry involved [14]. Issues with many lithium ion chemistries are safety concerns arising from the cell's high reactivity, which can lead to thermal runaway or even fire in the case of overcharge or cell damage [14]. In the case of a cell rupture, lithium metal reacts with oxygen to produce heat and potentially start a fire [14]. These concerns are paramount for all vehicle applications, where the battery must be able to withstand a crash without endangering the lives of the passengers. Many different companies have spent time researching solutions to these concerns, and as a result, many advances in the safety of these technologies have been made. Many recent advances have been into the area of lithium polymer batteries, which are very similar in many regards to lithium ion batteries, but make use of a solid electrode to prevent the dangerous lithium metal reactions [14]. In this type of battery, performance is very similar to that of lithium ion, with the added benefit of increased safety. It is highly likely that these lithium polymer batteries will be the battery of choice for the future hybrid electric vehicles, although other energy storage technologies have different promising aspects.

2.2.1.4 Other Chemistries

A variety of other battery chemistries exist which have been explored for vehicle use. A sodium-sulphur high temperature battery system was used on the Ford Ecostar concept vehicle which had a reported specific energy of 120Whr/kg and specific power of 188W/kg [19]. While these numbers seem promising, the sodium-sulphur chemistry requires the battery to be maintained at an extreme temperature of 300-350^{o}C, which requires a long startup time and significant energy use just to maintain [19]. The battery pack was only able to maintain itself at the needed temperature for about two weeks before it would need to be recharged, which is a significant drawback for its use in any hybrid vehicle [19]. An example pack of 32kWhr weighed 265kg with only a little more than half the weight being that of the batteries, which highlights the significant amount of thermal controls and protection required for this kind of pack (not to mention the added costs of those systems) [19].

Sodium chloride-nickel high temperature batteries have also been investigated for use in a vehicle setting. These "molten salt" batteries have shown specific energy and power ratings of 90Whr/kg and 75+W/kg respectively, and should achieve up to 1200 discharge cycles [19]. Like the sodium-sulphur cells, the NaCl-Ni cells require an operating temperature around 300^{o}C as well as significant auxiliary cooling systems for temperature maintenance [19]. A preliminary pack of 29kWhr weighs around 320kg, which includes an completely sealed vacuum-insulated battery box with cooling and heating system [19]. While looking at the numbers alone, these sorts of chemistries seem promising, but when considering the stringent requirements for energy storage in the harsh vehicle environment as well as the alternative technologies being developed, it is likely that we will never see a high-temperature battery system on a production vehicle.

2.2.2 Ultracapacitors

Recently, much interest has been aroused around the ultracapacitor as a potential hybridizing technology, both in hybrid passenger vehicles as well as on

larger platforms, such as buses. Starting from the basics, a capacitor is an electrostatic energy storage device that relies on charge separation to store energy. Conventional capacitors use simple conductive plates separated by a dielectric material [27, 39]. Through improvements in electrode design and separator construction that lead to changes in energy content and power capability, an ordinary capacitor becomes either "super" or "ultra" [27, 39]. The difference between supercapacitors and ultracapacitors is only in their construction; supercapacitors use porous carbon electrodes with high surface areas and do not involve an electrochemical reaction (Electrostatic), while ultracapacitors store their energy electrochemically in a polarized liquid layer which occurs at the interface between the electrolyte and electrode (Electrolytic) [27, 39]. Many times the terms supercapacitor and ultracapacitor are used interchangeably, however it is important to note that these are two distinct technologies with somewhat different operating characteristics. Subsequent discussion will use the term ultracapacitor in reference to Maxwell's ultracapacitor products from [38]. Equation 2.4 gives the energy contained within a capacitor assuming a constant capacitance.

$$E_{cap} = \frac{1}{2} C_o V_c^2 \tag{2.4}$$

Where C_o is the nominal capacitance in Farads and V_c is the capacitor voltage in Volts. As an energy storage device, ultracapacitors have much higher specific power but significantly less specific energy than batteries [35]. While recently printed books claim that current ultracapacitors have up to 2Wh/kg and 3000W/kg , current manufacturers spec sheets show modules capable of 4Wh/kg and upwards of 7000W/kg [35, 38]. Very low series resistance as well as relatively high cell voltage of up to 2.7 volts allow ultracapacitors to discharge their small amount of energy very quickly to meet peak power demands [35, 38, 39]. The lack of chemical reaction also makes them safe and virtually maintenance free [39, 53]. The biggest advantage of ultracapacitors in a vehicular application is that of cycle life; manufacturers specifications claim over one million cycles of useful life [38].

In a vehicle, where power demands can be highly transient, the high current capabilities of ultracapacitors have the potential to be very useful as a hybridizing

technology to meet peak power demands as well as capture regenerative braking energy [35, 19, 39, 53]. A very promising future hybrid vehicle configuration makes use of high energy density batteries as well as high power ultracapacitors to meet the variety of demands that a vehicle might encounter over its useful life [35, 19, 53].

2.2.3 Fuel Cells

Fuel cells operate on the principle of thermodynamic reversibility and were first developed through the work of Sir William Grove in the mid 1800's [53]. The fuel cell functions by converting hydrogen and oxygen gases into water via a catalyst, the result of this reaction is also a generated electric current [14, 53]. A variety of types of fuel cells exist, such as: Alkaline (uses KOH), Proton Exchange Membrane (PEMFC - uses a polymer electrolyte), Phosphoric Acid (PAFC), Molten Carbonate (MCFC), Solid Oxide (SOFC - uses solid-doped zirconium oxide) [14, 53]. The operating temperatures of these different technologies widely vary from 20^{o}C to nearly 1000^{o}C, while the efficiencies can range up to 60%, with the fuel cell achieving higher efficiency at partial loads [14, 53]. The PEM fuel cell is quickly becoming the preferred fuel cell technology for vehicular and small power generation needs due to its high power to weight ratio up to around 120W/kg for a stack [53]. A diagram of a single cell, which combines the anode, cathode, and electrolyte together into the membrane electrode assembly(MEA) is shown in Figure 2.7 [53]:

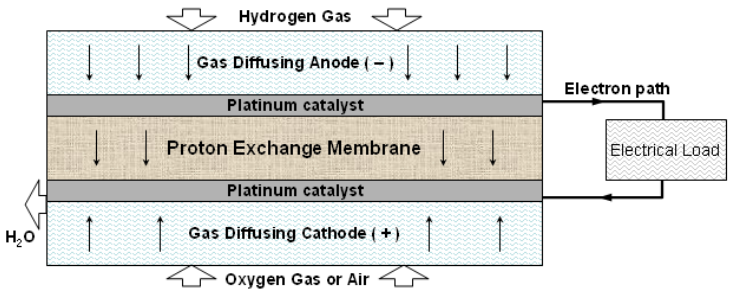


Figure 2.7: Proton Exchange Membrane Fuel Cell

In comparison with other energy technologies, each fuel cell has an open-circuit voltage of 1.25 volts, but the working voltage of 0.6-0.7 volts requires many cells connected in series to achieve higher system voltages [14, 53]. Much research has gone into improving fuel cell technology, especially in the areas of the usage of expensive platinum catalysts, power density of the stack, hydrogen storage, and effective packaging for a vehicle platform [14]. While a typical internal combustion engine has a volumetric power density of approximately 1kW/L and costs between 20-30$/kW, fuel cell systems have been demonstrated in the range of 0.3-0.5kW/L with significantly higher costs [14]. Much technological improvement is needed for fuel cell technology to become more viable, especially in the area of on-board hydrogen storage, which greatly affects both vehicle range and energy density figures.

Hydrogen can be stored in three forms, as a highly compressed gas, cryogenically cooled liquid/solid, or absorbed into a metal hydride [35]. To achieve a comparable range to that of a ICE vehicle, around 6kg of hydrogen needs to be stored (equivalent to around 22liters/5.8gal gasoline)[35]. In compressed hydrogen storage, not only is a significant portion of the energy being stored required to compress the gas initially (about 25%), but significant pressures are required and large volume/weight storage systems needed to hold the gas [35]. Additional issues are that of hydrogen leakage (simply due to molecular size) and the important safety concerns associated with a combustible compressed gas in the case of accidental tank damage [35]. Storing hydrogen cryogenically seems to be a better alternative, however many other issues associated with storage at 20K or -260^{o}C, such as effective insulation, hydrogen boil off, and temperature maintenance are quite technically challenging and costly to address [35]. The most promising solution as of now is the use of metal hydrides, which can store hydrogen in a stable form until it is needed [35]. These compounds only hold a percentage of their weight as hydrogen and therefore the particular chemistry, surface area, and metal hydride properties are of primary design concern to minimize the weight and volume of the system [35]. A quick comparison of the three hydrogen storage methods discussed clearly shows that metal oxides (in particular Mg-H2) are the most promising candidate for vehicle storage: a storage system for 6kg of hydrogen as a compressed gas requires 375 liters and 395 kg, as a cryo-

genic liquid this takes 86 liters and 140 kg, and a metal hydride Mg-H2 system requires only 73 liters and 175 kg [35]. Future developments will lead to better storage technologies, which in turn will make the fuel cell vehicle more practical and commercially viable.

2.2.4 Hydraulic Storage

Unlike the previously mentioned hybridizing systems, hydraulic systems do not make use of an electric motor for the conversion of stored energy into mechanical work. Hydraulic hybrid systems are made up of four components, which include the energy storage tanks (high pressure accumulator and low pressure reservoir), the hydraulic pump/motor, as well as the necessary control valves and actuators to determine the power flow. Figure 2.8 shows these hydraulic hybrid system components:

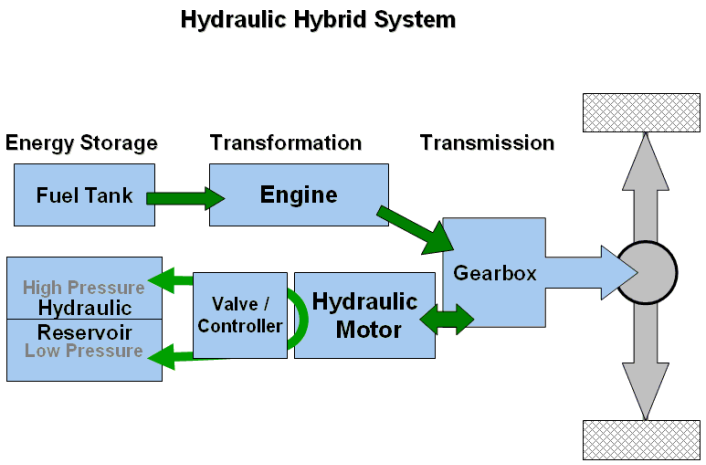


Figure 2.8: Schematic of a Hydraulic Hybrid Vehicle

In the actuation of the hydraulic propulsion assist, the high pressure fluid in the accumulator is discharged through the hydraulic motor to the low pressure reservoir. To capture regenerative braking energy, or to recharge the accumulator from another source of power, such as an ICE in a series hybrid, the hydraulic motor is operated in reverse as a high pressure pump. Some of the claimed benefits of

a hydraulic hybrid system, over and above the obvious benefits in fuel economy and emissions that all hybrid vehicles can achieve, is that of lower incremental cost (due to the readily available nature of hydraulic components) and good performance enhancement (due to the high power density hydraulic pump/motors available) [29, 31]. In terms of energy storage, a hydraulic accumulator's performance is somewhat similar to that of an ultracapacitor; the accumulator can store a relatively small amount of energy (2-5Whr/kg) but are capable of discharging this energy very rapidly (1000-7000W/kg) depending of course on the available accumulator pressure [16]. Issues not frequently mentioned but significant for hydraulic hybrid vehicles are those of noise and vibrations from the hydraulic systems, which need to be addressed for widespread implementations to be feasible [4]. It is likely that many of these issues can be mitigated with further technological developments in hydraulic hybrid technologies.

2.2.5 Mechanical Storage Devices

Mechanical storage of energy, usually in the form of a high-speed flywheel, has also been the subject of much debate and research for hybrid vehicles. As was previously mentioned, for the 2009 season the FIA (association governing the Formula One racing circuit) implemented a mandate to employ a KERS system able to provide a boost to each vehicle on every lap [11]. To achieve this, many teams turned to the carbon-fiber flywheel, a technology which can pack a lot of power and energy into a small weight and space, exactly what the F1 teams needed [11]. When considering the weight of an F1 car, around 600kg, and the remarkable amount of power available already (sometimes greater than 550kW) it is obvious that a highly engineered (and costly) system would be needed to successfully boost performance in this extreme environment [11].

As for the basics of flywheels, they are relatively simple devices which store energy as rotational kinetic energy, sometime rotating at speeds up to 60,000rpm [19]. Many light trains or buses have made use of flywheel storage for some time; where typically the flywheels were heavy and made of steel discs, newer advanced flywheels made of high-tech composites are being made lighter and

spinning faster [19, 35]. The energy of a flywheel follows a simple relationship given below [35]:

$$E_f = \frac{1}{2} J_f \omega_f^2 \tag{2.5}$$

As Equation 2.5 shows, the energy stored in a flywheel increases linearly with rotational inertia J_f and with the square of angular velocity ω_f. Depending on the construction of the flywheel, it is claimed possible to achieve up to a 98% charging efficiency, even though a highly refined prototype demonstrated only 86% [53, 11]. However, a couple of issues surrounding the use of flywheels in hybrid vehicles still are worth noting. First, while new advanced composite flywheels look promising with the theoretical energy density of around 200Whr/kg, the practical maximum energy content is much closer to 50Whr/kg due to containment and other packaging weights [35, 53]. The use of technologies like magnetic bearings and vacuum confinement chambers increase the capable rotational speed of the flywheel, but to effectively use it we still need to be able to charge and discharge it through the use of a continuously variable transmission (CVT) or electric motor [53]. While permanent magnet or switched reluctance motors might be very appropriate to use, their construction into the mechanical flywheel presents its own challenges [35]. Additionally, a spinning flywheel will induce gyroscopic effects, making a vehicle more difficult to maneuver, a phenomenon which is usually addressed by the use of two counter-rotating flywheels [35, 53].

Safety is also of paramount concern if flywheels are to seek application in passenger hybrid vehicles. Almost all failures in a flywheel system are catastrophic, leading to a nearly instant release of a great deal of energy, which the flywheel casing must be able to contain [35, 53]. Composite flywheels are better for this type of failure, since they tend to splinter and fragment in a fluid-like failure as opposed to the fractural failure of a steel flywheel [53]. A recent failure mechanism proposed to increase the safety of composite flywheels is that of the mechanical fuse, a slight necking in the flywheel's geometry near the outer edge where a failure is more likely to occur, thus enabling the casing to be designed to withstand this much smaller failure [35]. While current designs make flywheels unlikely candidates for hybrid vehicles, it is likely that we may see a mass-consumer flywheel hybrid someday in the future.

2.3 Comparisons of Available ESS Technologies

With the continuing development of technologies in the area of hybrid vehicles, a few trends are worth noting and comparing. The various energy storage technologies previously mentioned can be loosely grouped into two categories, along the lines of low-energy high-power sources and high-energy lower power sources. High power devices are shown to be ultracapacitors, hydraulic systems, and high power mechanical flywheels. High energy systems are those of the three types of batteries compared here, but also include the internal combustion engine and fuel cells, if those were of concern to this study. These groupings can be seen in the comparative plot of specific power and specific energy values shown in Figure 2.9. The values for this figure are based off a broad survey of actual energy storage products currently available in the market. It should be noted that all values presented in this table are those of individual cells (batteries and ultracapacitors) and do not consider packaging losses, which has the net effect of moving the plotted data point towards the origin.

The United States Advanced Battery Consortium (USABC) set forth five-year design goals for the power and energy characteristics of vehicular energy storage systems back in 2000, which are plotted alongside the available technologies and were taken from [3]. These goals are almost completely met, with the exception of the future specific energy target. A more in-depth exploration would show that the hindrance to meeting this goal is the lack of a battery cell chemistry that is practically capable of delivering this kind of specific energy level. A few developing technologies exist that may be able to push into this high energy regime, but these are not mature technologies for current hybrid vehicles. Future generations of vehicles could very well be primarily powered by batteries, however, the specific energy capabilities of ESS technologies need to be further developed if a 300+ mile range is desired for future vehicles.

Within the high energy content grouping, much progress has been made to developing new battery chemistries and improving the power and energy characteristics of existing battery technologies. The fundamental limitation to energy storage capabilities are chemistry based, where new battery chemistries (and

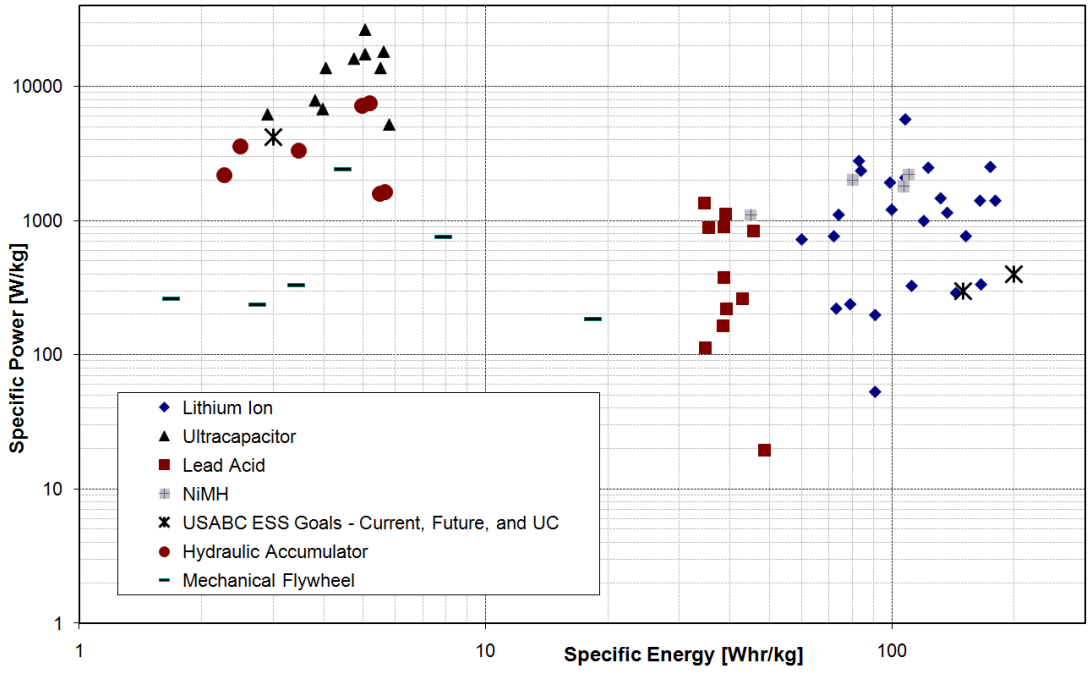


Figure 2.9: Comparative Plot of Available Energy Storage Technologies

alterations to current chemistries) are being slowly improved continuously [9]. Many battery companies have invested in new processing and cell construction methods which have greatly increased the peak power capabilities of many battery chemistries. A difficult tradeoff for battery chemistries to overcome is that the high power discharge of batteries results in less energy storage capability as well as reduced lifetime due to the reactant aging and thermal effects.

In the high power grouping, it is evident from the figure that current ultracapacitor technology has substantially pushed the envelope in terms of cell power capabilities, while at the same time improving the energy content available. The USABC goals for ultracapacitors have been met and improved upon, even when considering how these values decrease for a packaged ultracapacitor module [38]. Hydraulic accumulator technology (and hydraulic motors, as well) are close behind that of ultracapacitors in their energy and power capabilities, however many proponents claim they are better interim solutions due to lower incremental costs and more widespread familiarity with maintenance on hydraulic components

[29]. Much work has been done by the US EPA to investigate the potential benefits of hydraulics on a variety of vehicle platforms, for more information see [4].

It should be noted that the overall design goals of battery-electric hybrid systems vary widely. The third generation Toyota Prius® NiMH battery system is designed to be "low power" and "low energy" with capabilities of less than 20kW and usable energy storage of less than 800Whr [1]. With this sort of ESS capability, even with a 60kW drive motor, this hybridizing system is very limited as to how much it can really benefit the vehicle's fuel economy. Surprisingly enough, the Prius® hybrid system is being employed in many of the mainstream hybrid vehicles available in the market today, and is supposedly the "best" of what is available. A ultracapacitor pack providing the same amount of energy storage would be quite sizeable, but would provide more than ten times as much power capability.

In a heavy vehicle, a main barrier to fuel efficiency comes from short peaks of high power, so an appropriate goal would be to use the most cost effective method for reducing this power demand. A quick comparison of the cell costs from a lithium battery provider and an ultracapacitor provider can give insights into the reasons for selecting ultracapacitors for reduction of power demands on the heavy vehicle platform. The price per kilowatt of available power ranged from $1500 (continuous power rating) to $200 (peak power rating) for lithium batteries. Comparatively, for ultracapacitors the costs ranged from $300 (continuous) down to $15 (peak). A significant point here is that while lithium batteries may be rated for high power (and some are designed specifically for it), the cell's lifetime will severely degrade, sometimes even down to less than 500 cycles when drawn at such high power. Ultracapacitors, on the other hand, will retain a lifetime of hundreds of thousands of cycles even at higher power draw levels. The life cycle costs of using ultracapacitors as a hybrid energy storage system are vastly lower than using even the best lithium batteries, a result corroborated by [39]. Based on these facts, it makes sense that high power dense ultracapacitors are selected for their power assist capabilities in a heavy hybrid vehicle.

A solution for future hybrid vehicles might be to hybridize the energy storage system. In this way, the vehicle would retain the benefits of both technologies,

such as using an ultracapacitor-battery combination where the batteries provide the energy and the ultracapacitors the power. Experiments of both “passive” (direct connection) and “active” (use of controlled DC/DC converter) combinations of ultracapacitors with batteries have shown the potential to improve the power performance of lithium cells by three to seven times, which reduces battery currents and has a positive effect on battery lifetime [33]. Another source explored the use of a neural network-based optimal control to utilize the power boost of the ultracapacitors in combination with a battery, allowing for increased range and reduction in SOC variations of the battery pack [28]. However, this study is concerned with the use of only ultracapacitors as the hybridizing energy storage, since their high power capabilities are potentially useful in reducing the highest engine demands and allowing the hybrid vehicle to achieve better fuel economy.

2.4 Previous Research

A variety of research has investigated different uses of ultracapacitors in hybrid vehicles. Some researchers have investigated different ultracapacitor technologies and products currently available with respect to sizing and the effects this size would have on other hybrid vehicle components and system mass [15]. Another source explored the use of novel motor control methods to improve the regenerative energy capture capabilities of an ultracapacitor hybrid vehicle [47]. This study showed a “major” potential energy savings from the use of a large capacitance but low voltage pack without the use of a DC/DC converter [47].

Other research has explored the use of ultracapacitors on the hybrid bus platform, mostly concentrating on their power capabilities and the benefits these have on fuel economy. In [25] a rule-based control is developed for the control of a ultracapacitor and battery combination on an electric bus, where the benefits were to reduce the battery transients and current draw. An older study outlined the development and future prototype build of an ultracapacitor hybrid bus using a compressed natural gas microturbine, which in simulation demonstrated a doubling of the fuel economy and substantial reductions in emissions on a city

bus cycle [50]. This same cycle is considered here as well and results over this CBD Truck cycle will be seen in a later section.

The research which this study aims to directly improve upon are the results from a previous Clemson researcher in [12]. In this work, the use of ultracapacitors as a standalone hybrid power source is explored to determine the potential fuel economy benefits on a midsize passenger vehicle platform. The vehicle chosen is an SUV with a 160kW engine hybridized using a 70kW induction motor and a 78 Farad 93 Volt ultracapacitor pack. This study demonstrates the potential for up to 15% fuel economy improvement for this vehicle over the UDDS cycle using rule-based power management strategies [12].

Other research ideas have also been incorporated here from Clemson colleagues. In [56], a power management strategy for a battery HEV is created which aims to improve the fuel economy using future road grade information. In this approach, the minimization problem is framed as an instantaneous problem to which the equivalent consumption minimization strategy (ECMS) methods are applied [56]. Another researcher recently explored the use of traffic signal preview information (of the telematic kind simulated in a later chapter) for vehicle velocity planning in order to reduce fuel use of a conventional vehicle [7]. Later, in the chapter on Model Predictive Control, some of the prediction techniques of [24] will be used to help estimate future torque demands.

The following chapter will detail the modeling methods explored in this research, starting with the Powertrain Systems Analysis Toolkit software and building towards the stand alone MATLAB® model used for controls development.

Chapter 3

Heavy Vehicle Modeling

The additional degrees of freedom offered by a hybrid electric vehicle allow for the potential to improve the fuel economy through effective management of the energy and power demands of driving. However, in order to save time, effort, and money in building prototypes and conducting testing to calculate the potential fuel savings, simulation studies are generally employed. In recent years, advances in computational modeling capabilities have led to increased use of high fidelity full-vehicle simulations to investigate the implementation of new technologies and control strategies.

3.1 Problem Motivation

The motivation for this project is to improve the fuel economy of large vehicles through the use of ultracapacitors in a hybrid electric drivetrain. The vehicle targeted in this study are lower speed, heavy transport vehicles, such as military transport vehicles or garbage trucks, many of which currently use diesel engines for propulsion. The specific vehicle studied is that of the M1081, a 2.5ton Light Medium Tactical Vehicle (LMTV) used to transport cargo by the military. Previous work done on the Sport Utility Vehicle platform showed that it was possible

to achieve fuel economy improvements of 10-15% using a hybridizing system of ultracapacitors alone [12]. It is presumed that even larger improvements will be possible on the heavy vehicle platform due to the increased regenerative energy available as well as the larger transients in power demands from the added vehicle mass and size.

In addition to improving fuel economy, hybrid vehicles generally will have increased performance capabilities and decreased emissions when compared with the original vehicle. While these benefits are not targeted directly in this study, they will be mentioned as positive byproducts of the vehicle hybridization. The following sections will detail the modeling and simulation approach taken in this study. First components will be defined based on given data from [6] and simulations will be conducted to validate these modeling choices. The modeling approach used for controls implementations will be covered next, which considers component and system modeling equations also largely from [6]. A comparison of the two modeling approaches is also included, with a discussion of the relevant differences.

3.2 Vehicle Modeling in PSAT

The software package Powertrain Systems Analysis Toolkit (PSAT) is a high-fidelity, forward looking, MATLAB/ Simulink® -based modeling program designed to aid in the development of a variety of hybrid vehicle powertrains [6]. PSAT covers a wide variety of vehicle applications from light to heavy duty and includes numerous data sets (built into the models) which were derived from actual component test data [6]. In developing the PSAT software, a variety of simulated fuel economies were correlated with actual EPA test results on UDDS and HWFET driving cycles, with an overall error of around 5% [6]. This accuracy of modeling is highly dependent on the proper component selection and definition within the model, so if insufficient data is present to characterize a particular vehicle, the accuracy might be less than these test cases [6]. In this thesis, much work has been presented to ensure the proper component selection is both explained and

carried out in the simulations so that the simulation results are both acceptable and realistic.

3.2.1 Component Selection/Definition for Conventional M1081

Each of the critical components in this model (the engine, electric motor, and ultracapacitor energy storage) is defined based on mapped data which comes either from the PSAT software package or from data provided by Maxwell (for the ultracapacitor). This section will present the data sets and component maps for each of the components used in the modeling of the vehicle. These component maps are defined for use within PSAT but are also used outside of the software in a MATLAB m-file for the dynamic programming-based approaches. The components defined for simulations are matched as closely as possible to the specifications obtained for the real vehicle, since exact powertrain component definitions are not available. Before the components are defined, the powertrain configuration is to be defined; this parallel hybrid configuration is depicted in Figure 3.1.

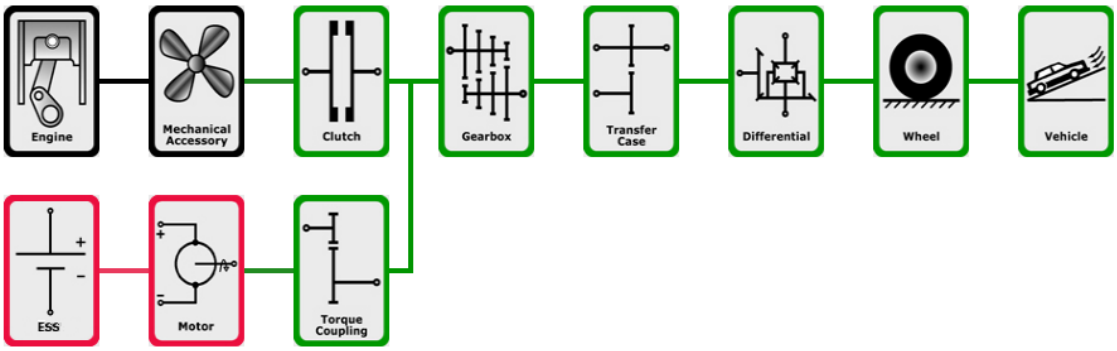


Figure 3.1: Parallel Hybrid Component Configuration

The following components are initially defined in the PSAT modeling environment and then later used in a MATLAB® m-file for the ensuing controls investigations. The M1081 uses a Caterpillar 3115 6-cylinder, 6.6-liter, turbo diesel engine capable of producing up to 168kW (225hp) [13]. Since this exact engine is not available in the PSAT component libraries, a very similar 7.3L, 171kW Detroit

Diesel Series 30 engine is used. The only difference between the selected engine and the real vehicle is the lack of turbo charging, which could change the engine's fuel rate and emissions characteristics slightly. The fuel rate and efficiency are defined based on map data, here shown in Figures 3.2 and 3.3 also displaying the engine power contours.

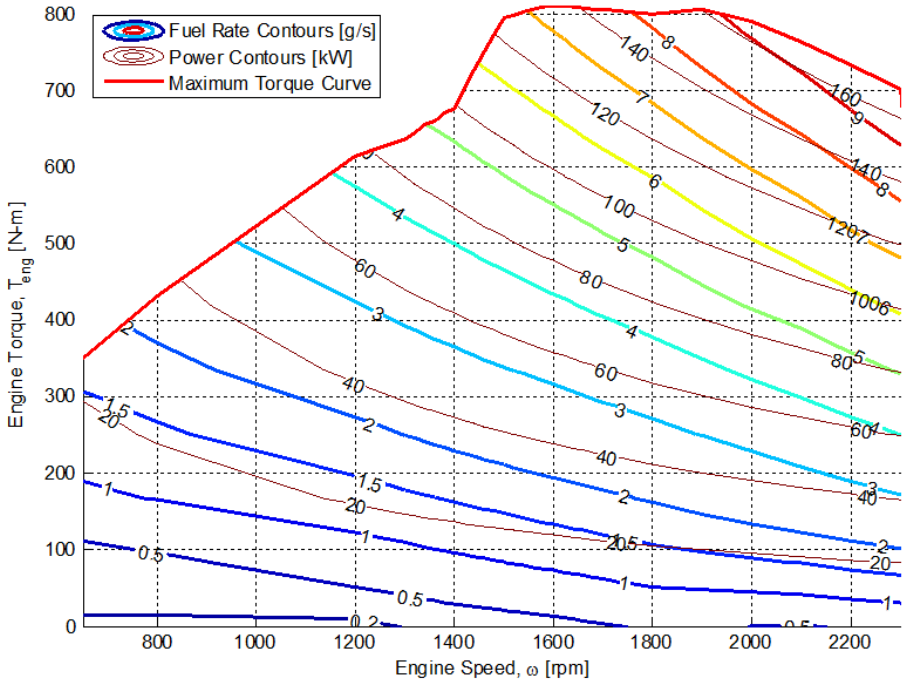


Figure 3.2: Engine Fuel Rate Map

The mechanical accessories for the M1081 are simulated using a constant power loss model. Since the M1081 is a large vehicle with numerous hydraulically driven accessories such as power steering, tire inflation system, and an adjustable air suspension, a constant power loss of 4000W is chosen for this vehicle. Electrical accessory loading is also constant and assumed to be 1500W for the M1081, however, this load is not applied to the hybrid models for reasons to be explained in a later section. These particular values are selected since they are default accessory loads for the heavy vehicle models in PSAT and are also realistic for the vehicle under consideration.

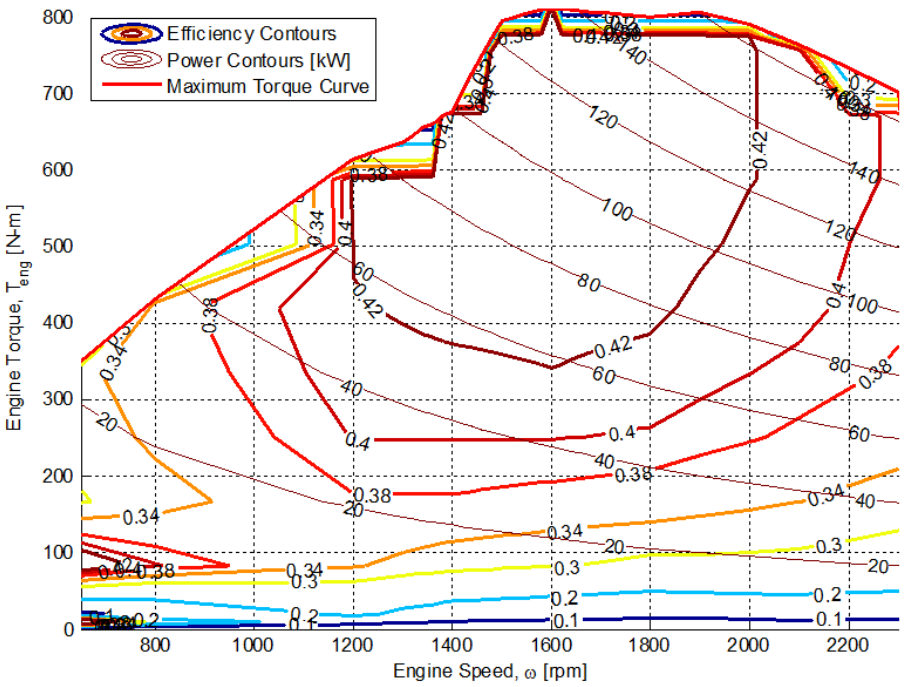


Figure 3.3: Engine Efficiency Map

The clutch model employed is a standard heavy duty vehicle torque converter model with data from PSAT. The transmission model for the M1081 is based on a heavy vehicle transmission from PSAT, but using the correct gear ratios for the seven-speed Allison MD-D7 transmission as given in [2] and in the Table 3.1. The gear efficiencies for these ratios are approximated using constant values similar to other PSAT transmission models.

Gear Number	1	2	3	4	5	6	7
Gear Ratio	5.64	3.45	1.84	1.39	1.00	0.76	0.66
Efficiency	0.90	0.92	0.94	0.96	0.98	0.98	0.98

Table 3.1: Transmission Gear Ratios and Assumed Efficiencies

The transfer case and differential ratios are both configured to match values given in [2]. The transfer case ratio used for the modeling is 2:1 and the differen-

tial ratio is 3.9:1, for an overall 7.8:1 drive ratio.

The wheel data provided in [2] is used in the wheel model as well. A critical component of the wheel model is that of rolling resistance, which is generally hard to get accurate data for. As simulated for the baseline cases, changing between on-road and off-road values of rolling resistance (0.008 and 0.045, respectively, as given in [2]) has a significant effect on the vehicle loading and resulting fuel economy in simulation. Using the on-road value, a fuel economy of 7.83 miles per gallon (mpg) resulted, while for the off-road value the fuel economy is 4.42 mpg. For this reason, simulations will use the wheel model with an average value of rolling resistance equal to 0.025 to better estimate the fuel economy over a range of driving conditions.

The vehicle models in PSAT are relatively simple and contain only a few key changeable parameters, which are also given in [2]. Accurate values for vehicle frontal area and drag coefficient are of great importance to the fidelity of the modeling, and the gross vehicle weight for all simulations is maintained closely across all simulations around the fully loaded vehicle weight of 9700kg.

Simulations conducted in PSAT using these vehicle components comprise the "baseline" vehicle configuration, which is used to validate the models but not as the baseline for fuel economy comparisons, for reasons which will be discussed later. For this case, the vehicle rolling resistance coefficient is set to 0.025 and the control strategy used is the default PSAT strategy for conventional ICE vehicles. The resulting fuel economy for the baseline configuration on the CSHVR cycle is 5.83 mpg, while the same vehicle on the UDDS (FUDS) driving cycle achieves 5.98 mpg. All fuel economy values are given in miles per gallon of diesel fuel. In [2], the same vehicle has been simulated "based on validated models" over the UDDS cycle with approximately the same gross weight and resulted in 5.9 mpg. Based on this corroborative information, the vehicle component definitions are seen to be quite accurate to the original internal combustion engine vehicle's capabilities.

3.2.2 Hybrid Vehicle Component Selection/Definition

The selection of the hybridizing components is important to the potential fuel economy improvements obtainable by the hybrid vehicle. The electric motor/ controller selected for the hybrid vehicle is a UQM PowerPhase 75 from Unique Mobility, a DC brushless permanent magnet motor defined using PSAT-provided data. It should be noted that PSAT simulates the electric motor and controller in one "model" since a given motor usually comes with a matched inverter. The UQM motor is highly efficient, operates at a slightly lower speed range than most induction motors, and has a continuous power rating of 36kW and peak rating of 75kW. This motor is selected for these ratings as well as its wide voltage capability and efficiency, which can be seen in Figure 3.4.

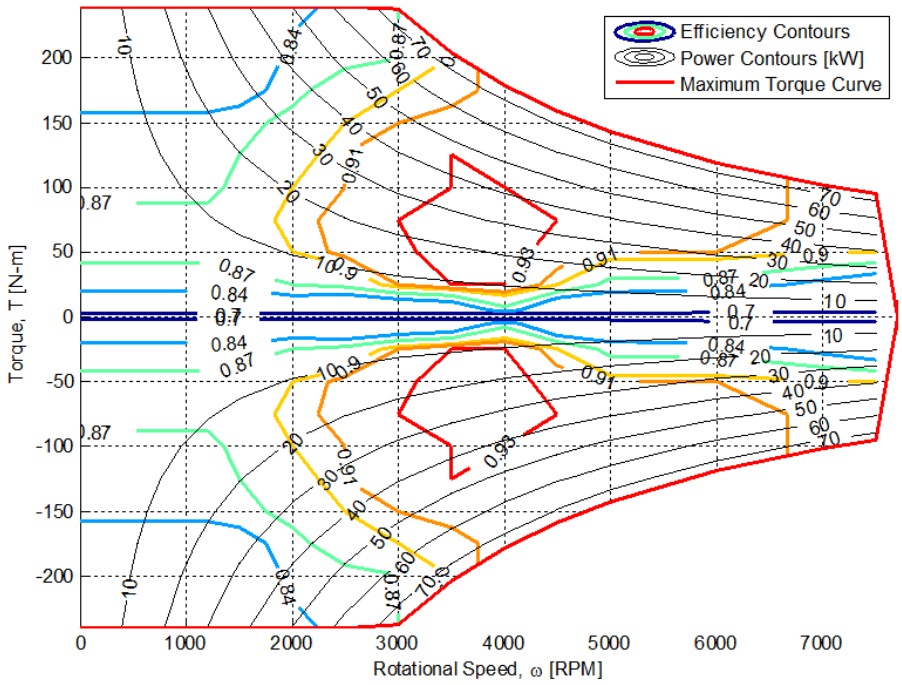


Figure 3.4: Electric Motor Efficiency Map

The torque coupler which connects the engine and motor outputs is a simple constant gear ratio. Due to the speed ranges of the motor and the engine, which

can be seen from the previously shown maps, the standard motor to engine speed ratio used is 2:1. Variations of this ratio are considered in a later section.

The ultracapacitor pack is chosen to contain 54 ultracapacitor cells in series which is equivalent to three 18-cell Maxwell modules. The series combination of three of these 48V modules is the lowest possible to achieve suitable voltage limits within the operating range of the UQM motor/controller chosen.

For an ultracapacitor hybrid electric vehicle, the ultracapacitor model is a critical component for the accuracy of the model simulations. PSAT has a standard ultracapacitor model which is based on map-based data for cell capacitance and resistance. This model, while realistic for many simulations, relies on many simplifying assumptions about the ultracapacitor cells. A more detailed model provided by Maxwell in [37] is detailed and compared to the built-in PSAT model in Appendix A. After the Maxwell model is adapted to run in PSAT and comparative simulations are conducted. In these simulations over the CSHVR cycle, a vehicle using the PSAT model for the ultracapacitor resulted in 7.92 mpg, while the same vehicle using the Maxwell model resulted in 7.82 mpg, a difference of 1.2%. While this is not seemingly significant, the differences in maximum current draw from the ultracapacitor pack are significant enough that the Maxwell model's increased fidelity is important. For further PSAT simulations, the Maxwell ultracapacitor model will be used to define the baseline hybrid vehicle energy storage system.

3.2.3 PSAT Control Methods

In simulation, PSAT uses a simple driver model which consists of a PI controller that follows the desired driving cycle speed trace by commanding a driving torque. Five different sets of driver parameters are available by default in PSAT and the chosen configuration is the "normal" driver with a proportional gain of 1000 and an integral gain of 0.5.

The PSAT powertrain control strategies are split into three categories, with a

separate shifting, braking, and propulsion strategy that can be specified for any model. Each of these strategies contains a set of changeable "control parameters" and the associated Simulink® and/or Stateflow® logic blocks used in the simulation. The strategies of the control blocks in PSAT will be briefly explained in this section.

Since the achievable fuel economy can be easily affected by choice of engine speed and torque, the shifting strategy is an important part of the controller. The shifting control used for all models is that of a "fixed speed" shifting strategy designed for parallel hybrid vehicles for the purposes of fuel consumption estimations[6]. This strategy gives the current gear selection as a function of engine speed and vehicle acceleration for both up-shifting and down-shifting. Using this strategy on the CSHVR cycle results in the engine shifting map shown in 3.5, where the engine speed is plotted as a function of vehicle speed.

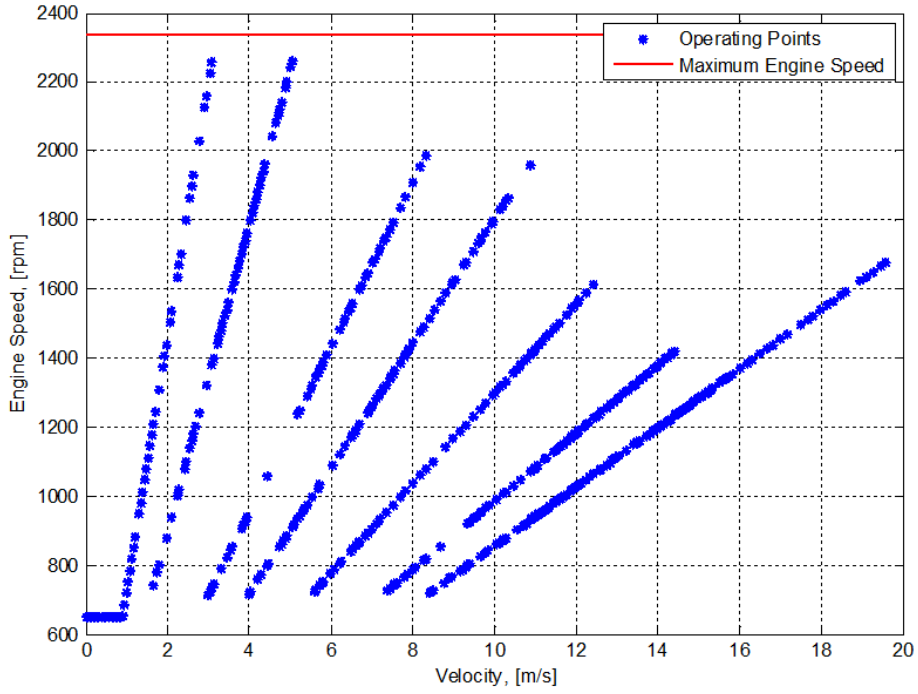


Figure 3.5: Engine Shifting on CSHVR Cycle

Each different gear ratio is easily seen on the map as one of the seven slanted

lines. The shifting strategy imposes both lower and upper limits on the vehicle speed allowed for each gear, and then chooses the appropriate gear based on current vehicle speed and acceleration. In PSAT simulations, where vehicle acceleration is a control parameter, this acceleration is based on the demanded wheel torque from the controller. When this shifting control strategy is used outside of PSAT in MATLAB® simulations, care is taken to mimic the same decisions that the Stateflow® decision blocks make.

The braking strategy in PSAT is relatively simple for both ICE-powered conventional drivetrains as well as for hybrid vehicles. For conventional vehicles, the braking controller just attempts to follow the driver-desired speed trace when the vehicle is decelerating using a friction brake torque command. For hybrid vehicles, the braking strategy seeks to recapture the most regenerative energy possible (without violating established system constraints) and beyond that commands friction braking torque.

The propulsion strategies in PSAT can be quite complicated and difficult to understand, especially for hybrid vehicles. For a conventional vehicle, however, the control block simply passes through the demanded torque by the driver to the engine, after checking it against the appropriate constraints for either maximum engine torque or maximum wheel friction torque. The hybrid vehicle propulsion control has the added challenge of determining the desired power split between motor and engine while ensuring the driving cycle demands are met.

The chosen propulsion strategy for the parallel hybrid vehicle is called the "direct power" controller, which decides on the power split based on torque demand and user-defined controller parameters using control logic implemented through Simulink and Stateflow blocks. More information on the logic implemented in the PSAT controller blocks can be found in the PSAT documentation [6]. The difference between the "direct power" controller and the normal parallel hybrid controller is that the normal controller has an added soft constraint on engine torque. This soft constraint attempts to operate the engine within 2% of the optimal torque to minimize the fuel consumption. This instantaneous optimization criteria actually results in a lower fuel economy than the direct power controller, which shows that constraining the engine torque is not necessarily a

better solution to the control problem. Further discussion of this control problem will be discussed after the baseline simulation results are presented.

3.3 Baseline PSAT Simulations

Once all the components are defined, a set of baseline simulations is carried out. These simulations are run for both the ICE-only powertrain as well as the hybrid configuration over four representative cycles for heavy vehicles as mentioned previously and detailed further in the Appendix: CSHVR, CBD Truck, Manhattan Bus, and UDDS.

The summary of the fuel economies found for the vehicle (ICE compared to hybrid) are tabulated in Table 3.2

Driving Cycle	CSHVR	CBD	Manhattan	UDDS
ICE F.E. [mpg]	5.83	5.12	3.99	5.98
Hybrid F.E [mpg]	7.83	7.70	6.31	7.44
Improvement	34%	50%	58%	24%

Table 3.2: PSAT Simulation Fuel Economies

It is interesting to note the large simulated improvements in fuel economy over each of these cycles when compared with the previously found 10-15% on the smaller vehicle platform. However, at this point some important considerations must be addressed. Since PSAT is a forward looking simulation and the vehicle is not confined to follow the speed trace, the resulting vehicle speed trace needs to follow sufficiently close to the desired trace in order to be a "successful" simulation. In the resulting saved simulation data, PSAT calculates two meaningful statistics about the vehicle's speed-following capability: "Absolute average difference on vehicle speeds" and "Percentage of the time the trace is missed by 2mph". These two statistics capture both the variability and the proximity of the vehicle speed to the desired speed trace and will be used to help quantify whether or not

the PSAT controller is able to "successfully" simulate the vehicle over the cycle.

For the baseline vehicle on the CSHVR cycle, the absolute average difference in vehicle speeds (AAD for short) is 1.32mph, while the percent of the time the trace is missed by 2mph (TTM for short) is 25.2%. Since the hybrid vehicle will have increased performance capabilities, the criteria for the successful simulation will be set at a level consistent with this level of performance: AAD of less than 1.3mph and TTM of less than 25%. Using these criteria, the only PSAT simulations able to be deemed "successful" are those carried out on the CSHVR cycle. The vehicle has difficulty with the other cycles for a variety of reasons: the CBD truck cycle and Manhattan bus cycle have very short spurts of quick acceleration that the PSAT controller has trouble following and the UDDS cycle has portions of high speed with acceleration which vehicle simply does not have the power capabilities to meet. Despite these shortcomings, however, the simulation results presented still show realistic fuel economy values that are similar to what the real vehicle driving the cycle would be capable of achieving.

For fuel economy comparisons and further controls development, the rest of the vehicle modeling is conducted solely in MATLAB® without the use of the PSAT software, though some of the component modeling data and methods are preserved. The details and equations surrounding this approach are given in the next section.

3.4 Vehicle Model for Controls Implementations

A simplified parallel hybrid vehicle model has been created using many of the modeling relationships and component data from PSAT and Maxwell given in [6, 37] and explained in this section. This hybrid vehicle model has a single state, the ultracapacitor State of Charge (SOC), which is defined on a voltage basis in Equation 3.1:

$$SOC = \frac{V_{uc}}{V_{uc,max}} \tag{3.1}$$

where V_{uc} is the ultracapacitor voltage, and $V_{uc,max}$ is the maximum allowable ultracapacitor voltage. This is the definition of SOC that will be used throughout this work. The voltage-based definition of SOC can be related to the SOC defined on an energy basis by the relationship:

$$SOC_E = \frac{V_{uc}^2}{V_{uc,max}^2} = SOC^2 \tag{3.2}$$

These definitions of SOC are only useful for ultracapacitors as battery energy content is not able to be quantified in such a simple manner. For all simulations, the SOC limits for the ultracapacitor pack are set such that 75% of the ultracapacitor energy is usable, that is: $SOC_{max} = 1.0$ and $SOC_{min} = 0.5$.

The control parameter for this hybrid vehicle model is the propulsion torque provided by the engine. This torque demand determines both the engine fuel rate and motor torque which leads to the use of the ultracapacitor energy storage. The modeling equations to show these relations are presented next.

3.4.1 Modeling Equations

Based on the vehicle demands calculations given previously in Section 2.1.3, the torque demand at the wheel is calculated as:

$$T_{dmd,wheel} = Fr \tag{3.3}$$

where F is the total tractive force required due to aerodynamics, road grade, inertia, and rolling resistance, and r is the wheel radius. The wheel torque demand is modified via component efficiencies and the selected gear ratio from the shifting strategy to become the torque demand at the engine:

$$\begin{aligned} T_{dmd} &= T_{dmd,wheel}\left(\frac{r_{gear}}{e_{gbx}e_{fd}}\right) \\ \omega_{eng} &= \omega r_{gear} r_{fd} \\ T_{dmd,eng} &= \max\left(0, T_{dmd}\right) \end{aligned} \tag{3.4}$$

where r_{gear} is the selected gear ratio, e_{gbx} is the efficiency of the gearbox in the selected gear, and e_{fd} is the final drive efficiency. ω_{eng} is the engine speed, determined from the wheel rotational speed ω, the gear ratio r_{gear}, and the final drive ratio r_{fd}. Since the engine is not allowed to provide negative torque and engine braking effects are not considered, the engine demand is limited to the positive part of the vehicle demands at the engine. In this vehicle model, the engine clutch allows the engine to be off when the vehicle is in motion, if this is desired. The clutch efficiency is applied only to the commanded engine torque in the calculations. In PSAT, the clutch has a variable efficiency based on slip calculations, however in this modeling a reasonable constant value for the efficiency is selected to best match with PSAT simulation data.

The other drive component efficiencies (not including the electric motor) are all considered to be constant in this modeling. The gearbox has efficiency values specific to each gear ratio, values based on the gearbox model of PSAT. From the torque demand at the engine, the control decision of applied engine torque determines the motor assist torque as:

$$\begin{aligned} T_{mot} &= T_{dmd} - T_{eng} + T_{eng,loss} \\ \omega_{mot} &= r_{TC}\omega_{eng} \\ P_{mot} &= T_{mot}\omega_{mot} \end{aligned} \tag{3.5}$$

where T_{mot} is the motor demand torque, T_{dmd} is again the vehicle torque demand at the engine, T_{eng} is the commanded engine torque, $T_{eng,loss}$ is the engine torque loss to accessories and friction (based on PSAT calculations using the methods of [26]), ω_{eng} is the engine rotational speed, ω_{mot} is the motor rotational speed, r_{TC} is the torque coupler ratio between the motor and engine, and P_{mot} is the motor power demand.

The engine fuel rate $\dot{m}_f$ is also determined by the selection of engine torque and the given engine speed, based off of mapped data shown previously in Figure 3.2 and following the functional relationship:

$$\dot{m}_f = f\left(T_{eng}, \omega_{eng}\right) \tag{3.6}$$

From the values of motor torque and speed, a lookup table is used to find the ultracapacitor electrical power, based on the following formulae:

$$\begin{aligned} P_{UC} &= P_{mot}/\eta \quad for \quad P_{mot} > 0 \\ P_{UC} &= \eta P_{mot} \quad for \quad P_{mot} < 0 \end{aligned} \tag{3.7}$$

where P_{UC} is the ultracapacitor power and η is the motor efficiency at a given operating point. The ultracapacitor pack current can then be calculated as given in Equation 3.8, which is then modified with a term accounting for the charging efficiency of the ultracapacitor. The reasons for the inclusion of this charging efficiency term are mentioned subsequently in the comparisons of Section 3.4.5. For more information on this derivation see Appendix A.

$$I_{UC,0} = \frac{-SOC \cdot V_{\max} + \sqrt{(SOC \cdot V_{\max})^2 - 4R_s P_{UC}}}{2R_s} \tag{3.8}$$

$$\begin{aligned} I_{UC} &= \eta_{chg} I_{UC,0} \quad for \quad I_{UC} > 0 \\ I_{UC} &= I_{UC,0} \quad for \quad I_{UC} < 0 \end{aligned} \tag{3.9}$$

where I_{UC} is the ultracapacitor current, SOC is voltage-defined state of charge, $V_{\max}$ is maximum pack voltage, R_s is the ultracapacitor pack resistance, and η_{chg} is the ultracapacitor charging efficiency.

From the current demand out of the ultracapacitor, the rate of change of the state of charge can be found as:

$$\dot{SOC} = \frac{I_{UC}}{C_o V_{\max}} \tag{3.10}$$

where C_o is the pack capacitance. The implementation of these equations allows for the control input of commanded engine torque to determine the output state (the next SOC) of the system.

3.4.2 Modeling Assumptions

In order to allow for the efficient simulation of the vehicle, a few key modeling assumptions have been made. Subsequent comparisons to simulation results from PSAT are used to validate the model and ensure that it is a reasonable and accurate simplification of the PSAT hybrid vehicle model.

The following list gives the modeling considerations and assumptions inherent in the model used here:

- The heavy vehicle model is initialized using parameters and data from PSAT initialization and calculation files.
- Engine torque losses are calculated in the same fashion as PSAT, based on [26]. Engine transient effects and start-stop losses are not considered. Engine fuel rate map data which PSAT extrapolates in Simulink® has been extrapolated in the MATLAB® code. Power losses to mechanical and electrical accessories are assumed constant at 4000W when the engine is on, a value based on losses found in the PSAT diesel engine model.
- The ultracapacitor model uses data from Maxwell [37], but assumes the simple circuit model (See Appendix A) with constant resistance and capacitance values. The pack contains 54 ultracapacitor cells of 3000F nominal capacitance in series.

  A constant line resistance value of 0.05Ω is added in to account for the added resistance of pack and motor wiring connections. This line resistance value was arrived at through approximate calculations using values from [48] and corroborated by simple laboratory experiments, which measured the voltage drop over a short segment of large gauge wire. This small resistance may not seem important for other energy storage systems, but when compared to the combined cell resistances of 0.02 Ω, the line resistance will have a significant contribution to the overall resistance of the ultracapacitor pack. The effects of the line resistance in the model will be considered further in Section 4.4.3.

- Each component efficiency is assumed constant at a value which is matched as closely as possible to the results from PSAT simulations. The efficiencies which are considered to modify the torque demand are those of the clutch/torque converter, gearbox, and final drive. The inertias of all components are lumped into the vehicle inertia.

3.4.3 Comparison to PSAT

The heavy vehicle model relies on numerous data values and modeling techniques from PSAT, however, these are implemented here without the added modeling complexities of Simulink® and Stateflow® . Since PSAT models have been validated through vehicle studies described in [6], it is important to validate the model calculations to the PSAT methods and simulation outputs.

In this case, two areas of concern have been the subject of detailed comparisons for the model. First, the engine fuel rate characteristics will be explored, including the engine torque losses and associated engine calculations. The engine modeling validation is important since the accuracy of the engine fuel rate is critical to obtain realistic fuel economy values. Second, the energy storage system modeling will be explored. When using a small energy storage device like an ultracapacitor pack it is critical that the associated calculations for current demands and state-of-charge (SOC) be accurate. In this case, small changes in energy constitute large changes in SOC, which can greatly affect the use of the ultracapacitor and the resulting vehicle fuel economy.

3.4.4 Vehicle Demands and Engine Fuel Rate Comparison

Calculation of the engine fuel rate is critical to the accuracy of the modeling if the desired outcome is improvement in fuel economy. In order to obtain an accurate fuel rate, the imposed vehicle demands must also be accurate as well. The following section details the PSAT comparisons and the modeling improvements

that resulted from these comparisons.

It should be noted that the vehicle speed in the simulations is not exactly the same; due to the forward-looking nature of the PSAT control method, it does not follow the prescribed path perfectly. A variety of different issues were identified with the help of the comparative simulations conducted over the heavy vehicle cycles:

- The vehicle power demands at the engine are checked for correlation with PSAT values. The correlation coefficient for the power demands is r = 0.94, which is very good considering the differences in the modeling methods.
- The engine fuel rate calculation is checked against the PSAT engine speed and torque demands in the calculations for fuel rate. The resulting fuel economy values matched nearly perfectly (minor rounding discrepancies) for all cycles considered.
- The engine torque and speed demands are dependent on the shifting strategy and gear choice. The gear shifting strategy has been modeled based on the PSAT Stateflow® -implemented shifting strategy and uses logic operations which are as close as possible in the MATLAB® code. Due to the differences in simulation environments, the engine speed is kept lower for the heavy vehicle model compared with PSAT. Since engine losses are generally proportional to engine speed, the lower speed will reduce engine friction losses and improve the fuel economy slightly compared with PSAT.
- The method of calculating the engine torque losses also has been modified through these comparisons. Initially they were not properly accounted for, but after checking with the calculations in the PSAT initializations and the corresponding reference [26], they were found to match with the PSAT values.

Overall, the engine fuel rate comparison showed very good agreement between the programs, with negligible calculation differences for the vehicle power demands, engine torque losses, and the engine fuel rate when compared to PSAT simulations.

3.4.5 Hybrid System Comparison

The other critical comparison to be made is that of the energy storage system. With such a small energy buffer provided by the ultracapacitor, it is critical that the modeling be as accurate as possible, since small changes in energy may lead to large changes in state of charge (SOC). In this case the model in consideration is that of the ultracapacitor cell provided by Maxwell in [37] and also explained in Appendix A which has been adapted to run in PSAT. In order to preserve as many of the modeling aspects of this model as possible in the much simpler MATLAB® environment, a detailed comparison has been performed, with the main issues addressed as follows:

- The ultracapacitor pack power demands can be seen in Figure 3.6. In this plot, the PSAT and MATLAB® -generated curves for each of the parameters were correlated to obtain a measure of modeling accuracy. The correlation for ultracapacitor power which is calculated from the motor torque and speed is r = 1.00.

- Since PSAT uses the same relationships to calculate ultracapacitor current from the power demands, it is not surprising that the MATLAB® program output follows closely. With the inclusion of a charging efficiency for the ultracapacitor, the correlation for ultracapacitor current is found to be r = 1.00.

- Ultracapacitor SOC trajectory over the cycle initially ended at an SOC well above 1.0 (these are unconstrained calculations, to allow for these issues to be seen). A charging efficiency has been implemented at a reasonable value of 92%, resulted in the curve shown in Figure 3.6. This modeling change vastly improved the correlation to the nonlinear PSAT ultracapacitor model to a value of r = 0.94.

- The PSAT simulation environment allows for both a variable resistance and capacitance of the ultracapacitor pack with changing discharge conditions and thermal warm-up effects. These were thought to be important to be

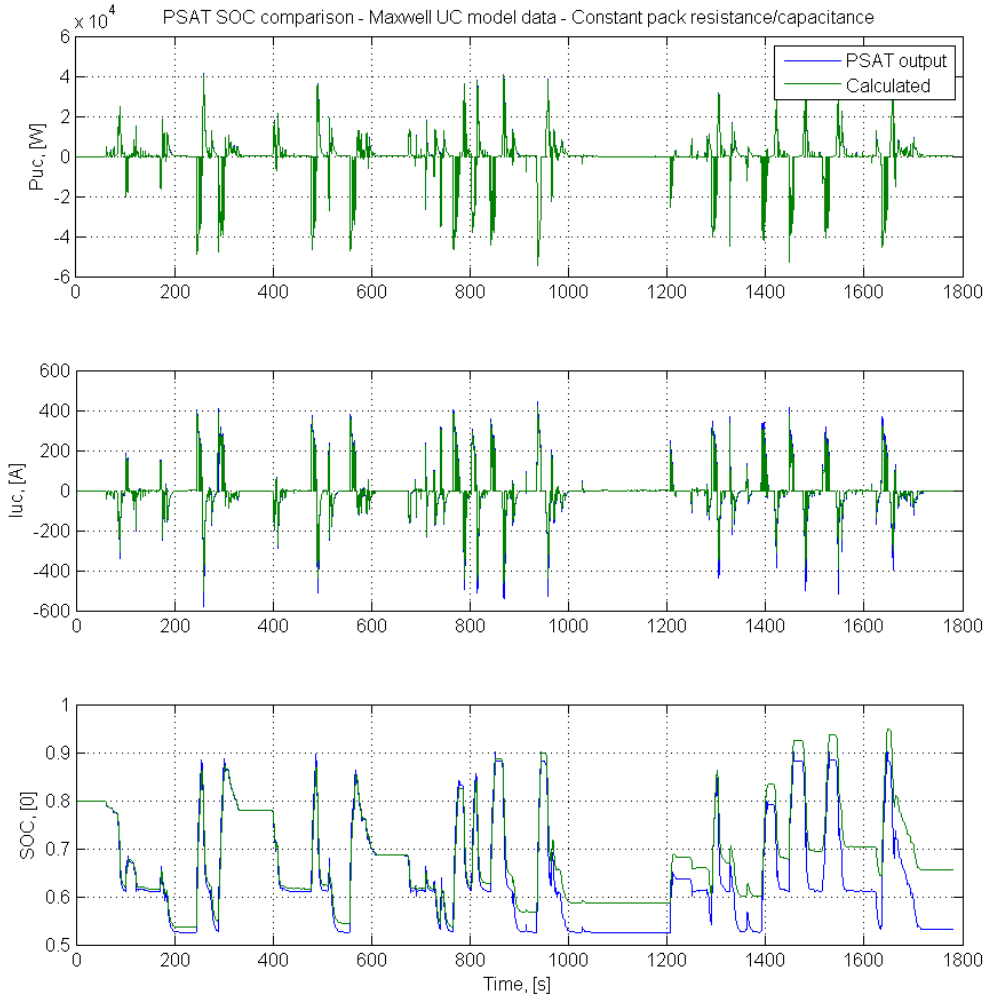


Figure 3.6: PSAT UC SOC Comparisons on CSHVR Cycle

included in the modeling, but were not found to significantly change the comparison results in the cycles studied.

- These correlation-based results held true for comparative simulations on the CBD Truck, Manhattan Bus, and UDDS cycles.

Proper calculation of ultracapacitor pack SOC is essential to the simulation of the ultracapacitor hybrid vehicle. Even a minor change in ultracapacitor pack parameters can cause a deviation in SOC over the cycle, which might result in

different control decisions being made. This would affect vehicle fuel economy, leading to a less reliable simulation result for the vehicle in question. The comparisons conducted here have shown the effects of minor modeling changes to the accuracy of SOC determination and have helped to validate the methods and results of the MATLAB® DP program.

3.4.6 Baseline Simulations on Different Cycles

The internal-combustion powered heavy vehicle is simulated over the different cycles using the MATLAB® -based fuel calculation to establish a comparative "baseline" fuel economy for the hybrid vehicle. These values will differ from the PSAT results due to the different velocity traces followed and the methodological differences between the approaches. The results for each of the cycles is summarized below.

- CSHVR: Baseline fuel economy of 6.94 miles per gallon (mpg), an increase of +19.2% over PSAT simulation, which demanded 41 seconds of maximum engine torque.
- Manhattan Bus: Baseline fuel economy of 4.39mpg, increase of +10% over PSAT, 2 seconds of maximum engine torque.
- CBD Truck: Baseline fuel economy of 5.10mpg, decrease of -0.2% from PSAT, no maximum engine torque.

It is worth noting the baseline ICE-powered vehicle for these simulations is NOT capable of perfectly following the velocity trajectory for the CSHVR and Manhattan Bus cycles due to its very low power to weight ratio. An inherent assumption in all vehicle testing is that there will be some deviation from the prescribed velocity curve, but in these simulations the vehicle is constrained to follow the given velocity. In this case, the low speed CBD cycle is the only cycle considered that the vehicle is fully capable of following the speed trace (without hybrid system assistance). The differences in fuel economy seen on the CSHVR

cycle of almost twenty percent improvement are a result of the time spent at maximum torque where the vehicle cannot meet the cycle demands and the differences between the PSAT control strategy and this simulation. Despite these differences, this baseline fuel economy figure is considered reasonable and realistic and will be used in comparisons to give potential improvement figures from the following MATLAB® -based simulations.

The following chapter will detail the investigation of the "best possible" fuel economy for the hybrid vehicle calculated using the Optimal Control numerical approximation method of Dynamic Programming.

Chapter 4

Optimal Control Investigations

To explore the maximum potential fuel economy benefits resulting from hybridizing the M1081 vehicle, an optimal control problem is formulated. In an optimal control formulation, the objective is to determine a solution to a mathematically formulated control problem which meets the physical system constraints while maximizing or minimizing some "performance criteria" [30]. After formulating the problem to be solved in terms of differential equations, many simple control problems can be solved analytically using one of a variety of methods. However, many more complicated problems may not be solved analytically due to coupling of variables, difficulties arising from enforcing constraining relations, or the non-uniqueness of potential solutions. The control problem for a hybrid vehicle is one such problem.

4.1 Background

An optimal control problem can be solved numerically using the Dynamic Programming (DP) method, which is a numerical method of minimizing a functional originally developed by Richard Bellman [30]. The DP approach relies on Bellman's "Principle of Optimality" which states that in order for an entire solution

to be optimal, the solution from any intermediate point on the path to the end must follow the same optimal path. A brief summary of the method will be presented here; for more information see [30] and the references therein.

The goal of Dynamic Programming is to define an optimal control of the form:

$$\mathbf{u}^*(t) = \mathbf{g}(\mathbf{x}(t), t) \tag{4.1}$$

where $\mathbf{u}^*$ is the optimal control vector (may be time-varying) which depends on the functional relationship or optimal control law g, the system state vector $\mathbf{x}(t)$, and time. The method requires a system to be defined such that:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \tag{4.2}$$

where $\dot{\mathbf{x}}(t)$ is the time derivative of the state vector, $\mathbf{x}(t)$ is again the state vector, $\mathbf{u}(t)$ is the control vector, and $\mathbf{v}(t)$ is the system input vector, all of which can have many components for a general system. The implementation of the DP method essentially reduces to an iterative calculation of a cost function at each time step of the form:

$$J^*(\mathbf{x}, t) = \min_{\mathbf{u}} \{J(\mathbf{x}, \mathbf{u}, \mathbf{v}) + J^*(f(\mathbf{x}, \mathbf{u}, \mathbf{v}), t+1)\} \tag{4.3}$$

where $J^*(\mathbf{x}, t)$ is the optimal cost-to-go from time t and any state in the state vector $\mathbf{x}$ to the end of the computational horizon, which is computed by a minimization over all admissible controls in the control vector $\mathbf{u}$ for the functional shown. The incremental cost for the current control decision, given the current state vector $\mathbf{x}$, control vector $\mathbf{u}$, and system input vector $\mathbf{v}$, is given by $J(\mathbf{x}, \mathbf{u}, \mathbf{v})$. The optimal cost for the future control decisions is based on the resulting state from the control vector, and is accounted for by the last term, $J^*(f(\mathbf{x}, \mathbf{u}, \mathbf{v}), t+1)$. These equations can be implemented in a computational program such as MATLAB® to carry out the Dynamic Programming method. Before this can be done, the appropriate definitions must be made for the cost function, system modeling equations and constraints.

4.2 Control Problem Formulation

In the case of fuel consumption minimization, which is the goal in the optimal control of the hybrid vehicle, the cost function to be minimized is that of the integral of the fuel rate over the path, as is given in Equation 4.4.

$$\begin{aligned} J &= \int_0^{t_f} \dot{m}_f dt \\ \dot{m}_f &= f(\omega_{eng}, T_{eng}) \end{aligned} \tag{4.4}$$

where J is the cost function, which is integrated from the initial time to end of the cycle. The fuel rate $\dot{m}_f$ is implemented as a lookup table in terms of engine speed and torque. This cost function is to be minimized subject to the following constraints:

$$\begin{aligned} \dot{x} = S\dot{O}C &= f(x,u,v) \\ SOC_{min} < x &< SOC_{max} \\ P_{UC,max,charge} < P_{UC} &< P_{UC,max,discharge} \\ I_{UC,min} < I_{UC} &< I_{UC,max} \\ T_{mot,min} < T_{mot} &< T_{mot,max} \\ T_{eng,min} < T_{eng} &< T_{eng,max} \end{aligned} \tag{4.5}$$

where the state of the system is the state of charge of the ultracapacitor $x = SOC$, $S\dot{O}C$ is the rate of change of SOC which is a function of the state x, the control u, and the imposed velocity v. The other constrained parameters $*_{min}$ and $*_{max}$ are the upper and lower limits on the state of charge x, the ultracapacitor power P_{UC}, ultracapacitor current I_{UC}, motor torque T_{mot}, and engine torque T_{eng}. The engine speed is constrained by the shifting strategy. And the constraint values are defined with appropriate values for the component models specified. The constraint values for these parameters are given in the MATLAB® program and are summarized here:

- SOC constraints are fixed at $SOC_{min} = 0.5$ and $SOC_{max} = 1.0$ with SOC defined proportional to voltage, for allowable pack voltages between 73 and 146 Volts.
- The constraints on the ultracapacitor power draw depends on the current

SOC and the ultracapacitor pack parameters, based on Equations 4.6 and 4.7 derived from modeling relationships:

$$P_{UC,max,charge} > \frac{V_{abs,max}\left(SOC \cdot V_{max} - V_{abs,max}\right)}{R_s} \tag{4.6}$$

$$P_{UC,max,discharge} < \frac{\left(SOC \cdot V_{\max}\right)^2}{4R_s} \tag{4.7}$$

where V_{max} is the maximum voltage of the ultracapacitor (2.7 Volts per cell) and $V_{abs,max}$ is the maximum open circuit (charging) voltage that can be applied to the ultracapacitor (2.8 Volts per cell).

- The current from the ultracapacitor is constrained by the maximum allowed to the motor and controller, which is 300 Amps. This value is reasonable for a hybrid drive of this size.
- Motor torque constraints depend on the current motor speed as to whether the motor is reaching the torque or power limit of the motor. This relationship has been shown previously in the motor map of Figure 3.4, where the motor peak torque is 240N-m and power is 75kW for both positive and negative torque values.
- The engine torque constraint also depends on the current engine speed, which has been shown previously in the engine map of Figure 3.3.

Now that the control problem has been formulated, the implementation of this problem will be investigated.

4.3 Energy Management Via Dynamic Programming

The heavy hybrid vehicle model outlined in 3.4 is used by a Dynamic Programming (DP) routine to solve the previously stated control problem in MATLAB® . The code used in this implementation is included for reference in Appendix D.

This numerical approximation method uses the known driving cycle demands to calculate the minimum possible fuel economy. From the wheel torque demands, the problem is discretized over time and the control inputs (engine torque demand) and outputs (state of charge of ultracapacitor). The assumptions and modeling considerations of the DP method are outlined here, followed by the simulation results.

- The DP routine is discretized over 101 points in both engine torque and SOC. This value has shown good torque agreement over the duration of the cycle when compared with lower resolutions. A resolution of one second in the time domain is used.
- Initial SOC value is chosen as 0.8 in accordance with PSAT simulations; negligible changes in the resulting fuel economy come from changing this starting value.
- If the control decision is such that the engine is not required to supply any torque (i.e. $T_{eng,opt} = 0$), then the associated fuel rate is set to zero, as if the engine is turned off.
- In accordance with the DP method, constraints are checked for the engine torque, motor torque/power, and ultracapacitor current/power and SOC to ensure these components are within their respective bounds. If a constraint is violated, a large cost value is assigned to the associated control decision to prevent this choice from being made.
- In order to prevent excessive engine on/off switching, a small "engine switching cost" is added to the cost function to ensure the engine stays on for at least five seconds at a time. Without this value the optimal solution demands for frequent switches of the engine on and off in a manner inconsistent with a practical implementation.

The addition of this engine on/off cost has changed the cost function slightly to the following form:

$$J = \int_0^{t_f} \left(\dot{m}_f + q\Delta_{eng,on/off} \right) dt \tag{4.8}$$

where q is the constant engine on-off cost, and $\Delta_{eng,on/off}$ is the engine on-off switching signal.

These assumptions are used in the DP method to solve the discretized control problem backwards in time through the cycle demands. The following section presents simulation results from the DP method for the purposes of providing an upper limit to the potential fuel economy gains for the hybrid vehicle.

4.4 Simulation Results

A wide variety of simulations have been carried out to investigate the effects of various parameters on the maximum achievable fuel economy. This section will present the results of these Dynamic Programming calculations, starting with the standard hybrid vehicle configuration on the CSHVR cycle. The baseline fuel economy values were given previously in Section 3.4.6, and are also repeated here for convenience.

After the standard hybrid configuration results, the different parameter variations which are investigated are: energy storage system capacitance, resistance, ultracapacitor cell size, energy storage system size (number of ultracapacitor cells), motor torque coupling ratio, and vehicle mass.

4.4.1 Baseline Hybrid Vehicle Simulations

The standard ultracapacitor heavy hybrid vehicle is run over the CSHVR cycle using the Dynamic Programming approach with a resulting fuel economy of 9.75mpg, a 40% improvement over the baseline fuel economy of 6.94mpg. Some of the important outputs from the simulation are shown in Figure 4.1.

As is evident from the figure, significant values of engine torque are still demanded for small portions of the cycle. While 40% improvement may seem like a

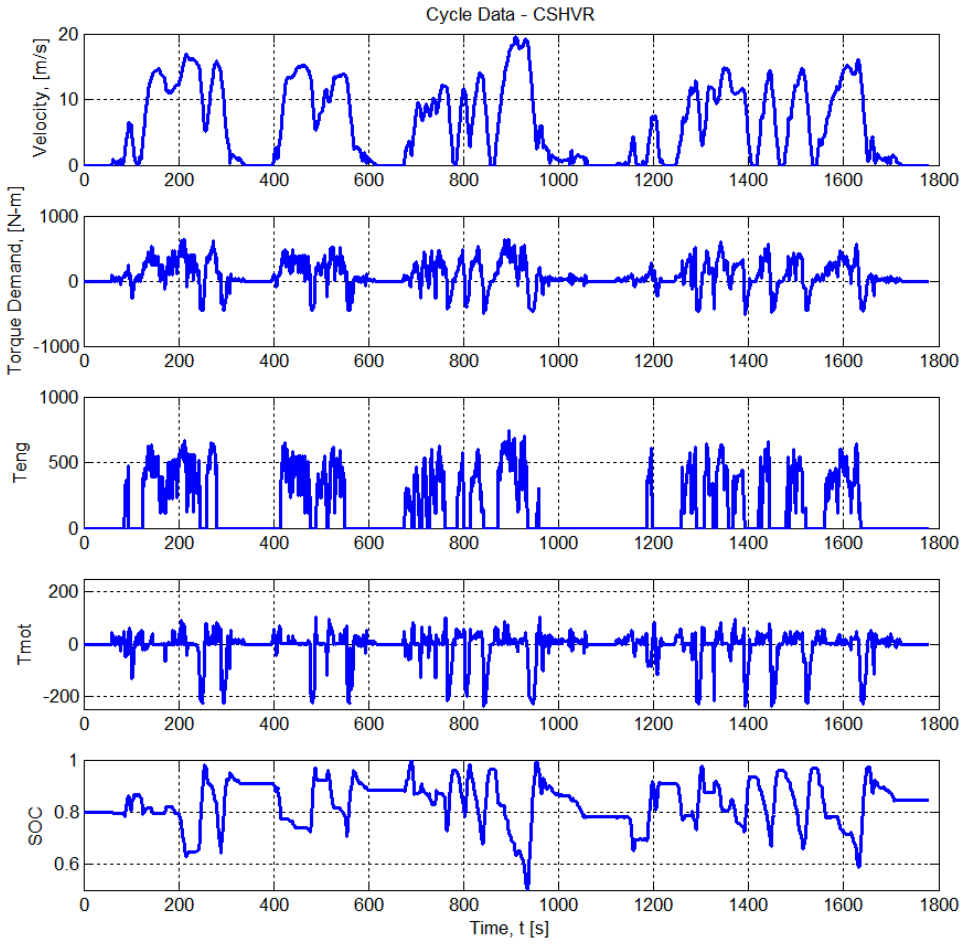


Figure 4.1: Baseline DP Outputs on CSHVR Cycle

lot, around 29% of this improvement is due to the engine remaining off for large portions of the cycle. The engine is able to stay off for 61% of the cycle time (1092 seconds), which can be seen in the third graph in Figure 4.1. The other 11% improvement in fuel economy comes from the decrease in engine torque from the electric motor assist and regenerative energy capture associated with the "optimal" control decisions made.

The SOC of the ultracapacitor energy storage shows a lot of variation and only occasionally approaches the constraint boundaries, as might be expected for a DP simulation. This shows the optimal control decisions are to maintain the SOC of the ultracapacitor within a reasonable region so it can always provide power

assist and capture. To help provide additional insights into the vehicle's operation on the CSHVR cycle, component maps for the engine and motor are shown along with the optimal operating points. The engine map is below in Figure 4.2.

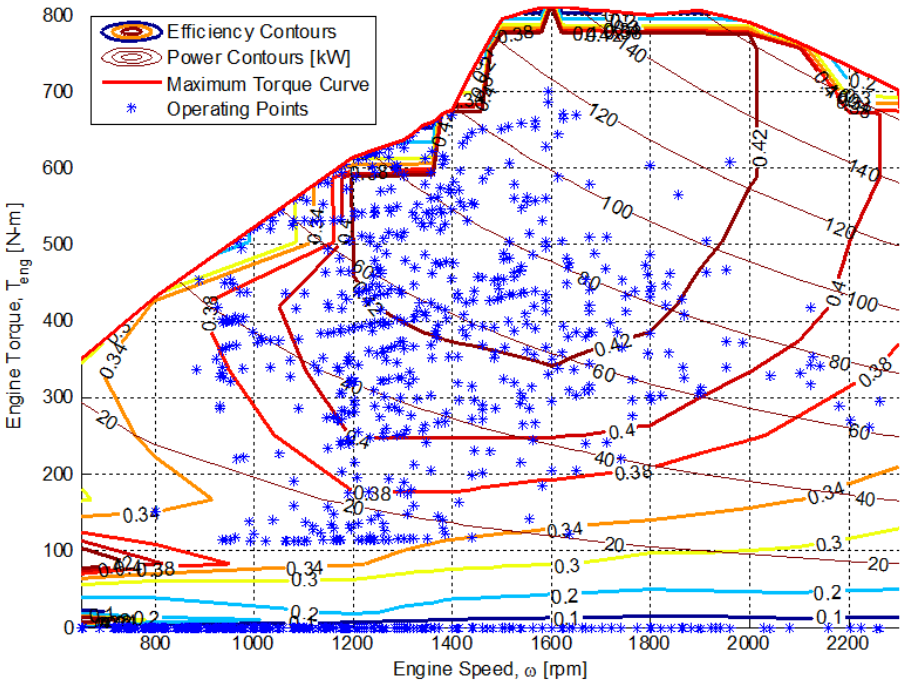


Figure 4.2: Engine Efficiency Map with Optimal Operating Points

The engine map shows how well the electric motor is able to limit the torque demand on the engine. While the engine is sized to power the vehicle and provide up to 170kW of power, it only is needed to use less than 120kW when optimally controlled, with most of the time spent under 100kW. Using this engine map as a guide, it may be possible to determine a good method for downsizing the engine, if this is desired to help improve the fuel economy. However, different engines have different characteristics, so the respective operating range and necessary gear ratios are also needed to be determined at the same time. Changes in the gear ratios will affect the electric motor performance, which changes the system and the optimal solution. It turns out to not be very easy to simply "downsize" the engine and improve the vehicle fuel economy, unless the possible negative effects on performance are disregarded (which we cannot do and still remain faithful

to achieving the vehicle's mission). If the engine is to be downsized, the motor would need to be increased in size accordingly. Next, the motor operating points for the same cycle are considered below in Figure 4.3.

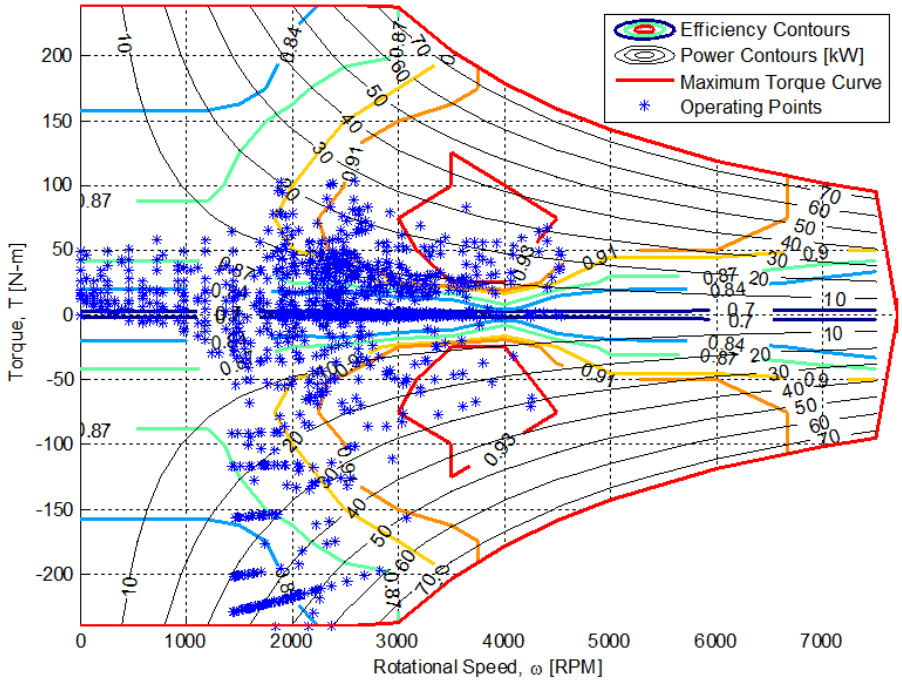


Figure 4.3: Motor Efficiency Map with Optimal Operating Points

As is evident from Figure 4.3, the electric motor is operated at a wide range of speeds and torques but does not fully use the envelope of its performance. At all speeds, only relatively small torque demands are made of the motor. Most of the demands are close to zero, as would be expected for most cycles, and the regeneration torque demands are sometimes more severe than those of propulsion.

An additional improvement which could be made would be to select the appropriate gear ratio with knowledge of the ultracapacitor and motor system's capabilities. This could allow for the slight improvement of the use of the motor system, possibly improving the engine efficiency of operation and the vehicle fuel economy as well.

Additional standard hybrid vehicle simulations are conducted for comparison with the baseline results on other cycles as well, namely on the Central Business District (CBD) Truck cycle and the Manhattan Bus cycle. The DP calculated fuel economy for the CBD Truck cycle is 9.67mpg, an improvement of 89% over the baseline case. For the Manhattan bus cycle, the potential fuel savings is even higher, at 9.29mpg, or a 112% improvement. Substantial improvements like this are mostly due to the lower speed nature of these cycles, with comparatively frequent stop-and-go micro-cycles which make heavy use of the ultracapacitor assist. The engine is also able to remain off for large portions of these cycles, saving substantial amounts of idling time. For more information on these simulations, see the Appendix C. The results shown here are, of course, "best case" figures and might not be realistically achievable using a forward-looking control method.

4.4.2 Capacitance Variations

In order to see the effect of increasing sizes of ultracapacitor energy storage on the resulting fuel economy, the DP method is applied to ten capacitance variations. For this purpose, capacitance values are linearly spaced between the largest and smallest Maxwell "BoostCap" ultracapacitor cells, 3000F and 650F, respectively [38]. These cells all have the same working voltages, and so the resulting range of pack capacitance is from 12F to 56F. At this time, the capacitance values are considered with the same (nominal) internal resistance value for the pack. The resistance effects are considered separately in the next section, as well as changes of both resistance and capacitance together.

The data points of Figure 4.4 show the nearly linear increase in the fuel economy with the increasing pack capacitance. The lowest value of capacitance shows a 27% improvement, compared to the 40% improvement of the largest pack capacitance (baseline case). Some of the non-linearity of the data comes from the slightly different control decisions that are made with different pack capacitances, which mostly come from the decreased energy content of the pack. Looking at the equations, it is noticed that the only term affected by the changing capacitance is that of SOC rate-of-change; knowing this and that the ESS capac-

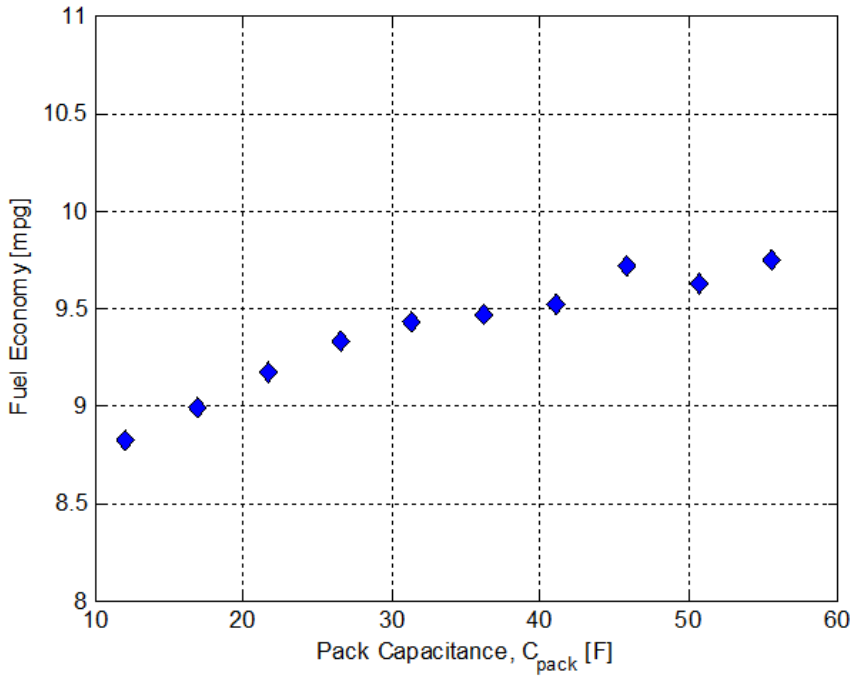


Figure 4.4: Capacitance Variation and Resulting Fuel Economies

itance is proportional to the energy content, it would be expected that lower capacitance values showed more variation in SOC over the cycle. The SOC profile over the cycle is plotted for three different cases of capacitance (Top C=45Farads, Middle C=31F, Bottom C=17F) in Figure 4.5.

As would be expected, the SOC profile shows more variation as the pack capacitance is decreased, corresponding to a decrease in available pack energy. Less capacitance leads to more variability in the SOC as well as the engine and motor torque demands, which are not shown here. Since capacitance is not the only variable in the ultracapacitor energy storage system, the effects of the other major variable, internal resistance, will be investigated next.

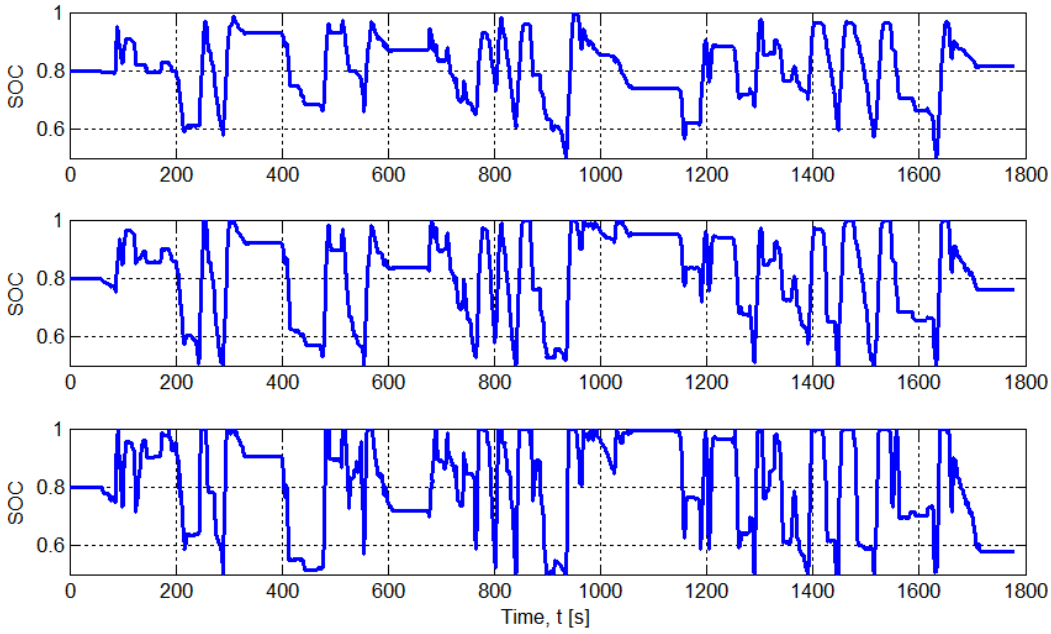


Figure 4.5: Optimal SOC Profile Change With Capacitance Variation

4.4.3 Resistance Variations

It is known from experience (and from basic electrical relationships) that internal resistance plays a major role in the power discharge capabilities (as well as the efficiency) of an energy storage system. In this study, the internal resistance of the pack is changed while the capacitance is held constant at the nominal value. The internal resistance values are reasonable for the variety of Maxwell Boostcap cells considered (3000F to 650F). The line resistance values used correspond to the nominal value of $50m\Omega$ and half this nominal value, to achieve realistic lower resistance values. These resistance values and corresponding pack resistances are shown below in Table 4.1.

Using these resistance values and running the DP code multiple times over the CSHVR cycle the fuel economy results shown in Figure 4.6 are obtained.

It is logical that increasing pack resistance would lead to decreasing the fuel

Line Res. [mΩ]	25	25	25	25	25	50	50	50	50	50	50
Cell Res. [mΩ]	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5	0.6
Pack Res. [mΩ]	30.4	35.8	41.2	46.6	52.0	55.4	60.8	66.2	71.6	77.0	82.4

Table 4.1: Tabulated Resistance Data for Resistance Variations

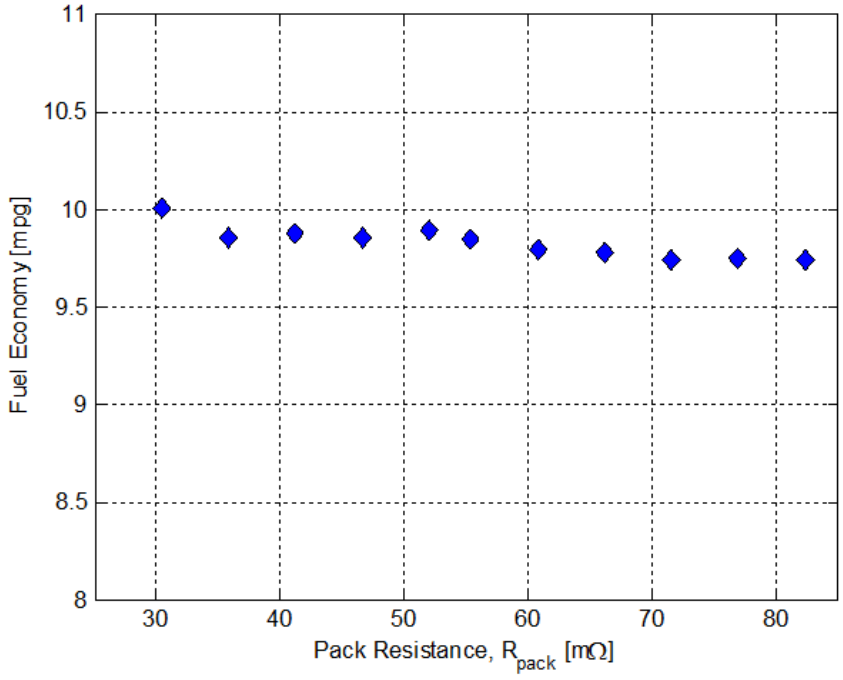


Figure 4.6: Resistance Variation and Resulting Fuel Economies

economy, simply because the added resistance "restricts" the flow of power from the ultracapacitor system. The fuel economies for these variations range from 10.0mpg (+44%) down to 9.72mpg (+40%). Higher resistance values were tested, but resulted in errors due to the inability of the vehicle to meet the required power demands at certain points on the cycle. This is explained by a brief explanation of the power restrictions imposed on the ultracapacitor system.

In the ultracapacitor power constraint Equations 4.6 and 4.7, increasing resistance linearly decreases the allowable power in and out of the ultracapacitor system, so a 50% increase in resistance decreases the ultracapacitor power capability

by 50%. The power limitation of the ultracapacitor pack has been explored, considering the values of the previously mentioned power constraints plotted over SOC for the three pack resistance values shown in Figure 4.7.

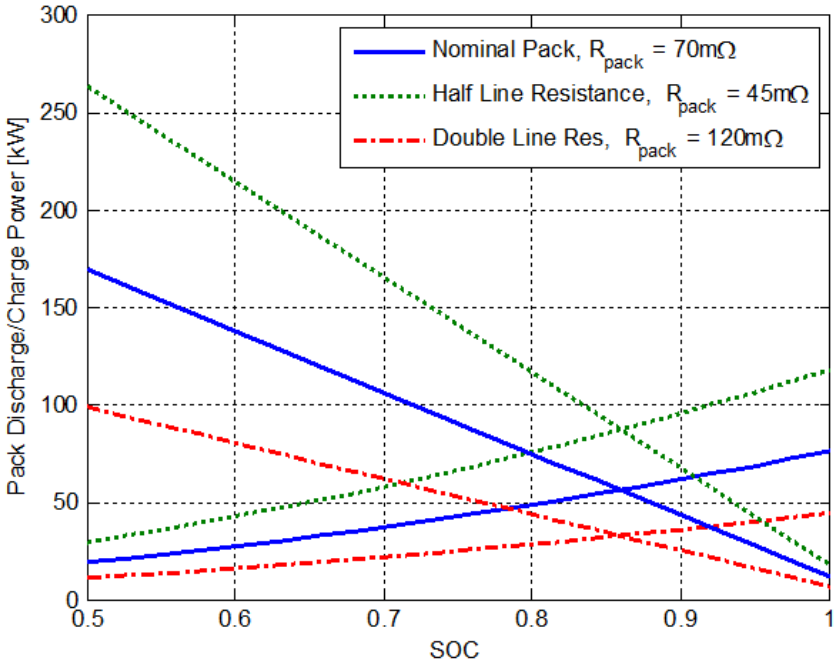


Figure 4.7: Ultracapacitor Maximum Charge/Discharge Power as a Function of SOC

As Figure 4.7 shows, the pack discharge capabilities are high at high SOC's, while the charging capabilities are high at low SOC's. The nominal pack is capable of discharging between 19kW and 75kW of power depending on the SOC. The decrease in line resistance by a factor of two increases this power by about 50%, as the doubling of the line resistance decreases the power capability by about 35%. The charging power capability is a steeper curve and follows the same relationships. The ultracapacitor power discharge increases with the square of voltage, and only linearly decreases with resistance. This means an increase in pack power capability would come from adding cells in either series or parallel, although higher voltage levels would be more efficient for the motor and inverter. Previous research attempted to add ultracapacitor power by adding a string of cells in parallel, but it might be more effective, both cost wise and volumetrically,

to add fewer cells than this in series to improve the power potential and system efficiency [12].

Since in real applications neither the resistance or capacitance changes separately, next both capacitance and internal resistance will be changed together to see the potential effects this has on the optimal fuel economy.

4.4.4 Cell-Specific Variations (R and C)

When considering the rather slight variations in fuel economy from the above variations in capacitance and resistance, it is desired to see how the trends change when both parameters are varied together, as would be the case in selecting among real ultracapacitor products. The trend of increasing cell capacitance leading to a decrease in cell internal resistance is seen in values from Maxwell products in [38]. This relationship is plotted in Figure 4.8 for five large ultracapacitor cell products produced by Maxwell, with the corresponding cell and pack data tabulated in Table 4.2.

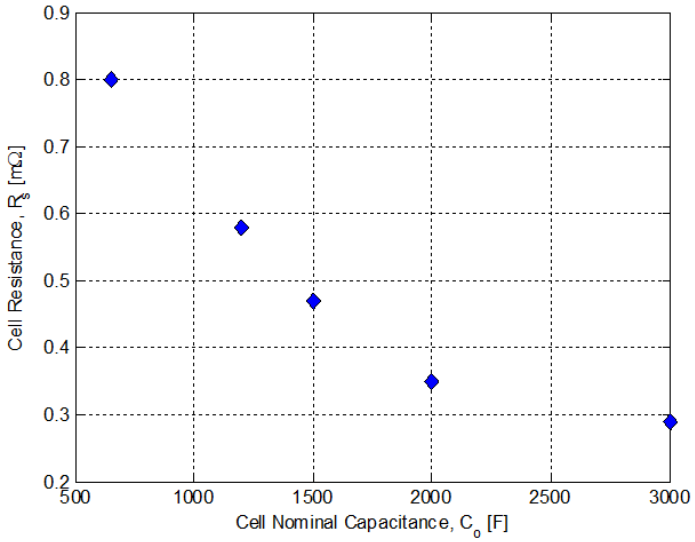


Figure 4.8: Cell-Specific Variations, Cell Capacitance vs. Resistance

The fuel economy results from DP simulations using these ultracapacitor cell

Cell Capacitance [F]	3000	2000	1500	1200	650
Cell Resistance [mΩ]	0.29	0.35	0.47	0.58	0.80
Pack Capacitance [F]	55.6	37.0	27.8	22.2	12.0
Pack Resistance [mΩ]	65.7	68.9	75.4	81.3	93.2

Table 4.2: Tabulated Data for Cell-Specific Variations

values are plotted versus cell capacitance in Figure 4.9.

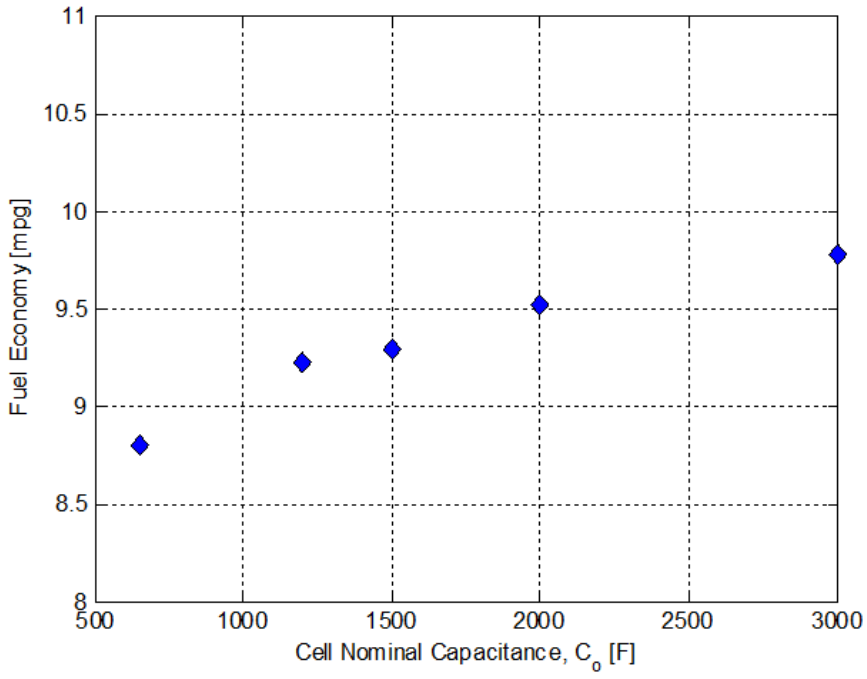


Figure 4.9: Cell-Specific Variations and Resulting Fuel Economy

Again the trend is seen where larger capacitance increases the fuel economy; the smallest capacitance cell size resulted in a 26% improvement with the largest cell achieving an improvement of 41%. Interestingly these figures are very much the same as the capacitance only variations, which means that the increasing resistance values do not seem to constrain the vehicle performance as much as the decreasing capacitance values.

4.4.5 Cell Number Variations

For a given electrical power demand, the current draw can be decreased by increasing the system voltage. The choice of 54 ultracapacitor cells in series is admittedly quite low with respect to operating voltage, since with the SOC limitations the lower and upper voltage limits are 72.9V and 145.8V, respectively. This choice is initially due to the fact that Maxwell has readily available modules that contain 18 of the 3000F cells in series, and the series combination of three of these modules is the lowest possible to achieve suitable voltage limits for the UQM motor/controller model chosen. This motor/controller is capable of handling up to 250V (still quite low for an HEV), so this study will vary the number of cells in series to see the effect of increasing the pack voltage (decreasing capacitance, increasing resistance and energy content) on the fuel economy. The values considered here range from 50 cells in series (pack size 50F,135Vmax) up to 90 cells (33.3F, 243Vmax). The resulting fuel economies for these DP simulations are shown in Figure 4.10.

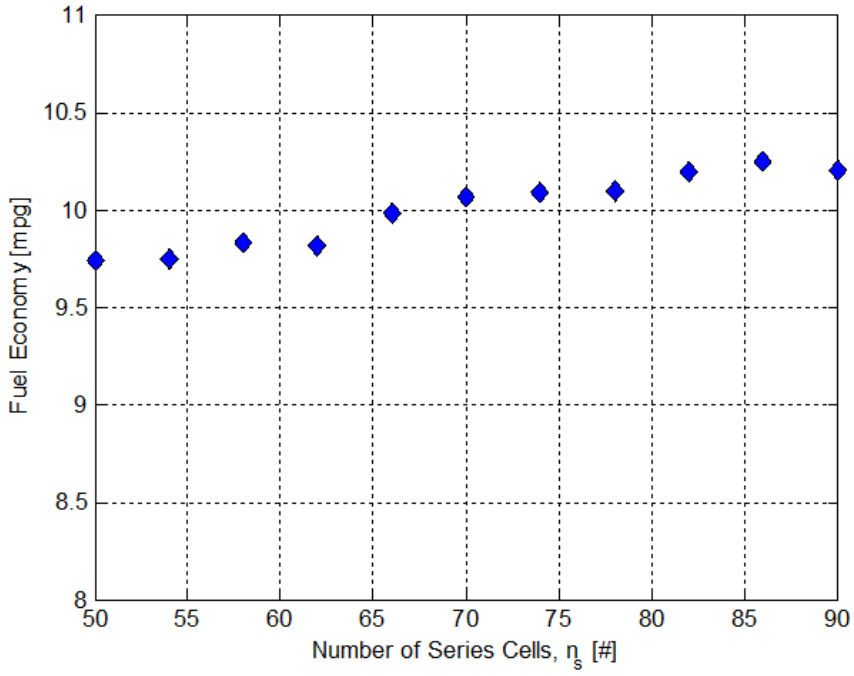


Figure 4.10: Cell Number Variations and Resulting Fuel Economy

Not surprisingly, as the number of cells increases, the fuel economy increases accordingly. This somewhat linear trend starts around 9.74mpg (40% improvement) for 50 cells in series and goes up to 10.2mpg or a 47.0% improvement at 90 cells. In a real hybrid vehicle, however, it is possible that the system voltage could be even higher to help limit current draw and resistive losses. In this case, the potential fuel economy benefits and power capabilities must be weighed against the size, weight, and cost of the system. A potential further exploration of this same trend using different cell sizes or voltage limits of larger packs might yield further insights into the best configuration of cells for the ultracapacitor system.

4.4.6 Torque Coupling Ratio Variations

Previous PSAT simulations have been conducted to see the impact that changing the engine to motor coupling ratio had on the resulting fuel economy, but since this had a forward looking controller, the results were mixed. Conducting the same sort of variations in the DP method will find the optimal solution for each potential ratio, allowing for a better comparison between solutions.

The original torque coupling ratio of 2:1 is chosen based only off the engine and motor maps and the speed ranges of each component. Since the peak power point of the motor is around 4000rpm, and the peak engine power point is around 2000rpm, the choice of coupling ratio is 2:1. In this study, a variety of ratios were investigated using the DP method, ranging from 1:1 (engine and motor speed the same) to 3:1 (motor goes three times faster than engine). The resulting fuel economies are shown below in Figure 4.11.

Some sort of trend was expected to be seen as the torque coupling ratio is increased, however the simulation results show a flat and relatively invariant trend for all ratios above 1.5. The expected trend was thought to result from the change in operating region of the motor towards the region of highest efficiency, where it might benefit the fuel economy more. Obviously, from the results it is seen that there is not a “best” solution for the torque coupling ratio which results in the best fuel economy. The complicated interaction of the motor and engine torque de-

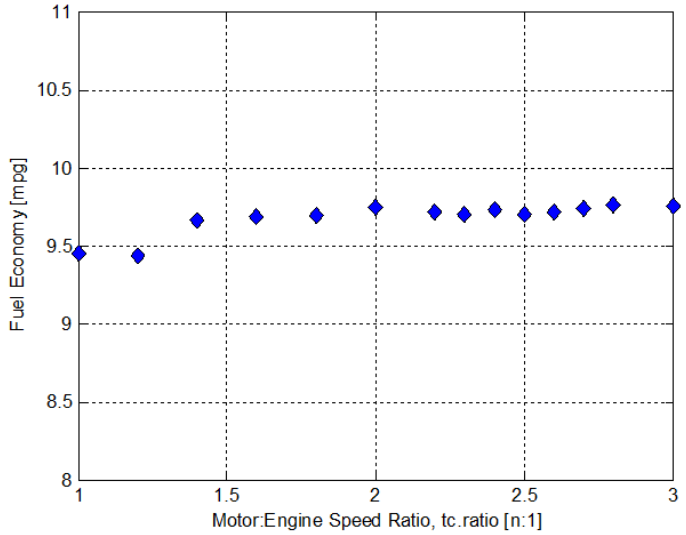


Figure 4.11: Torque Coupling Ratio Variations and Resulting Fuel Economy

mands with the Dynamic Programming method shows that the choice of torque coupling ratio is largely up to the design engineer, where it would be selected to give the appropriate low speed torque assist to aid in vehicle acceleration.

4.4.7 Vehicle Mass Variations

An important design consideration proposed in [2] is that lowering the overall vehicle weight is of significant importance if the vehicle's fuel economy is to be improved. This seems somewhat obvious, since from basic physics it is known that the power demands are in large part proportional to the vehicle's mass; inertia, rolling resistance, and grade climbing loads are all directly affected by vehicle mass, where aerodynamics are not. In this study, it is decided to explore the potential fuel economy benefits of decreasing the vehicle mass over a reasonable range, while still using the nominal ultracapacitor hybrid system to assist in meeting vehicle demands. The base vehicle weight fully loaded is around 9500kg and the unloaded weight is around 7300kg, so the range of weights considered here is from 5000kg up to 9500kg [2]. A reduction in weight of this order of mag-

nitude is reasonable when considering the advances in light weight materials, solid modeling and analysis, and vehicle design methods that have come about in the last decade.

The baseline vehicle has been simulated over the same cycle, with the only change being that of vehicle mass. The results of these simulations are shown in Figure 4.12.

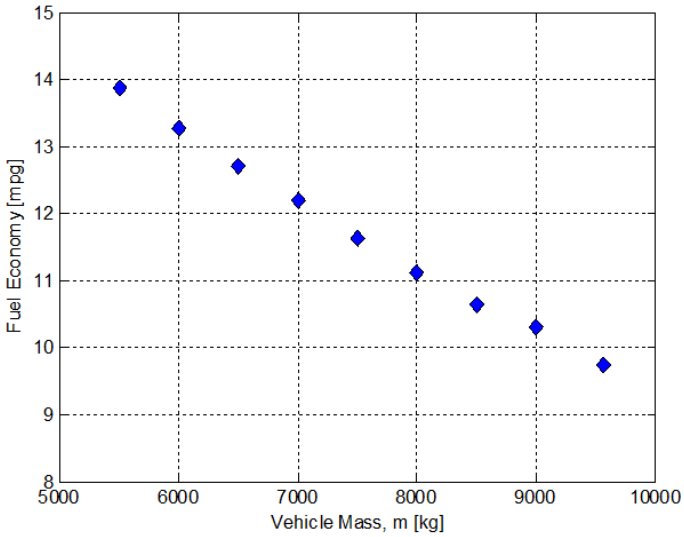


Figure 4.12: Vehicle Mass Variations and Resulting Fuel Economy

As would be expected, the decrease in vehicle mass has the effect of decreasing the vehicle loads and increasing the resulting fuel economy. From the baseline simulation of 9500kg which resulted in a fuel economy of 9.75mpg (+40%), a vehicle of nominal weight of approximately 7500kg might expect to achieve 11.6mpg and the lightest weight vehicle at 5000kg could get 14.6mpg. Percentage improvements can be calculated along with these figures, however they are not realistic to present alongside these values since they are with reference to the ICE configuration at the baseline weight and hence would exaggerate the potential benefits.

4.4.8 Summary of Results

Considering all of the results of the Dynamic Programming simulations conducted, the best solution to improve fuel economy for a heavy vehicle is to decrease the vehicle weight as much as possible, a finding which agrees with the recommendations in [2]. Varying the motor-to-engine speed ratio was thought to affect the fuel economy, but was shown to not result in any significant trends.

When considering how the ultracapacitor pack parameters are varied to benefit fuel economy, the resistance and capacitance showed the expected trends; a larger capacitance, lower resistance pack resulted in the best fuel economy. Increasing pack size through the number of cells in series also increased the potential fuel economy, with the best improvement (considering the number of cells needed) coming at around 70 cells. However, more detailed modeling could be conducted, especially in the area of motor/inverter modeling and the response to or capability to use the ultracapacitor energy storage. The modeling here did not consider the inverter separately since the appropriate modeling data is not available in PSAT. It is likely that the inverter response characteristics could have a significant effect on the potential use of the ultracapacitor pack for power boost as modeled here.

The following chapter begins the development of a Model-Predictive Control (MPC) routine. This control method is a forward-looking approach which makes optimal decisions over a finite prediction horizon and makes use of future information. In contrast to the backward-calculating DP approach, MPC has the potential to be implemented in real-time for hybrid vehicle energy management.

Chapter 5

Model Predictive Control Development

Model Predictive Control, or MPC for short, is an optimization-based receding horizon control strategy which, unlike pure dynamic programming techniques, has the potential to be implemented on a hybrid vehicle. The previous chapter solved the nonlinear fuel minimization problem over the entire cycle length by assuming known future demands using dynamic programming. In this chapter, the method of dynamic programming is still used for solving the fuel minimization problem within the receding horizon MPC framework. Initially, future demands over the prediction horizon are assumed to be known and later these demands are estimated based on current demand. Of course, the MPC solution will be suboptimal compared to that of a full horizon DP, but it is desired to explore the compromise made on fuel economy in order to reduce the computational effort and create a real-time implementable control strategy. The next section will give background information on the MPC method relevant to the work at hand.

5.1 Background

This section provides an explanation of the MPC method as it is implemented in this research, along with the relevant modeling equations. Other MPC research relevant to the area of hybrid vehicle controls will be presented and related to the current research. A discussion on the role of future information in the MPC prediction method is undertaken, as this relates to the work of this and the following chapter.

5.1.1 MPC Algorithm

The Model Predictive control method relies on the dynamic model of the heavy ultracapacitor hybrid vehicle system presented previously in Section 3.4. Once the system open-loop model, pointwise-in-time hard constraints on states and controls, and a performance index are all defined, the MPC method can be implemented [10, 32, 8].

MPC uses the system dynamic model (and could use a model for disturbances, but this is not implemented here) to predict the evolution of the states of the system as a function of control inputs over a finite future prediction horizon. A depiction of the MPC system input prediction is included in Figure 5.1, where the example input is driving torque demand.

Like the DP method, MPC calculates a sequence of optimal control inputs by minimizing a cost function over this prediction horizon. The cost function or performance index for MPC is initially identical to that from the DP method, with a difference in the limits to reflect the receding horizon approach:

$$J(i) = \int_{i}^{i+n} \left(\dot{m}_f + q\Delta_{eng,on/off} \right) dt \tag{5.1}$$

where i is the current time step, n is the length of the prediction horizon, and $\dot{m}_f$ is the engine fuel rate, q the constant engine on/off cost, and $\Delta_{eng,on/off}$ is the engine

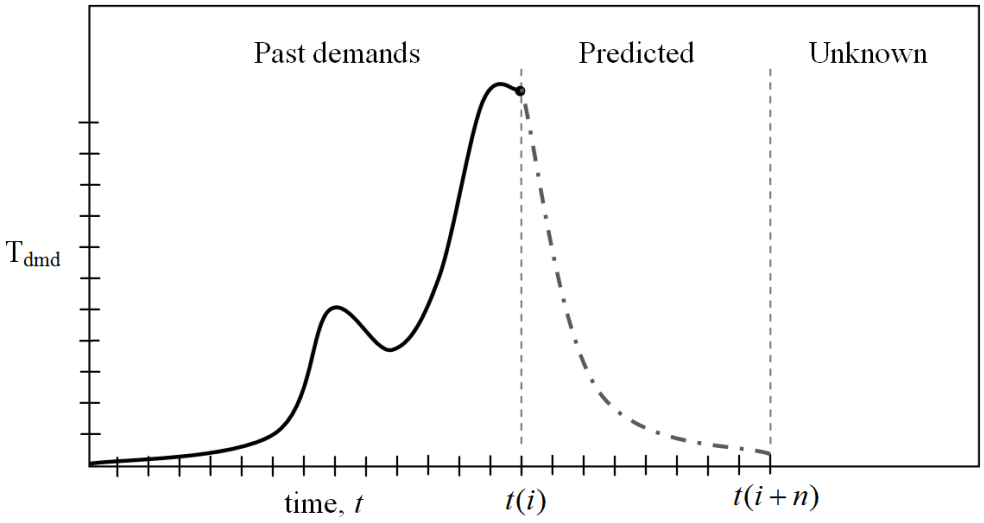


Figure 5.1: Model Predictive Control Method

on/off switch signal.

The MPC method minimizes this cost function over the prediction horizon and calculates a set of optimal control actions over a finite control horizon (of length less than or equal to the prediction horizon). The method applies only the first input of the calculated control sequence, the prediction horizon is moved forward one step and the process is repeated. The same system constraints are enforced for the MPC solution to the control problem as were covered previously for the DP method in Section 4.2.

Because of its receding horizon nature, the Model Predictive Control strategy can adapt to changing future demands, an advantage over DP. Since the typical MPC horizon length is a fraction of the whole cycle time, the nonlinear optimization problem can be solved much faster than a DP optimization solved over a whole cycle. If required for implementation in a fast process, the computational effort can be further reduced if the cost function is quadratic and the system constraints are either linear or linearized, using methods found in [8, 10].

5.1.2 Related Research

The methods of MPC have been applied to a wide range of control problems, but MPC has typically been found in industrial applications where the processes are relatively predictable [10, 43]. A brief history of the MPC method and survey of industrial MPC methods has been covered in [43]. Recently, more and more research has been focused on using MPC on faster responding systems like hybrid vehicles to achieve a variety of different purposes. Hybrid vehicle control problems to which MPC has been applied include: anti-skid traction control in [18], clutch engagement to minimize undesirable torque feedback in [45], estimation of vehicle mass and road grade to aid in vehicle cruise control in [54], and current management and control of a fuel cell and ultracapacitor power system in [5, 51]. MPC has also been used to control a parallel hybrid four-wheel drive vehicle using system identification techniques to reduce the nonlinear vehicle model into a mixed integer quadratic program for real-time implementation [21].

This work is based partly off of the work of another Clemson researcher who is applying MPC to the Toyota Prius® hybrid, where the future torque demand is predicted using a simple exponential decay model [24]. This prediction method is investigated and implemented in this research as well, starting in Section 5.3. Other prediction methods have been proposed for MPC prediction, such as the work of [34], where acceleration and braking trends are supplemented with known route and GPS data to predict the future demands in power-split hybrid transit bus. This sort of application is very well suited to the MPC method, where telemetric future information is able to augment the system response model to aid in predicting the future demands. Another example comes in [49], where very simple traffic information is applied to help control the torque split ratio of the parallel hybrid vehicle. In this case, the MPC method is implemented with a very short preview and is aimed at vehicle speed control when following a leading vehicle both with and without passing [49]. Similar telemetric-based future information prediction methods will be explored in this study as well, beginning in the following chapter.

5.1.3 The Role of Future Information

It is at this point that the nature of the future information becomes critical. Initially, building off of the DP results and modeling, full future information (a completely defined velocity-versus-time profile) will be assumed available over the prediction horizon. This is the logical first step to transition from the DP method to the receding-horizon Model Predictive Control.

Once the MPC method is vetted for functionality, a prediction horizon variation will be carried out in order to see the dependence on the length of the prediction window has on the resulting fuel economy. After a suitable horizon length has been determined, the role of the availability or lack of information about future power demands will be investigated.

In realistic implementations, it is impossible to be certain about the future driving demands over the next 30 seconds or even the next two seconds. The prediction method is a critical part of the MPC framework. A variety of prediction methods have been proposed for the solving of the future information problem, including the use of current speed/acceleration trends in [34] or the use of an exponential decay relationship for the driving torque as in [24]. The use of a simple exponential decay relationship has shown promising results and will be explored as the prediction method in this study.

Later, in the next chapter, we will investigate the use of telematic information in the form of future route, grade, traffic, or traffic signal data to construct the expected future demands on the vehicle. This area likely will be more and more applicable in the coming years as vehicle control systems become more sophisticated along with more infrastructure becoming wired and networked together.

5.2 MPC Programming and Validation

The Dynamic Programming coding has been slightly changed to become a receding horizon MPC routine by nesting the main DP loop inside of another loop, so the DP calculations are effectively run over the whole prediction horizon for each time step. At the end of this calculation, the first decision is applied and the corresponding fuel rate calculated. Over the whole cycle, the sequence of control decisions leads to a total fuel economy that should be close to but always less than that of the full horizon DP method.

In order to make sure the results of the MPC method are accurate, a comparative simulation or validation of the routine is done against the DP results. In this case, the prediction horizon is set equal to the cycle length and the DP calculations are thus only called once. This validation run resulted in the exact same fuel economy as the DP baseline result, as would be expected. This ensures that the same calculations are being performed in the MPC loop as were previously done in the DP coding so any differences observed in the resulting simulations are purely due to the differences in the control strategy methods.

Again, in the MPC method the cost function will be based on engine fuel rate with the same added engine on/off cost. This engine on/off cost functions as before, to eliminate unrealistic engine on/off switching.

5.2.1 Initial MPC Simulations and Issues

Initial simulations using the first version of the MPC routine strictly based on the DP coding resulted in rather interesting but troublesome control behavior. The tendency of the control method, especially for short prediction horizons, is to overuse the motor assistance lowering the resulting SOC trajectory down near the low SOC limit. This behavior can be seen in Figure 5.2, where the SOC profile for the DP method is juxtaposed to the results from the MPC method with three and ten second prediction horizons.

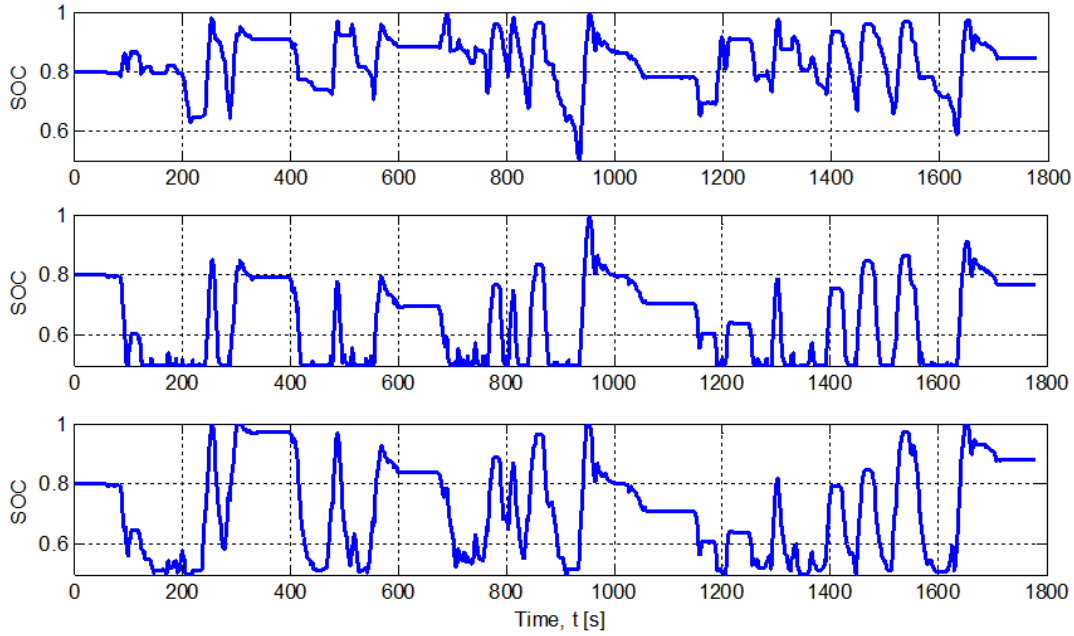


Figure 5.2: SOC Comparisons: Top DP, Middle MPC 3s Horizon, Bottom MPC 10s Horizon

As we can see in the figure, the DP method does not allow the SOC to drop as close to the lower boundary as the MPC method. Since the short-sighted MPC control drops the SOC to such a low level, it is unable to meet some of the desired torque demands over the cycle resulting in "simulation errors" which need to be addressed. Further detail of these errors are illustrated in Figure 5.3.

At these points, the controller has used up the limited energy available to assist in minimizing the fuel used over its short prediction horizon. After the first few demand above the maximum engine torque, the ultracapacitor reaches the low SOC and is prevented from assisting further. This effect leads to a torque agreement error and an erroneously high fuel economy. One can expect that the sum of instantaneously optimized behavior over a time does not yield the optimal solution for the total time. This means that the MPC method's minimization of the fuel cost over the short horizon will not achieve the same optimal policy as the same calculations carried out over the whole cycle, and thus can be much improved. Fortunately, there are methods to try and correct this short-sighted

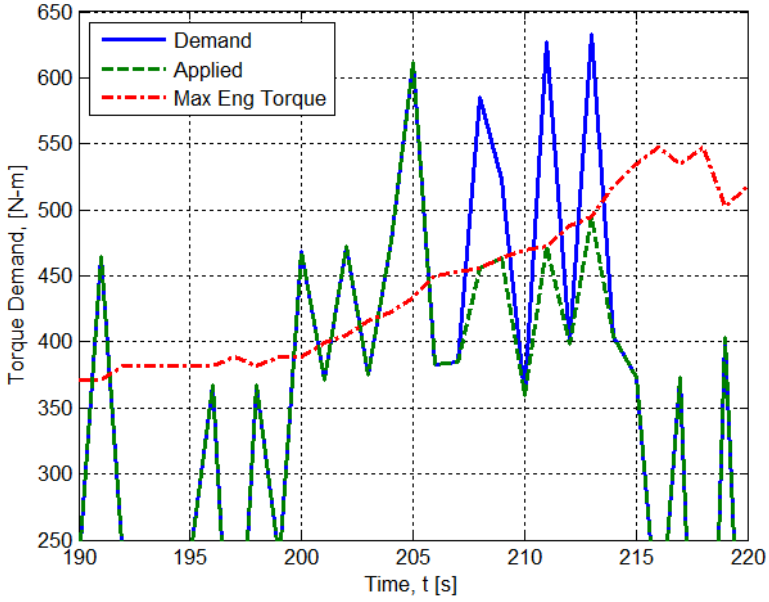


Figure 5.3: MPC Simulation Errors Due To Low SOC - 10 Second Preview

behavior, such as adding an SOC-based penalty cost into the cost function. The details and results of this method are presented next.

5.2.2 MPC With SOC Penalty

To avoid the simulation errors associated with the short-sighted nature of the MPC method, an SOC-based penalty cost has been analytically developed to maintain the vehicle SOC within a narrow range. A first attempt at avoiding these errors employed a fixed cost value for all SOC values below a certain threshold. This method proved insufficient and resulted in the SOC profile simply following the threshold boundary as it had followed the original low SOC boundary. Errors still occurred with small values of this cost, while values suitably large to prevent the errors greatly changed the resulting performance and negatively affected the fuel economy. As a result, this “hard” cost method is not deemed sufficient to constrain the SOC and eliminate the MPC’s short-sighted errors.

A “soft” SOC constraint has been created based loosely on the results of the DP method. It is thought that the SOC would best be coerced to fall within the same general region that it does when the vehicle is controlled optimally, but not to rigidly confine it to this region. The derivation of this SOC cost is presented here; Starting from the optimal demand proportions (directly from the baseline hybrid vehicle DP simulation) and curve fit shown in Figure 5.4.

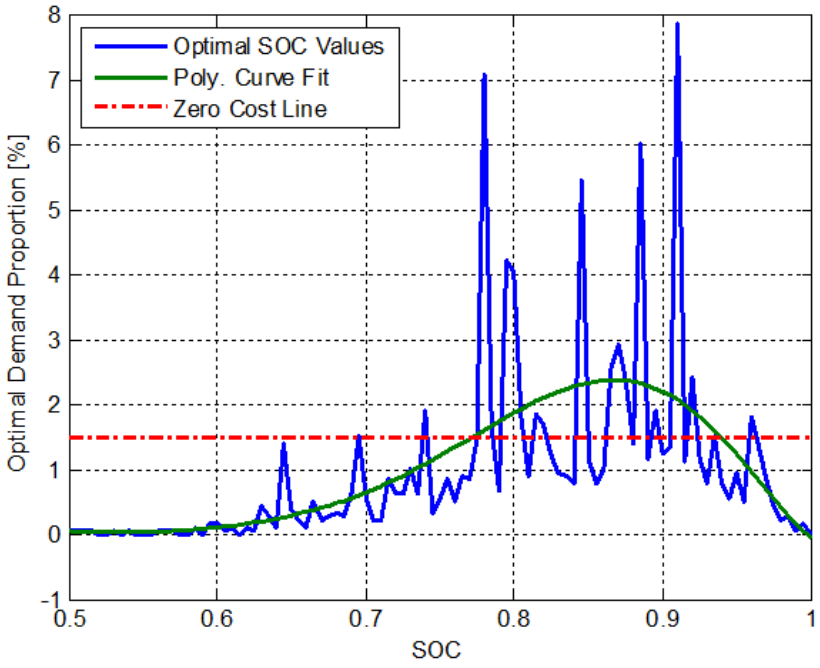


Figure 5.4: DP SOC Spectrum and Curve Fits

The optimal SOC curve fit is used to generate the SOC-based cost function that follows in Figure 5.5. For the SOC values within a particular range, which correspond to 75% of the SOC values, there is no assigned SOC cost, shown in the Figure by the zero cost cutoff line. This soft cost function is added to “convince” the MPC control method that the SOC should not vary too far from this safe region, so as to ensure the short-sighted MPC control will be able to meet the upcoming vehicle demands.

This SOC cost function is defined as a scaled function with values from zero to one. When used in the MPC routine, a SOC cost value multiplies it to give the ap-

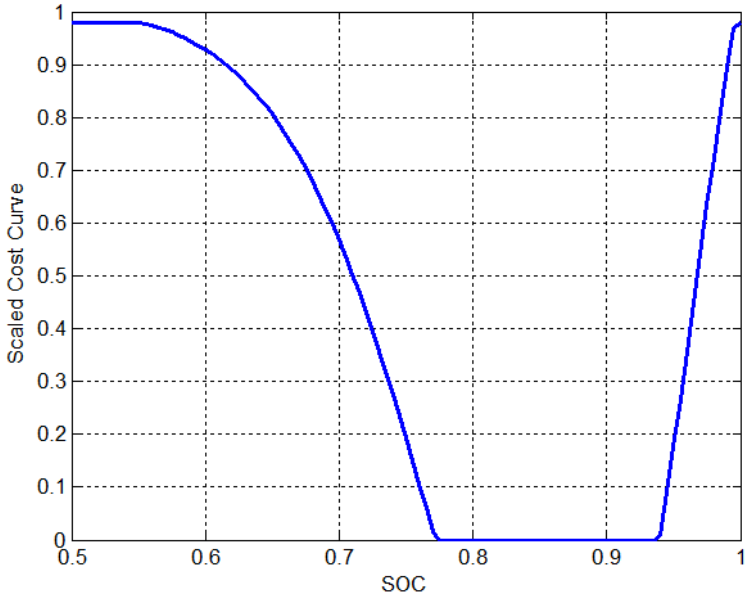


Figure 5.5: MPC SOC Cost Function-Scaled

propriate weight in the overall cost function. After testing many different values for this SOC cost multiplier on a variety of different horizon lengths, a general relationship emerged that governs the best choice of SOC cost to give the best fuel economy for each horizon length. A figure depicting this relationship for the three second horizon length is shown in Figure 5.6. With the added SOC-based soft constraint cost, the performance index for the MPC routine now is of the form:

$$J(i) = \int_{i}^{i+n} \left(\dot{m}_f + q\Delta_{eng,on/off} + h(SOC) \right) dt \tag{5.2}$$

where $h(SOC)$ is the added SOC cost as a function of the current SOC. The tuning of the SOC cost has been observed to be very sensitive; if this cost is set too low, the SOC plunges too low and results in simulation errors, if the cost is too high, the usable SOC is restricted and fuel economy is affected. At a certain point, the fuel economy peaks and then slightly declines before the torque errors associated with too low of SOC and the MPC's short-sighted control decisions appear. The best choice of SOC cost is this maximum point, which has been found for a few cases through testing and the resulting trend is extrapolated above a ten sec-

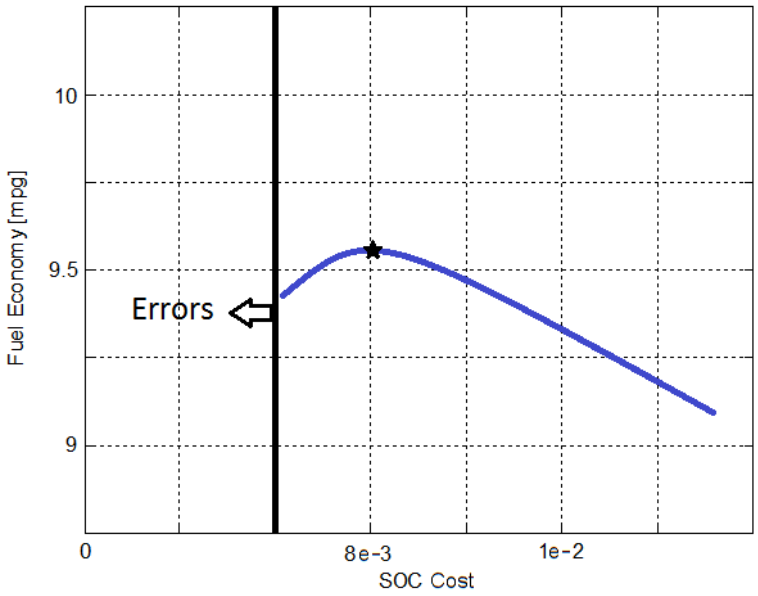


Figure 5.6: MPC SOC Cost Relationship

ond horizon. In this way, the previously observed low SOC errors are avoided for all horizon lengths considered, with longer horizon lengths requiring less SOC cost for correction. The following section details the baseline MPC results for the cycles considered.

5.2.3 MPC Results and Horizon Variation

With the SOC cost function added and appropriately tuned, the MPC routine is run using "full future" information for the CSHVR and other cycles. In this case the complete future torque demands are assumed to be known over the prediction horizon. The resulting output of an MPC simulation with a prediction horizon of 10 seconds is shown below in Figure 5.7.

The MPC routine in this case achieves a fuel economy of 9.61mpg, only 2% below the DP-calculated maximum. The demand responses look very similar to that of the DP response, with some slight differences in how the SOC changes for

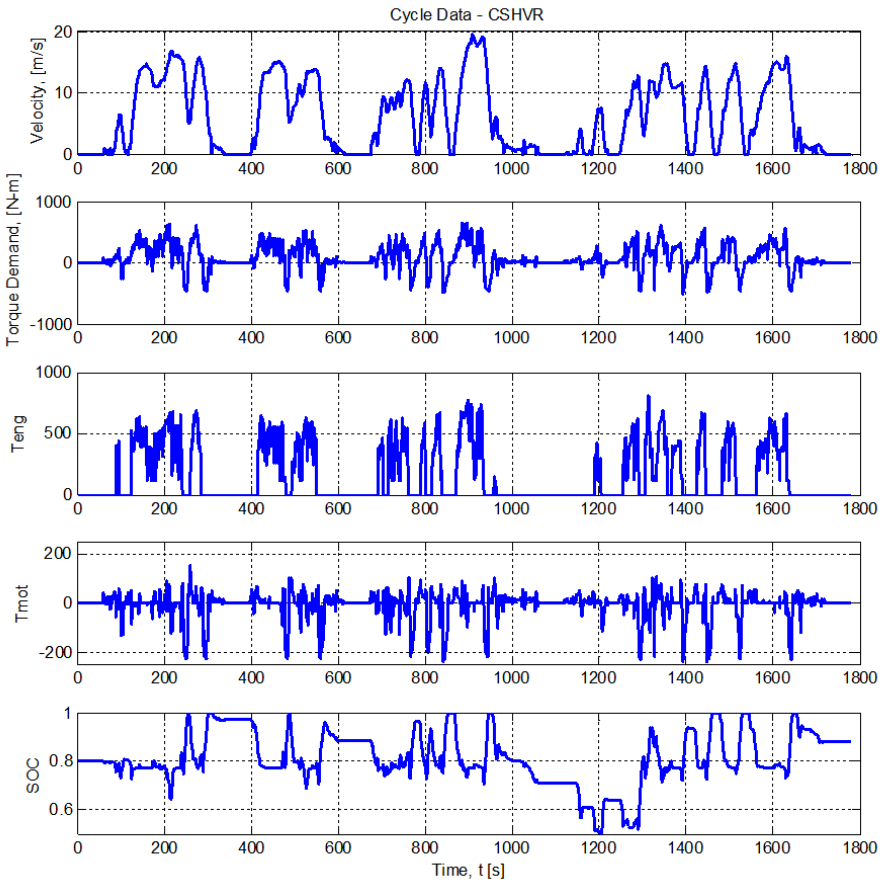


Figure 5.7: MPC Response - Ten Second Prediction

small portions of the cycle.

It is desired to investigate the effects that the length of the MPC prediction horizon has on the resulting fuel economy for the M1081 hybrid vehicle. To this end, the MPC prediction horizon is varied from 3 seconds up to 30 seconds to see the effects of this increased level of future information. As the horizon length is changed, a change in fuel economy is expected in the positive direction, as in an increase in fuel economy with increasing horizon length. The question to be answered here is: How long of a horizon length is needed to significantly benefit the fuel economy of the ultracapacitor hybrid vehicle? It is presumed that only a short horizon will be needed due to the short pulse-power nature of the ultracapacitor system, but this is to be determined. These simulations, conducted over

the CSHVR cycle using the appropriate SOC cost values for each horizon length, result in the fuel economy figures plotted below in Figure 5.8.

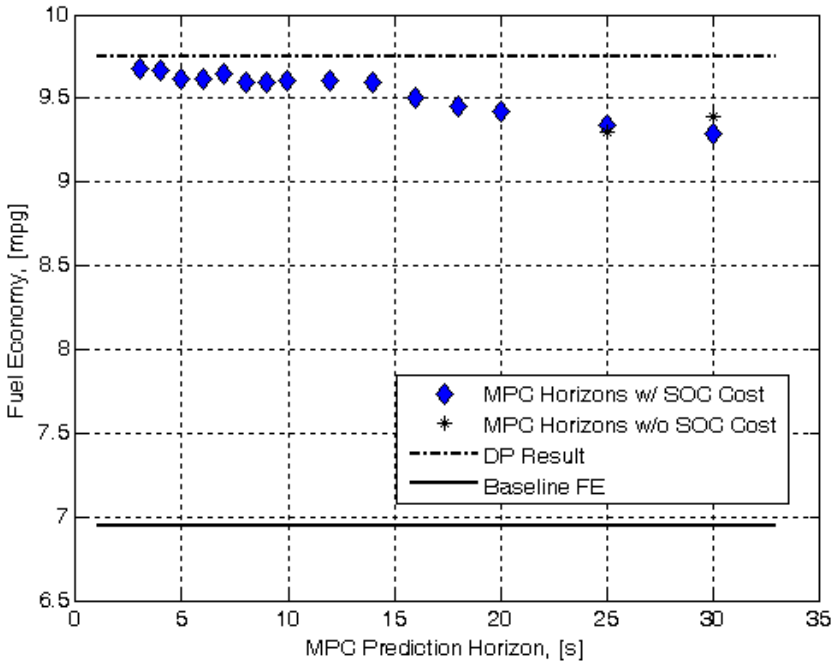


Figure 5.8: Variation of MPC Horizon and Resulting Fuel Economy, CSHVR cycle

This plot shows data points for each of the MPC horizon lengths along with the baseline fuel economy and the DP-calculated maximum fuel economy on the CSHVR cycle. As can be seen from the data, a slight downward trend is noted as the horizon length is increased. This is somewhat contrary to what was expected but shows that the limited nature of the ultracapacitor energy storage does not necessarily need the benefit of far future information to improve the fuel economy significantly. The explanation for this is two-fold: In short horizons the effect of the fuel consumption minimization is more apparent and is not confounded (or "watered down") by the added costs designed to prevent excessive engine on/off switching and use of the limited SOC. Also, the shorter prediction horizon allows the control to make better decisions (pertaining to fuel consumption) since it will not be affected by events farther in the future, which may or may not actually need to be considered to make the best decision at this instant. Therefore, a sufficiently accurate model, with realistic hard (system parameters)

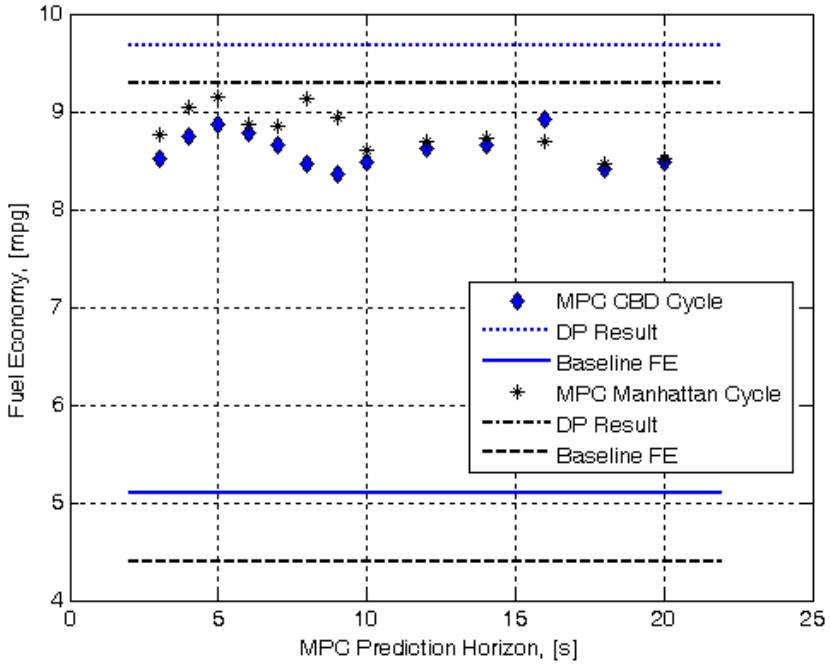


Figure 5.9: Variation of MPC Horizon Length, CBD and Manhattan cycles

and soft constraints (SOC) can realize substantial fuel economy benefit with the MPC control method and a prediction horizon of under ten seconds.

Based on experience and knowledge of the sensitivity of the fuel economy numbers from these simulations there is no significant difference in the fuel economy figures for any prediction horizons below fifteen seconds. An interesting thing can be noted for the longer prediction windows studied. For prediction horizons of 25 and 30 seconds the MPC controller is capable of traversing the cycle without the short-sighted low SOC effects, and hence can be run with an SOC cost of zero. The last two data points on Figure 5.8 show the comparative result between no SOC cost and a very small SOC cost which is the result of the extrapolation. At this point, it is seen that the SOC cost is no longer beneficial to the fuel economy above a thirty second horizon.

To further check these horizon length results, the same routine is run on the CBD Truck and Manhattan Bus cycles. The horizon lengths explored for these

cycles are three through twenty seconds, with the fuel economy results shown in Figure 5.9.

A less defined trend is seen with these cycles, where the presence of far off future information does not necessarily benefit the fuel economy. In both of these cases, a short horizon length of between four and ten seconds seems to show a good benefit to the potential fuel economy using the MPC routine with fully defined future demands. However, it must be noted that these results only stand true when the complete future information is present, which is not practical in realistic implementations.

The following section will approach the MPC problem from a slightly different perspective, that of predicting the future demand based on an exponential decay relationship. Later, telematic future information will be employed to help in this prediction and possibly aid in the minimization of fuel consumption over a given cycle.

5.3 MPC Demand Prediction

The "Predictive" part of the Model Predictive Control methodology has heretofore been assumed as unimportant, since the short range future demands have been treated as a known quantity. Since it is impossible to know for certain what future demands the hybrid vehicle will see, in practice, the demand prediction method a critical part of the MPC routine. A variety of strategies have been used in other places to predict future system demands, such as: assumed constant values (zero order holds), linear or exponential decay curves, system response-based prediction, and probabilistic or stochastic predictions. In this section, the investigation will start with the exponential decay trends and move towards future demand prediction using added telematic route information in the next chapter.

5.3.1 Discrete Exponential Decay Trending

From the work of Ali Borhan et al. [24], a simple exponential decay relationship is used in the nonlinear constrained MPC framework to predict the future torque demand in a power-split hybrid vehicle. In this work, the time constant for the torque decay and penalty weights assigned to the fuel rate are all quantized functions of the current torque demand. Higher torque demands are assumed to decay much more quickly than lower demands, and the fuel cost is penalized more on lower torque demands since these are usually the less efficient operating region of the engine [24]. The heuristic governing equation for the exponential decay is given in Equation 5.3, where k is the current time, m is the prediction time, and τ_d is the time constant for the decay rate. The time constant τ_d is related to λ by the relationship in Equation 5.4, which will be important in the next section

$$\begin{gathered} T_{dmd}(k+m) = T_{dmd}(k)e^{-m/\tau_d} \\ or \\ T_{dmd}(k+m) = T_{dmd}(k)e^{-m\lambda} \end{gathered} \tag{5.3}$$

$$\tau_d = 1/\lambda \tag{5.4}$$

The decay method used in [24] will be applied here on the M1081 heavy hybrid vehicle using similar torque decay constants, with the torque regions modified of course for the huge discrepancy in vehicle demands. An example of the results of the future torque prediction algorithm based on exponential decay is shown below in Figure 5.10.

As can be seen in the figure, the future torque demands over the MPC prediction horizon are assumed to start at the current "given" torque value and decay exponentially from there. The discrete torque regimes and associated time constants for the decay relationship are given below in Table 5.1

Once applied to the previously coded MPC routine containing the SOC costs for each horizon length, the program is re-checked for functionality. It has been noticed that, due to the nature of the exponential decay prediction curve, the engine on/off strategy used previously has to be slightly modified in order to avoid

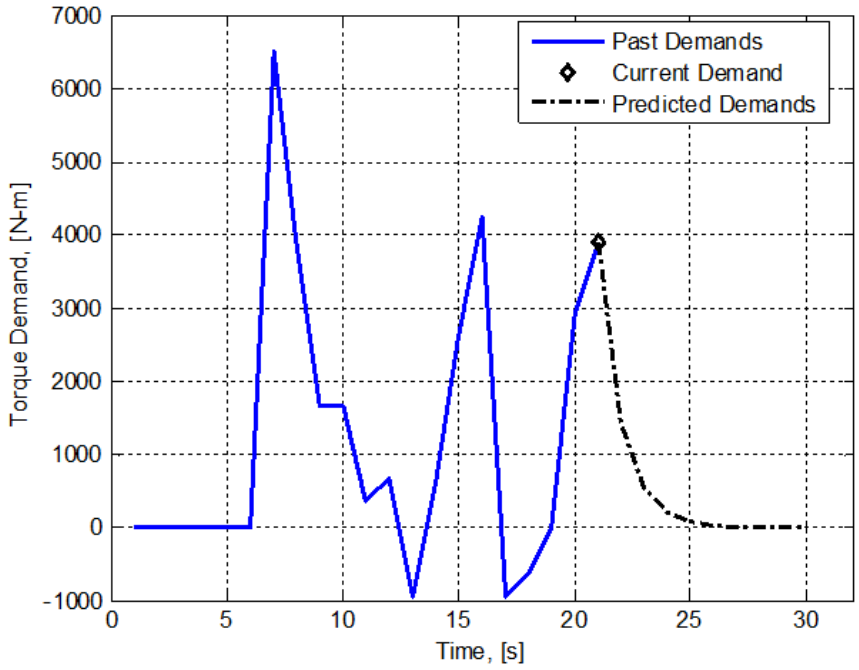


Figure 5.10: MPC Prediction Method - Exponential Torque Decay

poor controller performance. In many cases the decay prediction method would command the engine to switch off and then back on at the next time step, which is not realistic and does not follow the previously stipulated engine on/off delay times. To correct this issue, an engine off "signal" is used that delays turning the engine off until after two consecutive engine off commands are issued.

Another horizon variation study is then carried out, this time concentrating on shorter prediction lengths from three to fourteen seconds, due to the previously noted significant fuel economy benefits seen in this region. The results of this horizon variation are shown below in Figure 5.11.

The resulting fuel economy figures for the horizon lengths considered basically follow the same trend as is observed for the MPC results with "Full Future" information. However, the prediction-based results are much farther away from the DP maximum line; compared to the approximately 2% decrease from the optimal fuel consumption when using "Full Future" information, the discrete decay

Torque Demand Range	Time Constant, τ_d	Decay Rate, λ
$T_{dmd} \geq 7000$	$\tau_d = 0.1$	$\lambda = 10$
$7000 > T_{dmd} \geq 2500$	$\tau_d = 1$	$\lambda = 1$
$2500 > T_{dmd} \geq 0$	$\tau_d = 10$	$\lambda = 0.1$
$T_{dmd} \leq 0$	$\tau_d = 0.1$	$\lambda = 10$

Table 5.1: Discrete Exponential Decay Values

method shows at best a 5% decrease and at worst an 8% decrease from optimal. These results show that this particular prediction method has some room for improvement.

It is not the fault of the exponential decay method, as we will see, but rather the inadequacy of the chosen discrete regions and corresponding decay values for the prediction of future torque. Again the trend is observed where the shorter horizon approach seems to benefit the fuel economy more than longer horizons. This is likely due to the inaccuracy of the lower torque predictions, where the control is not able to make as good of decisions since it inaccurately anticipates finite future torque demands out to the end of the horizon. It is likely that this discrete exponential decay method can be improved with a change in the method for calculating the decay rates, using the methods described in the following section.

5.3.2 Decay Parameters Based on Expected Vehicle Demands

The particular parameter values used in [24] were largely determined based on experience and testing to determine suitable values and regions. A more logical and straightforward method for determining these decay parameters is presented here, based on the expected vehicle demands over the cycle. In practice, this method might be applied over the known future route, as given by GPS telematic data, or over a stochastic model-based prediction of the expected future driving demands. If we quickly recall the power spectra for the M1081 vehicle previously shown in Figures 2.5 and 2.6, a similar plot can be created for the torque demands of the cycle and in most cycles considered the overall trend is relatively

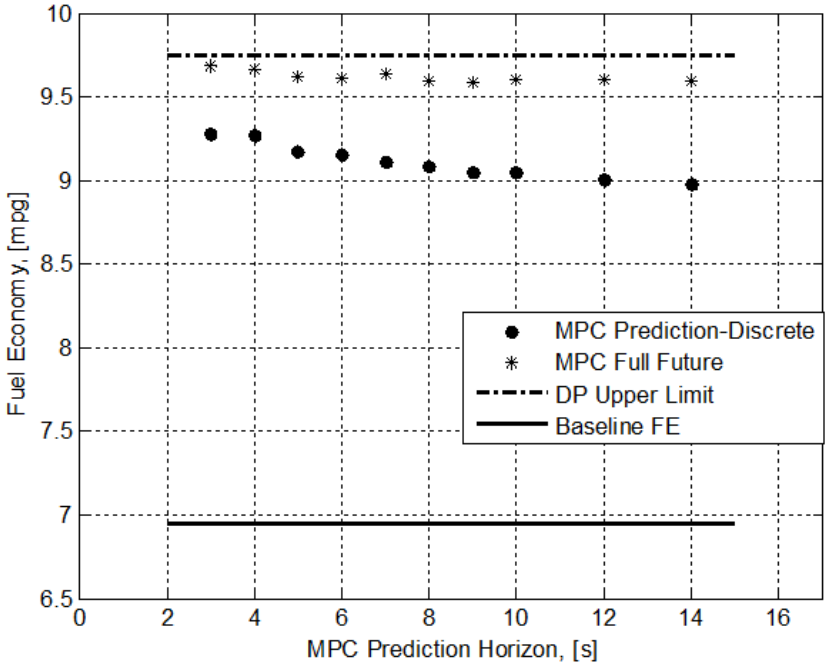


Figure 5.11: MPC Horizon Variation - Discrete Decay Values

similar.

By plotting the torque demands against their relative temporal frequency (in terms of cycle time above a certain demand level), the relationship shown in the solid line of Figure 5.12 results.

The slope of this line is essentially a piecewise estimation of the decay rate λ, or time constant τ_d, which can be numerically approximated and used to create a smoother function relating the current torque demand to the most likely decay rate. After the first order, piecewise polynomial curve fits were carried out on the expected vehicle demands for small regions resulting in approximately 20 decay rate estimates over all the expected torque values, the results can be used in the MPC routine in place of the discrete values used previously. An example of a few of these linear fits along the curve (exaggerated for plotting) are also shown in Figure 5.12.

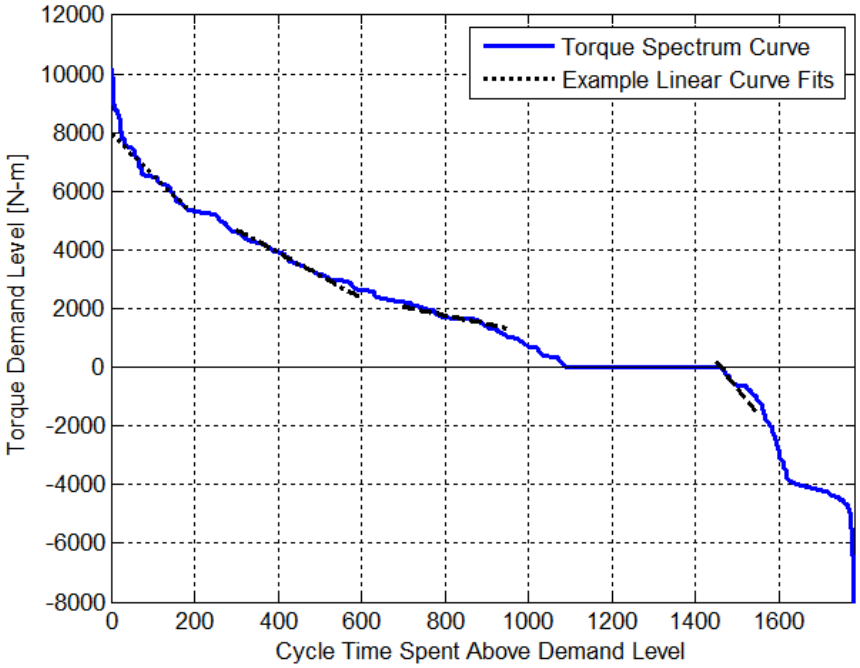


Figure 5.12: CSHVR Torque Spectrum With Piecewise Linear Curve Fits

Each of these segments corresponds to the approximate slope in a certain region, giving an associated decay rate or λ value for torques in that region. An interesting comparison is that of the decay values for the discrete method of [24] plotted alongside these demand-based values. This plot of torque value versus decay rate for the torque regions considered can be seen in Figure 5.13.

Interestingly enough, the same sort of general trend is seen between the two curves; for higher torque values (both positive and negative) the decay rate is larger, whereas for low torque values the torque demands are expected to decay more slowly. Obviously the resulting parameter values are not going to be exactly the same, since the decay rate selection in [24] is done for a much smaller vehicle. However, it is important to note that this approach is a more analytic method to determine and tune the exponential decay rate parameters of this prediction model. In practice, the "expected demand" information may not be as accurate as is assumed here, so the calculation of these decay parameters may be less accurate. However, with knowledge of the vehicle and at least a good approxima-

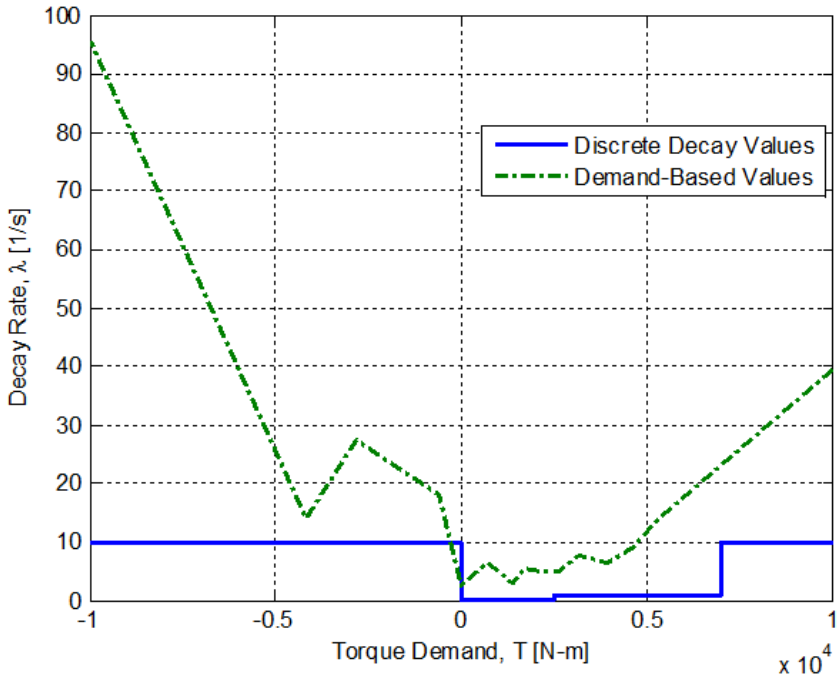


Figure 5.13: MPC Decay Rate Comparison, Discrete vs. Demand-based Values

tion of the expected loadings, similar demand torque curves can be generated and suitable decay rate information extracted.

Once the decay rate values have been established for this method, their application is again carried out as before and is run over a variety of horizon lengths. Again, only shorter horizon lengths are considered, with the prediction horizon ranging from three to fourteen seconds. The results of this horizon length study are shown below in Figure 5.14.

The previous discrete method shows a decrease in fuel economy values over the horizon lengths, but this method with its larger decay rates shows a flat trend across all horizon lengths. In this figure it is a bit difficult to tell exactly how close the values are, since they are all so close together, but it is evident that the larger decay rates (and shorter time constants) of the demand-based decay method give fuel economy values that are around 4% below the DP maximum or 36% above the baseline ICE-only case.

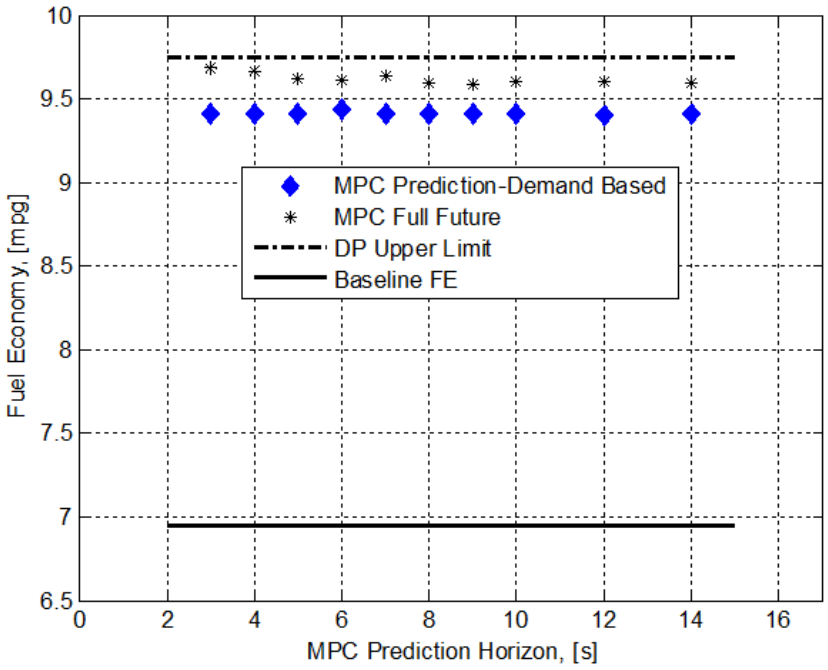


Figure 5.14: MPC Horizon Variation - Demand-based Decay Values

For easier comparison, Figure 5.15 shows a closer look at the MPC results over the horizon lengths considered. Again the decrease in fuel economy with increasing horizon length can be seen for the discrete torque decay prediction method, while the more robust demand-based method shows no change as horizon length is increased. Since the demand-based results show such a small window for improvement, it will likely be difficult to improve upon this method even with the use of future information.

5.4 Results and Conclusions

This chapter has transitioned from the method of Dynamic Programming, which requires fully present future information, to a receding-horizon, real-time implementable Model Predictive Control-based strategy for minimizing the fuel consumption of the heavy ultracapacitor hybrid vehicle. The results of a variety of

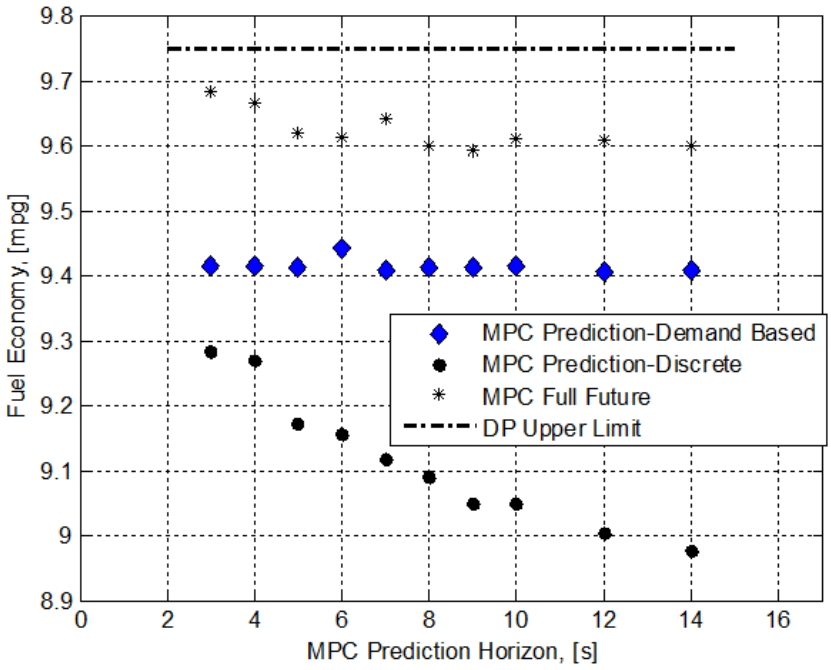


Figure 5.15: MPC Horizon Variation - Comparison of Decay Methods

MPC simulations show that a substantial fuel economy benefit can be obtained from even a relatively short five to ten second prediction horizon. These attainable benefits are just a few percent short of the DP-calculated maximum fuel economy, even when using a prediction method that does not incorporate future information.

The MPC prediction methods explored here include exponential torque decay trends using discrete decay rates and a more continuous method for determining these decay rates from expected vehicle demands. The results have shown that the exponential decay method is indeed an effective method of predicting future torque demands in the MPC framework. The decreasing trend in fuel economy observed in the discrete prediction case comes from a few different factors: The prediction method assumes a slow decay rate for lower torque values, leading to substantial residual predicted torque values that affect controller performance. The original fixed values for decay rates were supplanted by vehicle demand-based values, which provided a more realistic representation of the actual decay

behavior for this vehicle and cycle.

The demand-based exponential decay prediction method results in consistently good fuel economy values over all horizon lengths, indicating the robustness of this prediction model. The results shown confirm the fact that the MPC method is able to achieve very nearly optimal fuel consumption with a short prediction window and sufficiently robust prediction model.

The next chapter explores the use of telematic future information and the potential to improve the fuel economy of the hybrid vehicle further, by more effectively making control decisions when considering future driving conditions.

Chapter 6

MPC With Telematic Future Information

The results from the application of the demand-based exponential decay relationships were promising, leading to a fuel economy value of around 4% less than the DP calculated maximum for all short horizon lengths. With such a narrow potential room for improvement, it may be difficult to improve the prediction method further. However, it is desired to investigate the potential benefits of applying new technological innovations to possibly improve these results further. New technologies such as cell-based GPS, widely available internet connectivity over cellular service networks, and localized telematic systems which broadcast road condition information all play their part in giving the driver (and vehicle control system) of the near future the capability for better control over a hybrid vehicle's fuel consumption.

6.1 What are Vehicle Telematics?

Telematics is a word used to describe the combination of telecommunications and informatics, the study of information processing [55]. This work is con-

cerned with vehicle telematics, which applies telecommunications and computing tools in the vehicle platform to achieve a variety of goals. The commercial uses of vehicle telematics have been around since the early days of global positioning systems (GPS) and originally found wide commercial application in areas like trucking and shipment tracking [52]. In the last decade, the number of electronic components in vehicles has been increasing rapidly, with GPS navigation systems now a common component of many vehicles. A familiar widespread application of telematic technology is General Motors' OnStar^{TM} system, which provides services like emergency response and vehicle diagnostics in addition to the vehicle's GPS capabilities [52].

6.1.1 Recent Advances and Applications

Many different market segments are quickly converging on the vehicle platform and seeking to integrate telematic technologies to deliver services and conveniences to users. GPS navigation companies have quickly learned that drivers are interested in all sorts of services: from finding local restaurants, shopping locations, and movie times, to helping them become better drivers through a "driver challenge" aimed at improving fuel efficiency [22]. Modern GPS receivers can use three dimensional map data to help drivers visualize their location or plan the most efficient driving route, give real-time (and more recently predictive) information about traffic conditions, and also alert the driver if they happen to be driving over the speed limit [22].

In their commercial trucks, Freightliner now offers predictive cruise control which looks at the upcoming road grade and adjusts the vehicle cruising speed to improve fuel consumption [46]. A fellow Clemson researcher is investigating the potential fuel economy benefits from the use of future road grade information in a battery-based hybrid vehicle [56]. Some passenger vehicles are starting to use new on-board sensors for things like adaptive cruise control, collision avoidance, driver performance and attentiveness monitoring, and for vehicle security purposes [36]. Many of these new technologies have very exciting implications for vehicle safety and convenience, but without the additional degree of freedom

available in a hybrid vehicle, there are only small fuel economy improvements attainable.

6.1.2 Vehicle Control Using Telematic Future Information

Only a small amount of research has focused on the use of future information to improve fuel economy in a hybrid electric vehicle. The research in [36] focused on the development of a driving behavior model to help predict future inputs based on a nonlinear driver model, recent driving statistics, and historical traffic information. Another application of future information is that of [44], where vehicle velocity is predicted with the help of future road grade information in an ECMS online optimization routine over a relatively long (hundreds of seconds) horizon length. Optimal control methods have been applied on PHEV control simulations which use a simple driving model based on acceleration and deceleration trends from current and past traffic information to help predict future demands [42]. However, the focus of these research efforts were not on the types and character of the future information, but rather the simple application of this information to predict demands.

6.2 Future Information Generation

The City Suburban Heavy Vehicle Route, or CSHVR, is a heavy vehicle cycle originally developed for emissions studies on heavy vehicles in [41]. Previously in this thesis, it has been shown that the CSHVR cycle is of suitable character for the M1081 vehicle's mission. Here, future information will be generated for this cycle which will be used to aid in the prediction of future cycle demands. The sorts of information created will be representative of information that is either currently available through commercial sources or from local government traffic authorities. The following types of information will be considered:

- A. Speed limit information. This is already available in many places through

commercial GPS receivers [22]. This information will become more widely available in the future as mapping companies accumulate this type of route metadata.

- B. Traffic information. Currently only provided as a general "level of congestion" in many commercial GPS units, this could easily be made available as an average speed of traffic through an area [22].

- C. Traffic signal information, in this case implemented as known "stop" and "go" locations. This information is sparsely available, depending highly on the local municipality and their ability to provide this information in real-time. GPS-based maps of stop sign locations would also be useful for obtaining this information, especially in suburban areas. Onboard a vehicle, very short horizon ($<$3 seconds) stop and go information could be obtained through radar sensors of the kind employed for vehicle collision avoidance systems [36].

Notice that in this case future elevation information is not considered. Since the CSHVR cycle does not have an associated grade profile paired with it, which would drastically change the demands on a vehicle of this size, this information is not realistic to consider at this time. If driving cycles were available that had realistic velocity and grade information for this vehicle, this could be considered as an area of further investigation, but for now this is left for future work.

6.2.1 Future Information on CSHVR Cycle

The CSHVR cycle speed profile is analyzed and artificial telematic future information generated using general knowledge about urban driving conditions. The following Figures 6.1 through 6.4 depict this artificial information for each of the three types.

Figure 6.1 depicts the vehicle speed profile for the CSHVR cycle along with the generated speed limit information. The suburban cycle speed limits given are

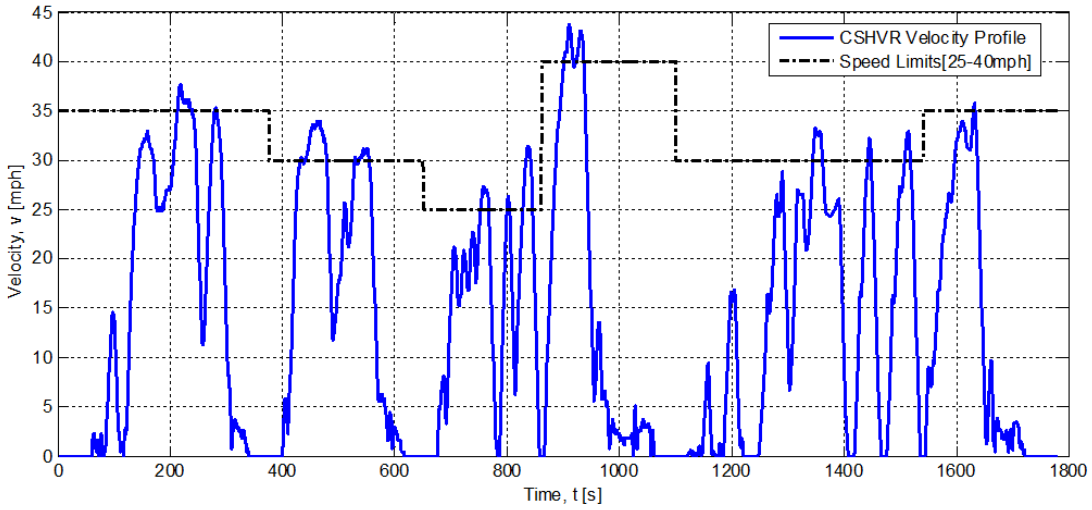


Figure 6.1: Speed Limits over CSHVR Cycle

appropriate for suburban driving and range from 25mph up to 40mph. Typical suburban driving does not usually allow for the very strict following of the speed limit, so as would be expected, the velocity trace does not always follow the speed limit. These speed limits are important to the determination of expected vehicle speed using the "general" traffic information shown in Figure 6.2.

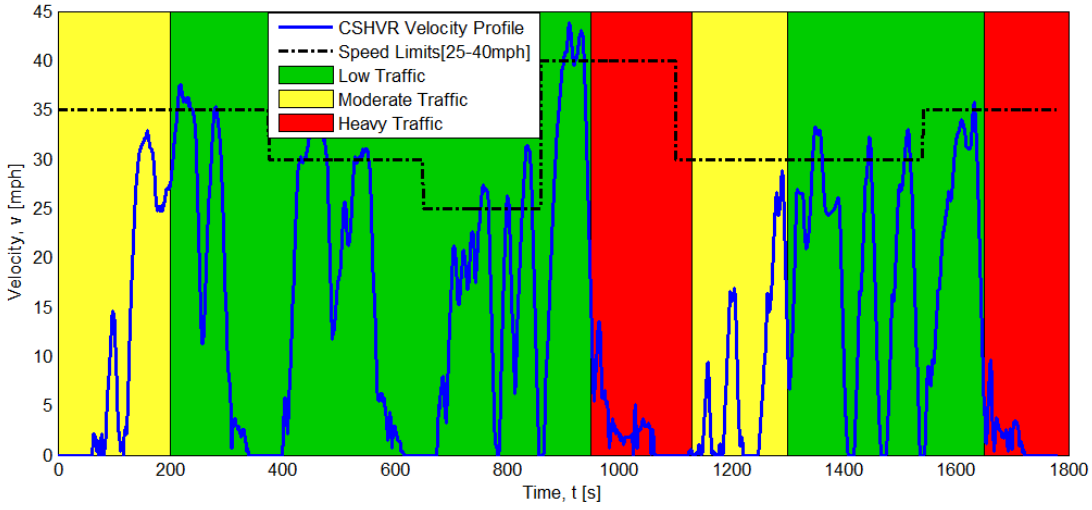


Figure 6.2: Traffic Flow Information over CSHVR Cycle

This “general” traffic information is given as a level of traffic congestion, similar to what is given by modern GPS units [22]. In this case it is assumed that low traffic (green) means traffic is flowing normally around the speed limit, moderate traffic (yellow) means traffic is slightly congested and the expected velocity is two-thirds of the speed limit, and heavy traffic (red) means the expected speed is one-third of the speed limit.

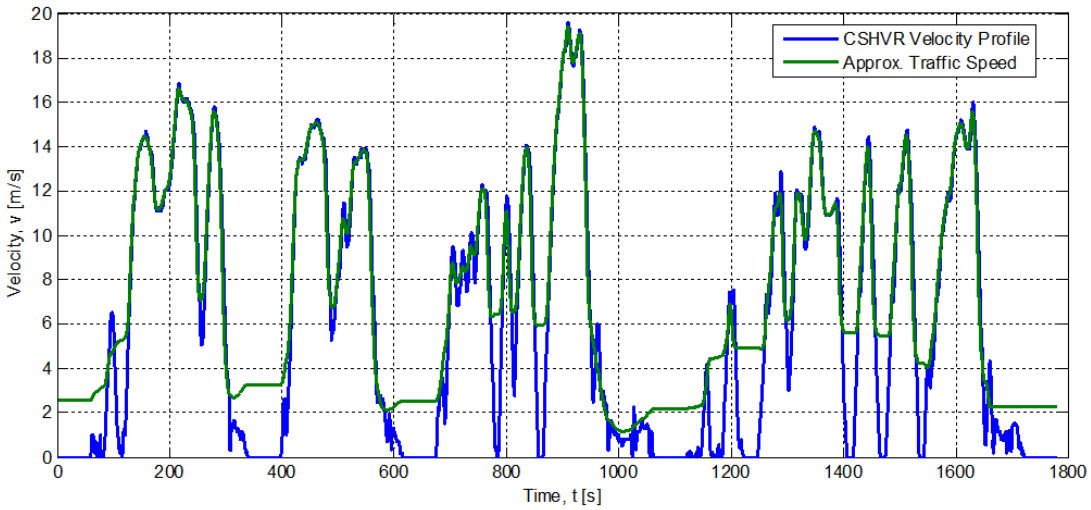


Figure 6.3: Traffic Speed Information over CSHVR Cycle

It is realistically possible to get approximate current traffic speeds from modern GPS units as well, since this data is used by the service providers to generate the general traffic information previously mentioned. Hence, in this study a secondary type of future traffic information is that of “approximate” traffic speeds at points along the cycle. This information, shown as a curve fit over the CSHVR cycle in Figure 6.3, is generated by smoothing of the velocity versus distance curve. In this manner, an approximate vehicle speed is obtained at each location along the route, but the traffic speed does not necessarily represent the individual vehicle speed due to stoppage events like traffic signals.

Stop and go information developed for the cycle is shown in Figure 6.4. These signals are based on the assumption that as the vehicle comes to a stop or accelerates there is an associated signal (from a traffic light, known stop sign, or otherwise) that is available for approximately 10 seconds or before and after the

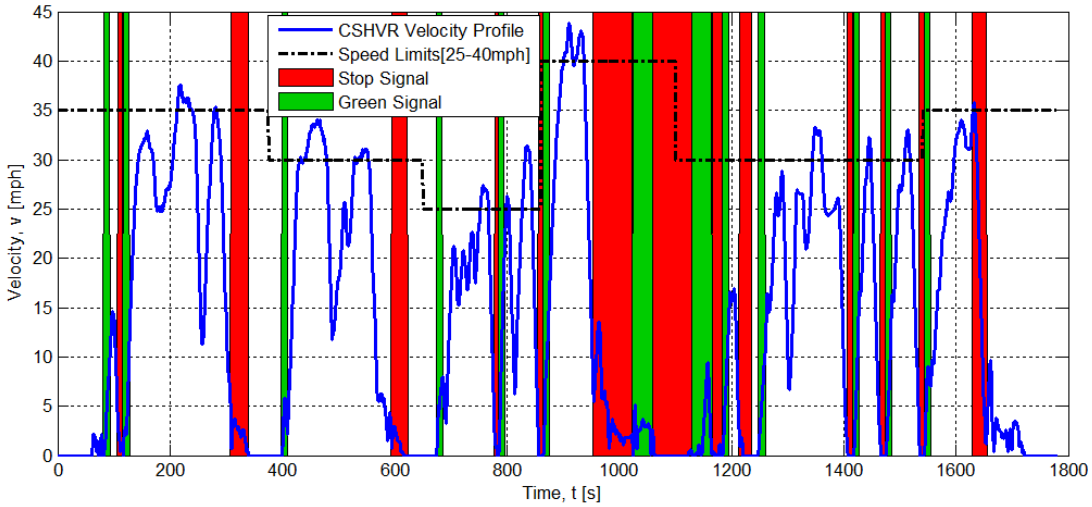


Figure 6.4: Signal Information over CSHVR Cycle

stop. Certain portions of the suburban cycle are assumed to be spent "waiting" at a light in heavy traffic, such as the time between 1000 and 1200 seconds.

With these types of representative telematic future information, the next task is to establish rules for how to use it effectively in the MPC prediction method, which is covered in the next section.

6.3 Future Information Use in MPC Prediction

The ultimate goal for the prediction method in this MPC framework is the calculation of a torque versus time curve over the prediction horizon. The challenge presenting itself with the use of telematic future information is that this information usually relates to the future velocity curve of the vehicle. This velocity information must be translated into an expected torque demand using previously described vehicle modeling techniques and known gear ratios. An additional challenge not incorporated here is the inclusion of shifting strategy predictions into this future prediction to assist in modeling future demands.

6.3.1 Velocity/Torque Decay Models

Since future information is generally sparse, over the cycle there is not always enough information to generate the expected demands using only the provided future information. Thus, the exponential decay model presented previously has been employed to fill in these gaps in two different ways:

- Torque Decay: Since the method of demand-based engine torque decay has proved to be a suitable prediction model, it is again employed here in the same manner as before.
- Velocity Decay: Since most future information gives knowledge of future vehicle speeds, a simple exponential decay curve is also applied to determine the future velocity demands. This method then uses the velocity curve to determine driving torque demands and will be used to compare to the torque decay method mentioned before.

Each of these decay models has been used on each type or combination of types of future information which are explained next. These sections will also cover the appropriate prediction rules which have been considered for each of the five cases.

6.3.2 I - Speed Limit Information

In this case, the future speed limit along with past driving information is the only information available for future prediction. The rules implemented for the use of this information are as follows:

- If the vehicle is accelerating for the last four seconds, the vehicle speed is assumed to increase towards the speed limit with a time constant proportional to the acceleration value. The acceleration curve is in an exponential

fashion, of the form "$v_{des}\left(1-e^{-t/\tau}\right)$", so the predicted speed will slowly approach the desired speed similar to a first order system response. (where v_{des} is the desired velocity, e is the natural number, t is time, and τ is the time constant)

- For the velocity decay case, if the vehicle speed is above the speed limit it is assumed to be decaying back to the speed limit. If the vehicle is not accelerating, the speed is assumed to decay to zero with a time constant proportional to the deceleration value. The decay curve is also exponential, decaying with a typical $v_o e^{-t/\tau}$ first order response. (where v_o is the initial velocity)

- For the torque decay case, all of the time the vehicle is not accelerating the engine torque demands are assumed to decay to zero in an exponential fashion. The time constant of this decay is proportional to the assumed vehicle demands as presented previously.

A set of MPC simulations has been run over the horizon lengths from three to fourteen seconds, using both the velocity decay and the torque decay prediction models. The results of these simulations are shown below in Figure 6.5 where they are compared to the DP results, the MPC full future results, and the MPC exponential decay method without future information.

As is evident from the figure, the torque decay method results in a higher fuel economy for all the horizon lengths. The velocity decay method in general does not seem to predict the future demands as well, resulting in poorer fuel economy, a trend which will be further substantiated in later results. When comparing the results of the MPC with demand-based torque decay (dash-dot line) with the speed limit future information with torque decay, the speed limit information does generally seem to benefit fuel economy just slightly at almost all horizon lengths. When it is considered that the fuel economy values for the demand-based decay are approximately 4% less than the DP-calculated maximum, the benefit of the speed limit future information is at most a half percent increase.

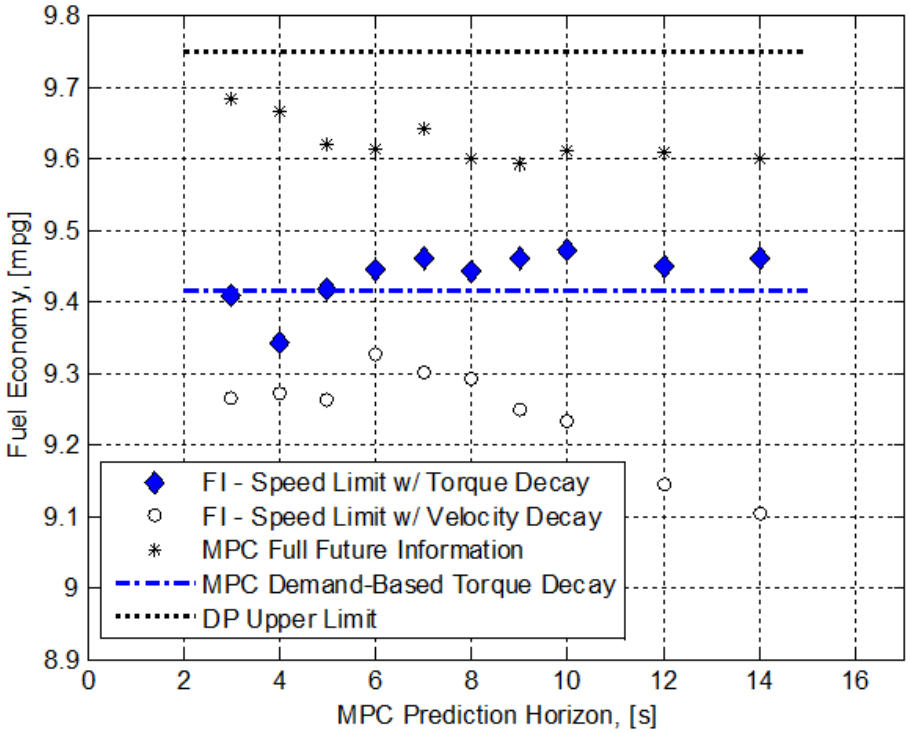


Figure 6.5: MPC Results with Speed Limit Future Information

The curious result showing the slight decrease of fuel economy at a horizon length of four seconds is the result of a few different interactions. At this horizon length, the predictive control using future information needed a slightly higher SOC cost than used previously in order to meet the torque demands of the cycle at one particular point, which is partly to blame for the lowering of the fuel economy. The other effect which resulted in this behavior comes from the chosen prediction rules and how the prediction method switches between them as the cycle is run. This odd trend appeared for all types of future information considered with the torque decay. Many different methods were implemented in an attempt to circumvent this issue, but none of these resulted in better performance than the results presented here.

6.3.3 II - Speed Limit with General Traffic Information

Again the speed limit will be made use of, but in this case it will be modified by the used of "general" traffic information available along the future route. This case is very similar to the first and has essentially the same rules implemented to predict the future demands. The difference here is that the traffic information is used to modify the speed limit as stated before: Low traffic means traffic is flowing normally with the expected vehicle speed being around the speed limit; Moderate traffic means slight congestion with the expected velocity set at two-thirds of the speed limit; In heavy traffic the expected speed is one-third of the speed limit.

The same horizon lengths are considered, along with the torque and velocity decay methods mentioned previously. Figure 6.6 shows the results of these simulations plotted alongside the other comparative curves and data sets.

Again it is seen that the torque decay method gives higher fuel economy compared with the velocity decay method, although this time the difference is clearly much smaller. The same simulation rules for the velocity decay method prove to be very robust when the expected speed information is improved with traffic information. Still, in this case, the fuel economy performance over all horizon lengths is on par with the demand-based torque decay.

6.3.4 III - Approximate Traffic Speed Information

Since it would be possible to obtain approximate traffic speed information along the route, it is desired to see the benefit this sort of information might provide to the MPC prediction method and the resulting fuel economy. Again the same rules have been implemented as with the speed limit information before, where in this case the expected speed is that of the traffic.

Simulations run over the horizon lengths from three to fourteen seconds using both the velocity decay and the torque decay prediction models resulted in the

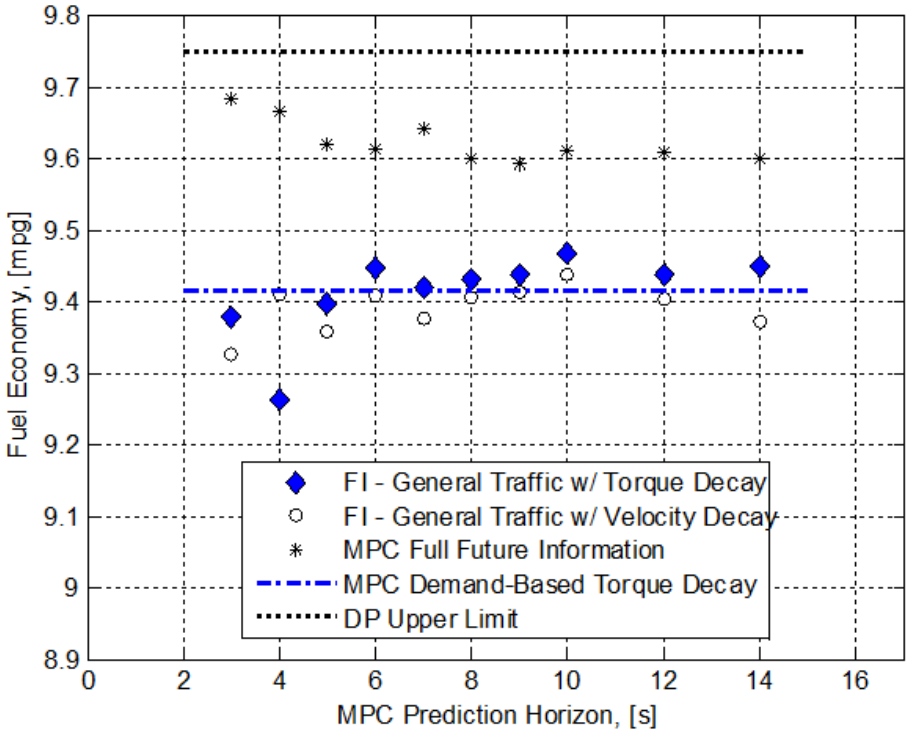


Figure 6.6: MPC Results with Speed Limit and General Traffic Information

fuel economy figures shown in Figure 6.7.

The torque decay method again is improved just slightly by the addition of the future information, for an average increase of a quarter of a percent. For some horizon lengths the velocity decay method performs almost as well as the torque decay method, but overall the velocity decay performance is about a half percent less fuel economy. Now that a few different types of future information have been looked at, it is interesting to note the trend that the improvement obtained by the added future information seems to be slightly more pronounced for the longer horizon lengths. This trend will be further illustrated later, after the last two types of information are presented.

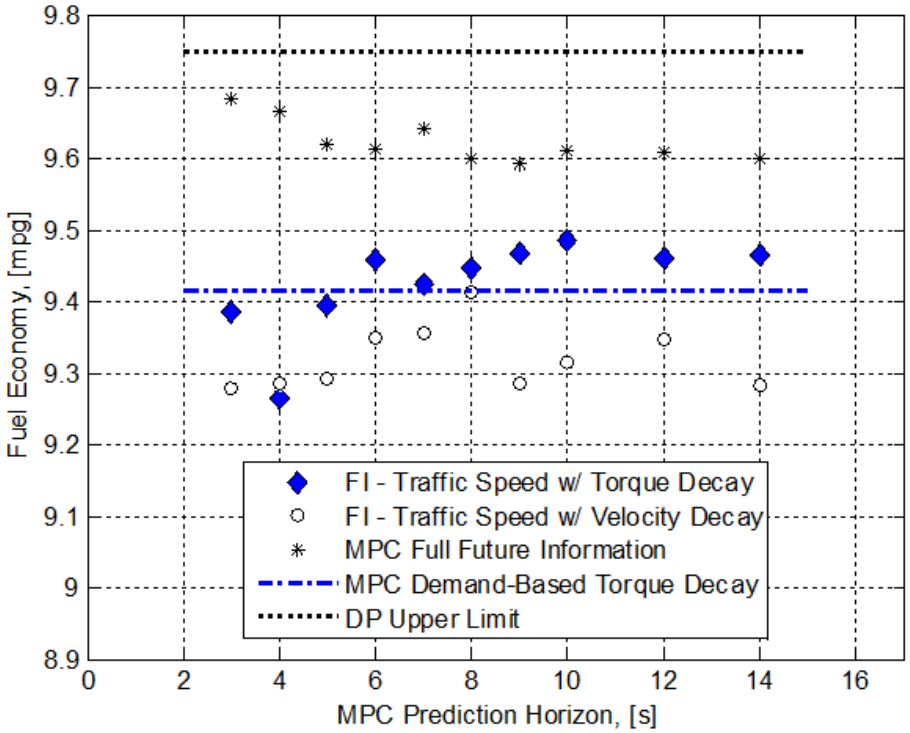


Figure 6.7: MPC Results with Traffic Speed Future Information

6.3.5 IV - Speed Limit with Stop Signal Information

Here it is assumed that the speed limit is known along with traffic signals which serve to "warn" of an approaching stop or increase in speed. The rules developed for this future information are slightly different than before:

- If the vehicle is accelerating for the last four seconds or a future "go" signal is present, the vehicle speed is assumed to increase its speed towards the speed limit with a time constant proportional to acceleration.
- If the vehicle is decelerating and the future "stop" signal is present, the speed is assumed to decay to zero.
- The times when neither one of these conditions are present are spent in either velocity decay or torque decay mode.

Again the simulations are run over the horizon lengths using both the velocity decay and the torque decay prediction models. The results of these simulations are shown below in Figure 6.8 where they are compared to the DP results, the MPC full future results, and the MPC exponential decay method without future information.

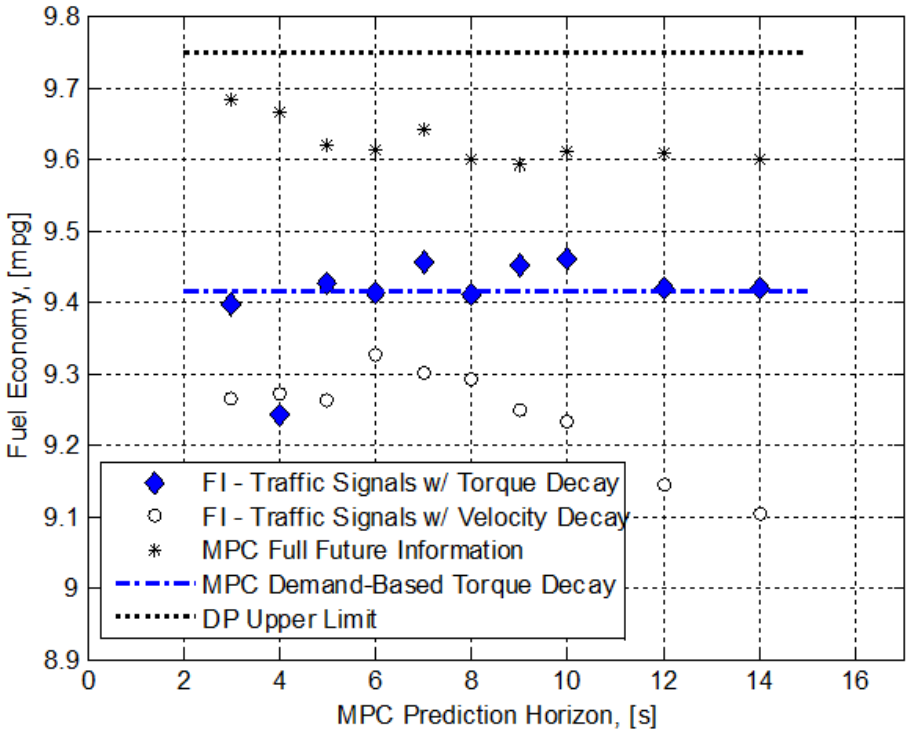


Figure 6.8: MPC Results with Speed Limit and Traffic Signal Future Information

As was seen with the speed limit information alone in Case I, the velocity decay method does not perform as well as the torque decay method when considering the fuel economy results. The addition of stop and go signal information to the torque decay method slightly improves the results for a few of the horizon lengths considered, but not by more than a half percent.

6.3.6 V - Traffic Speed with Stop Signal Information

The final case considered uses the most future information, which includes the future traffic speeds in addition to stop and go signal information. In this case, the prediction rules are a combination of the rules previously stated and are essentially the same as those used in the speed limit with stop and go Case IV. MPC simulations with this future information gave the results shown in Figure 6.9.

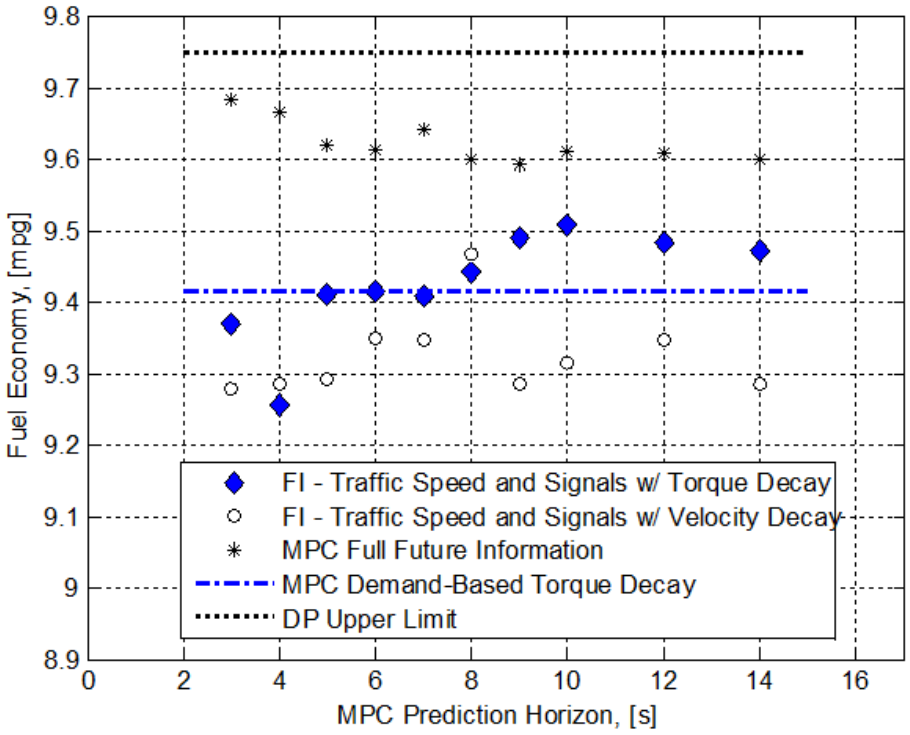


Figure 6.9: MPC Results with Traffic Speed and Signal Future Information

As is evident from the figure, the torque decay method results in a higher fuel economy for almost all horizon lengths. Not surprisingly, the future information benefits the longer prediction horizons more than the short ones and gives a maximum improvement of almost 1% over the demand-based exponential decay method. This is significantly better considering the total potential improvement is just under 4%, and the highest expected improvement is shown by the "Full future" data point for that horizon length, so the reasonable margin for improve-

ment over the previous method is around 2% total.

6.4 Summary of Results

The different types of future information implemented here to improve the prediction method in the MPC framework are representative of realistic information available through currently available telematic technology. The demand-based exponential torque decay method proved to be a robust method for predicting future demands which showed further improvement when augmented with future information.

The future information considered here has shown a larger benefit in fuel economy for the longer horizon lengths considered, which was somewhat expected. It makes sense that knowledge of farther off future demands allows the MPC prediction model to make better predictions and hence traverse the cycle closer to the optimal fuel economy. To further illustrate this for the longer horizons, Figure 6.10 shows the results of all the torque decay-based MPC simulations using the various types of future information.

As can be seen from the figure, there is a very small range of possible improvement above the demand-based torque decay method of the previous chapter. This torque decay method, once properly tuned to the expected cycle demands, showed the capability to obtain fuel economy values approximately 4% lower than the DP calculated maximum. The MPC simulations using these same horizon lengths using "Full Future" information performed around 2% lower than the DP maximum. Between these data points there is only a total of around 2% total improvement in fuel economy that could be obtained from the addition of future information or other improvement to the prediction strategy.

The different types of future information assisted the prediction method more or less, as can be seen in Figure 6.10. As would be expected, relatively general future information like stop signals and general traffic data does not allow for the improvement of fuel economy quite as much as the more accurate traffic speed

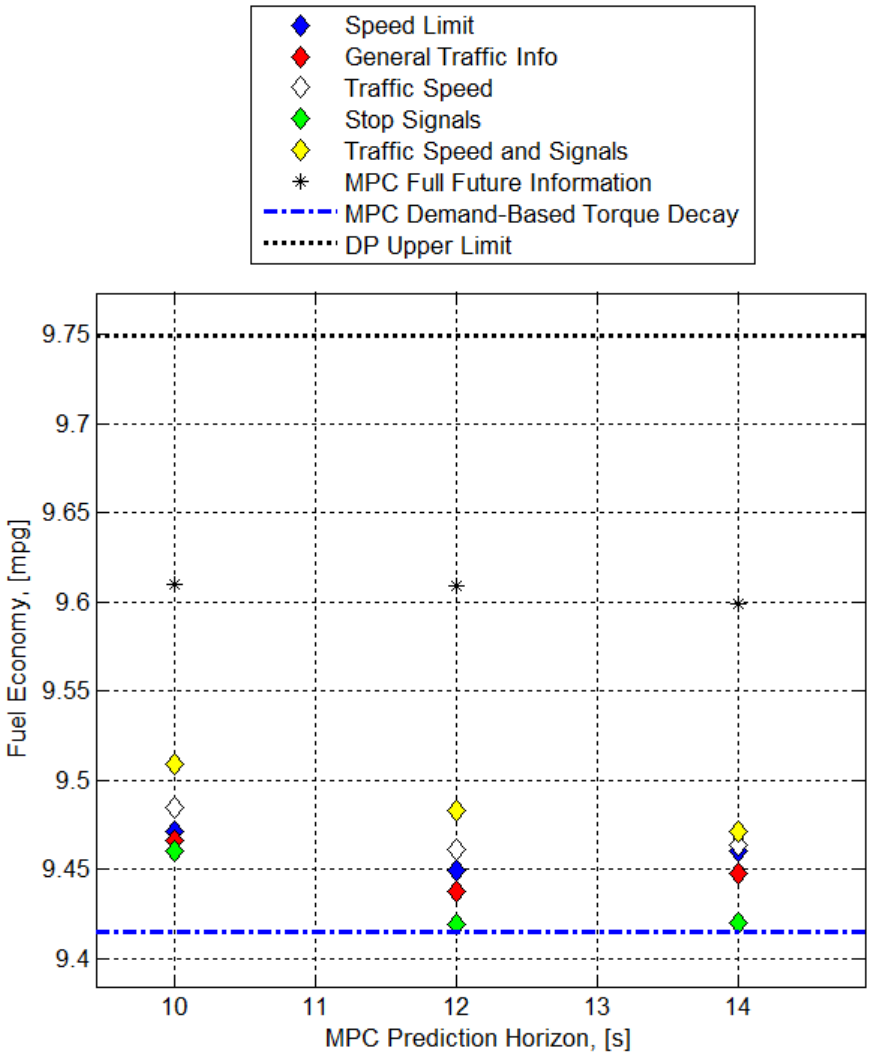


Figure 6.10: MPC Results - Future Information Benefits on Longer Horizons

or traffic speed with stop signals. However, it is interesting to see that the addition of the future information has the potential to improve fuel economy up to around 1% for these somewhat longer horizons considered. On the whole, the horizon lengths deemed best for the fuel economy improvement of the ultracapacitor heavy hybrid vehicle are quite short and a fuel economy of around 3% less than the DP calculated maximum is possible for a horizon length of around 10 seconds.

Chapter 7

Conclusions and Future Work

Starting with the calculation of imposed cycle demands on the heavy vehicle platform, a full vehicle model has been developed and validated against known performance metrics for the M1081 military transport vehicle. The use of ultra-capacitors to provide a power boost has demonstrated potential fuel economy improvements of nearly forty percent on the chosen heavy vehicle cycle. The fuel economy values found are heavily dependent on the choice of cycle and vehicle component parameters and for other urban cycles could result in even higher fuel economy. Since the upper limit to these fuel economy values is calculated by Dynamic Programming techniques, a more realistic forward-looking Model Predictive Control (MPC) method has been detailed which makes use of an exponential torque decay prediction method based on the work of [24]. The discrete regions of torque and decay rate of this MPC prediction method were improved based on the vehicle cycle demands to better represent the future expected demands. Through a variety of simulations at different prediction horizon lengths, a consistent improvement in fuel economy of around 36% has been shown for this "demand-based" prediction method, compared to the 30-34% for the discrete method. This illustrates the robustness of the MPC method that can give nearly optimal performance when a good prediction model is used.

The use of telematic future information has also been shown to aid in the prediction method and improve the potential fuel economy. A variety of simulated

future information signals were used along with simple rules to improve the prediction of future torque demands. The use of this information showed the potential to improve fuel economy up to one percent above the "demand-based" prediction method results. These promising results show the ability to improve the fuel economy even closer towards the attainable maximum with the presence of relatively simple information currently available in telematic and GPS devices.

A variety of difficulties face the adoption and acceptance of ultracapacitors as a power boost in hybrid vehicles. The primary difficulty is a familiarity one; these components are relatively new to many people and as such their effective implementation is not widely known. The primary control challenge facing the use of ultracapacitors is managing the use of the very small energy buffer, and this issue will always be present. While a few companies are working to produce ultracapacitor hybrid transit buses, it will take time before they are widely seen and accepted as a working technology. Returning to the life cycle costs of ultracapacitors and some of the benefits for their use from [39], there is an obvious need to explore these technologies further. In [39], the author claims the life cycle energy costs of using ultracapacitors is more than 60 times less than using lithium batteries, but using current prices the costs are seen to differ by a somewhat smaller factor of 20. Considering the robustness of the ultracapacitor technology in its power capabilities even at low temperatures, efficiency of power delivery, performance in harsh environments and extraordinary cycle longevity, it is likely that ultracapacitor products will become more and more common in all sorts of automotive applications [39].

Future research topics stemming from this work are numerous and should be mentioned at this time. In order to improve the realism of the modeling and simulation conducted, future simulations should make use of a vehicle velocity and road grade profile specifically tailored to the target vehicle's purpose. This is very important for heavy vehicles considering even small changes in road grade can result in significant power and torque demands. If possible, a real map-based driving route should be used which could also contain information about speed limits, real traffic conditions, signal locations and timings, along with the road elevation data. This would allow for a more complete assessment of the potential

benefits of future information in predicting future torque demands and improving mileage. A possible and realistic future goal would be to implement some of the developed control strategies on a real hybrid vehicle and conduct on-road testing to validate potential fuel economy benefits seen in simulations.

Additional future research may come from areas within the hybrid vehicle modeling conducted here. The consideration (and possibly even control) of more realistic accessory loads in the ultracapacitor hybrid vehicle platform is important. In this study, simplifying assumptions have been made which may not be entirely practical, especially considering the potential magnitude of electrical and mechanical accessory loads on heavy vehicles as well as passenger vehicles, loads which are ever-increasing due to the incorporation of more and more onboard technology. These accessory loads would have a significant effect on potential fuel consumption, both on a hybrid vehicle and the conventional vehicle. The use of components like a DC/DC converter were not considered here due to the perceived lack of benefit, however, it would be important to consider the use of such power electronics components if the vehicle was a battery and ultracapacitor hybrid. More potential fuel savings could also be obtained from the optimization of shifting strategy with the added consideration of the hybrid system capabilities. This could lead to slight performance and fuel economy improvements compared with a shifting strategy which is not optimized for the ultracapacitor heavy hybrid electric vehicle.

Appendices

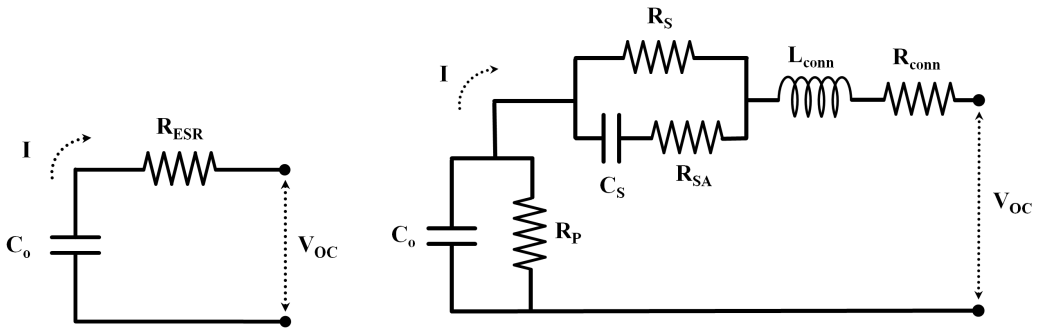
Appendix A Ultracapacitor Modeling Supplement

A.1 Modeling the Ultracapacitor

The ultracapacitor model is the most critical component in the modeling of the ultracapacitor hybrid electric vehicle. Small variations in parameters within this model can greatly affect both the hybrid system's performance and resulting fuel economy of the vehicle. This section will detail both the original PSAT model for an ultracapacitor as well as the updated "Maxwell model" based on data and files provided by Maxwell.

Typically, an ultracapacitor is modeled as a simple RC-circuit, as that which is shown in Figure 1a. This model includes both a main capacitance component as well as an "Equivalent Series Resistance" or ESR component. For simple circuit modeling the use of this model with constant parameter values may be sufficient, however, in highly transient cases, this model might not be as accurate as desired. The basic equation that governs the energy stored in a capacitor was given previously; this equation, along with the basic formulations of Ohm's law and resistive power losses are used in the PSAT model to calculate the energy, power, and state of charge of the ultracapacitor. The PSAT model makes use of two-dimensional lookup tables for cell capacitance, charging resistance, and discharging resistance, which are mapped according to temperature and discharge currents with data originally provided by Maxwell [6]. A variety of calculations are performed in the Simulink modeling blocks to calculate parameters like cell temperature, open circuit voltage, and cell state of charge[6]. In this way, PSAT is able to provide a fairly accurate yet simple model for the ultracapacitor cell.

While the original PSAT model likely would produce sufficient results, a more detailed and well-validated model provided by Maxwell has been configured to run in the PSAT simulation environment. The equivalent circuit diagram of this model is shown in 1b. Maxwell, one of the primary manufacturers of ultracapacitor products, has written a technical document (currently un-released) that details the development of a complete electrical and thermal model for their



(a) PSAT UC Model (b) Maxwell UC Model

Figure 1: Ultracapacitor Cell Modeling Comparison

"BoostCap" ultracapacitor line and validates it over a variety of tests[37]. A provided MATLAB/Simulink® model for the ultracapacitor cell has been adapted to run in the PSAT simulation environment to compare with the original PSAT model.

Where the PSAT model uses only an equivalent resistance and capacitance, this model includes multiple capacitances, resistances, and an inductance, including the important contact resistances which are present within an ultracapacitor pack. In the Maxwell model, the capacitance C_o is a two dimensional table lookup value which is indexed with respect to cell voltage and current draw[37]. The data for this is built into the model and is critical to the simulation due to the very strong voltage dependency of the main capacitance [37]. A plot of this voltage dependency can be seen in data taken from the Maxwell model, shown in 2.

Additional features of the model include cell aging, which accounts for capacitance and internal resistance change over the lifetime of cells, and thermal calculations to calculate cell operating temperatures[37]. All of the modeling features were correlated well with test data for cells and the thermal modeling is correlated on the pack level [37]. Maxwell is currently working on scaling this cell-level model to full packs and validating the scaled model over their different ultracapacitor modules [37].

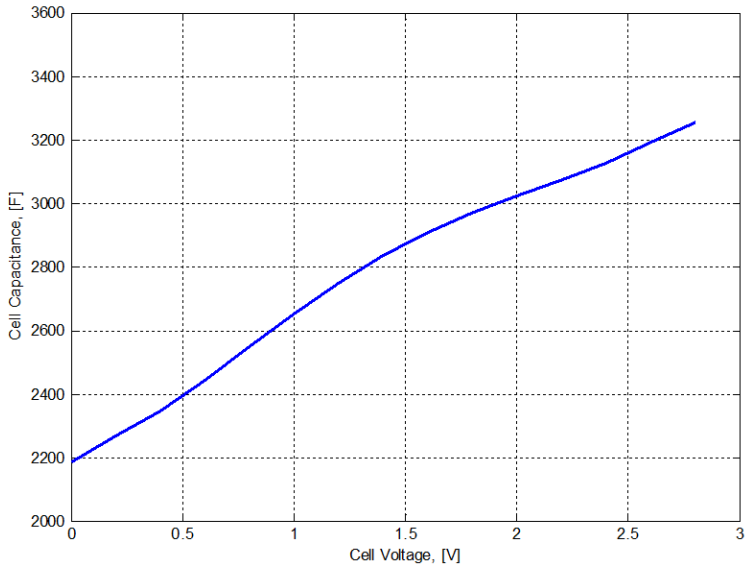


Figure 2: Ultracapacitor Cell Capacitance Map

For some modeling scenarios, it is useful to make the assumption of constant capacitance assumption to aid in deriving the dynamic equations of the system. In these cases, the capacitance of the ultracapacitor is assumed to be constant at the nominal value of the above curve between 2.6V and 1.5V (3000F for this particular cell), where it will be operating most of the time. The figure Later derivations will show

Comparative simulations in PSAT were conducted using both the PSAT ultracapacitor model and the adapted Maxwell model. These simulations were run on the CSHVR cycle and used the hybrid version of the M1081 vehicle with all parameters carefully matched to ensure comparable simulations. The differences in the resulting fuel economy results for the two models were only around 1%. This small of a difference might not seem very significant, but when the SOC trajectories and vehicle performance numbers are compared, it is obvious that the slightly different modeling considerations of the Maxwell model lead to a more restricted use of the ultracapacitor energy storage.

A.2 Modeling and Constraint Equations

Returning to the simple model of the ultracapacitor:

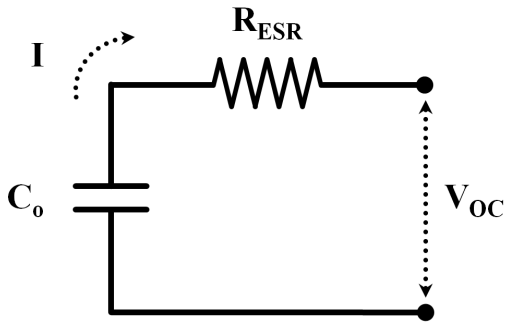


Figure 3: Ultracapacitor Cell Equivalent Circuit

The data provided by Maxwell in [38] is used to define the values of C_o and R_{ESR} (here noted as R_s). Initial derivations will assume the capacitance is constant and the desired outcome is a relationship to define the rate of change of SOC. The State of Charge of the ultracapacitor is defined on a charge basis as follows:

$$SOC = \frac{q}{q_{\max}} = \frac{C_o V}{C_o V_{\max}} = \frac{V}{V_{\max}} \tag{1}$$

Using the following relationships for the charge stored in the capacitor and the voltage loop equation of the circuit, a relationship for the rate of change of SOC is obtained.

$$q = C_o V_C \rightarrow I = \frac{dq}{dt} = C_o \frac{dV_C}{dt} \tag{2}$$

$$V_{UC} = V_C + V_R = -\frac{q}{C_o} - IR_s \tag{3}$$

$$P_{UC} = -IV_C - I^2 R_s \rightarrow I^2 R_s + ISOCV_{\max} + P_{UC} = 0 \tag{4}$$

The resulting Equation 4 is a quadratic equation in I which has the solution:

$$I = \frac{-SOC \cdot V_{\max} + \sqrt{(SOC \cdot V_{\max})^2 - 4R_s P_{UC}}}{2R_s} \tag{5}$$

Returning to the definition of SOC gives the input-output relationship:

$$\dot{SOC} = \frac{I}{C_o V_{\max}} = \frac{-SOC \cdot V_{\max} + \sqrt{(SOC \cdot V_{\max})^2 - 4R_s P_{UC}}}{2R_s C_o V_{\max}} \tag{6}$$

This relationship is used to determine the output of the system,$\dot{SOC}$, from the input power required of the ultracapacitor ESS, PUC, for an ultracapacitor pack at a given SOC and with constant resistance and capacitance.

When modeling the ultracapacitor, it is critical to include the proper constraints for the system. In this case, there is both a charging and discharging power constraint which must be considered. The discharging constraint comes from the terms under the radical in Equation 5, and gives the following:

$$P_{UC,max,discharge} < \frac{(SOC \cdot V_{\max})^2}{4R_s} \tag{7}$$

The charging constraint comes from the maximum open circuit voltage which can be applied to the ultracapacitor cells (in this case $V_{abs,max}$ =2.8V). Simple circuit equations lead to the relationship:

$$P_{UC,max,charge} > \frac{V_{abs,max}\left(SOC \cdot V_{max} - V_{abs,max}\right)}{R_s} \tag{8}$$

These constraints are easier to visualize if plotted over the range of allowable SOC's, as has been done in Figure 4.

As the figure shows, the nominal pack is capable of discharging between 19kW and 75kW of power depending on the SOC. Interestingly enough, the ultracapacitor power potential increases with the square of voltage, and only linearly decreases with resistance. This means that to increase the pack's discharge power capability, it is much better to string more cells together in series to give a higher operating voltage than to simply add strings in parallel.

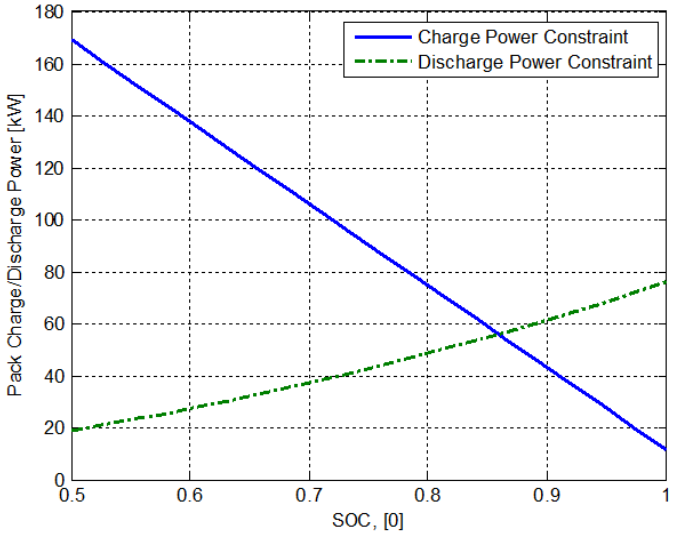


Figure 4: Ultracapacitor Charge/Discharge Power Constraints

A.3 Capacitance with Linear Relationship

The previous Figure 2 showed the strong voltage dependency of the cell capacitance plotted from Maxwell data. It is desired to know how this might change the modeling equations, so if the capacitance is now assumed to be a linear function of voltage as follows:

$$C_o(V_C) = \alpha V_C + \beta \rightarrow \frac{\partial C_o}{\partial V_C} = \alpha \tag{9}$$

The derivations proceed as before, this time using the product rule of derivatives:

$$q = C_o V_C \rightarrow dq = dC_o V_C + C_o(V_C) dV_C \tag{10}$$

Simplifying this using a first order approximation, this becomes:

$$dq = \frac{\partial C_o}{\partial V_C} dV_C + C_o(V_C) dV_C = \left(C_o(V_C) + \frac{\partial C_o}{\partial V_C} \right) dV_C \tag{11}$$

Using the integration of dq to obtain an expression for q:

$$q = \int_0^q dq = \int_0^{V_C} \left(C_o(V_C) + \frac{\partial C_o}{\partial V_C} \right) dV_C \tag{12}$$

$$q = \int_0^{V_C} (\alpha V_C + \beta + \alpha)\, dV_C \tag{13}$$

$$q = \frac{1}{2}\alpha V_C^2 + (\alpha + \beta) V_C \tag{14}$$

Going back to the charge based definition of SOC gives:

$$SOC = \frac{q}{q_{\max}} = \frac{\frac{1}{2}\alpha V_C^2 + (\alpha + \beta)\, V_C}{\frac{1}{2}\alpha V_{\max}^2 + (\alpha + \beta)\, V_{\max}} \tag{15}$$

Since the only term in this equation that is non-constant is V_C, we could take the derivative directly to get the expression for $S\dot{O}C$ This will result in a complicated expression containing dV_c, which will need an additional state to consider in the modeling. At this time, further exploration of this non-linear capacitance effects have been left to further research.

Appendix B Driving Cycle Information

In order to help define comparative baselines for vehicle consumption, a simple MATLAB® routine has been created to calculate a variety of power and energy requirements for a few standard driving cycles. This appendix will provide details on the driving cycles considered and the data obtained.

B.1 Driving Cycle Data: Sport Utility Vehicle

The first vehicle considered is that of an average sport utility vehicle(SUV) with the general specifications listed in Table 1.

Make	Model	Year	Engine Size	Power kW [hp]	Curb Weight kg [lbs]	Fuel Economy City / Highway
Ford	Explorer	2009	4.0L 6-Cylinder	157 [210]	2018 [4450]	13 / 19 mpg

Table 1: Baseline Passenger SUV Specifications

A wide range of driving scenarios will likely be encountered by a typical SUV ranging from slow urban traffic to high-speed highway driving. As such, a variety of cycles are considered for this vehicle as follows: New York City Traffic (low speed stop and go), Artemis Urban (moderate speed stop and go), UDDS (Urban Dynamometer Driving Schedule - moderate speed, more stop and go), EPA LA92 (low to high speed, more aggressive), Artemis Extra Urban (moderate and high speed), and Artemis Highway (highway driving at high speeds). The speed versus time profiles for the first three of these cycles is plotted in Figure 5. These cycles are the lower speed of the group and are more representative of city driving. The second three cycles are plotted in Figure 6. These are the interurban or highway style cycles, on which higher power demands are expected.

Using the velocity versus time input from the driving cycle and basic information about the vehicle (such as weight, drag parameters, and rolling resistance) the program calculates torques and powers required to overcome rolling

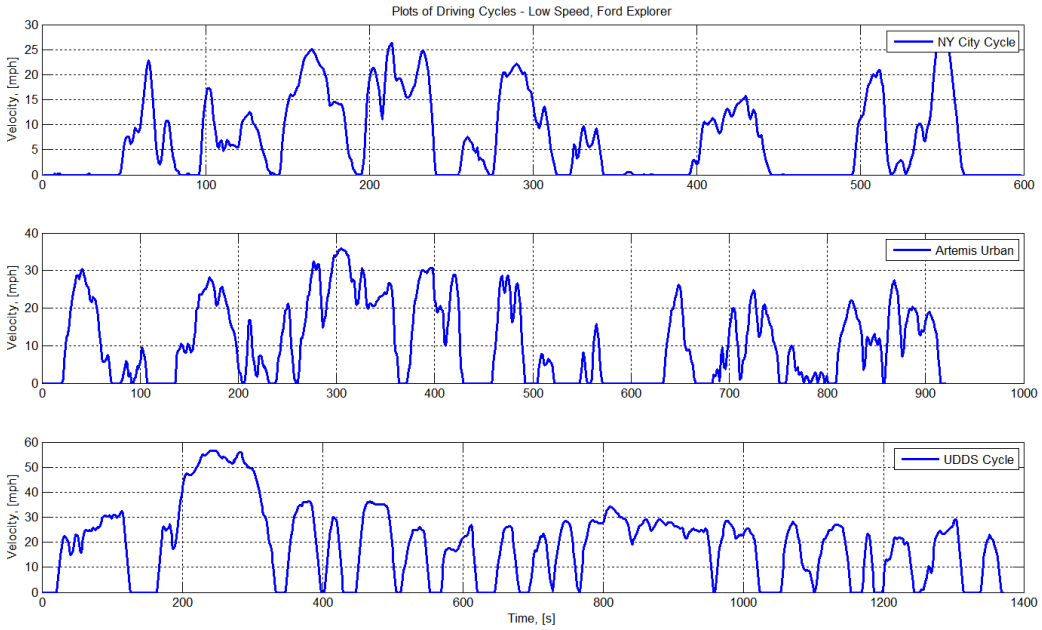


Figure 5: Passenger Vehicle Driving Cycles - Lower Speed

resistance, aerodynamic drag, inertia, road grade, and electrical accessory losses. From these calculations, characteristic energy and power requirements are tabulated to help describe the vehicle driving on this cycle. The detailed results of this MATLAB® routine for the sport utility vehicle run over these given cycles are presented in Table 2

B.2 Driving Cycle Data: Military Transport Vehicle

This research is primarily concerned with the use of ultracapacitors in larger vehicles, with the targeted vehicle being a 2.5-ton light military transport vehicle, in particular the M1081 standard cargo truck. This vehicle's purpose is to transport goods and troops at relatively lower speeds of up to around 50mph over a variety of terrains and conditions. The specifications for the M1081 are listed in Table 3 and were obtained from [13, 2].

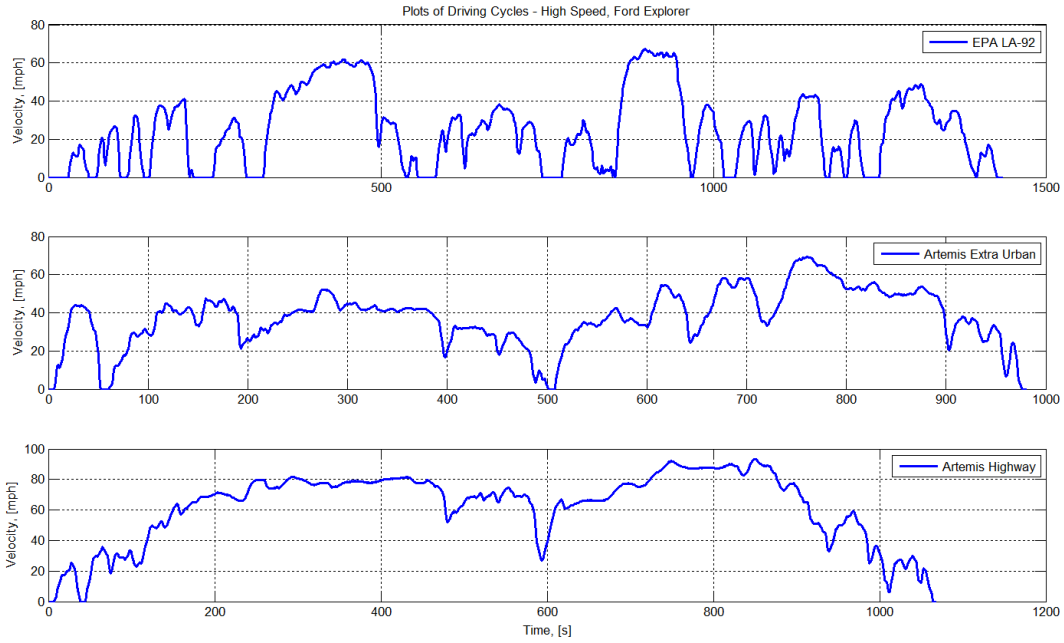


Figure 6: Passenger Vehicle Driving Cycles - Higher Speed

The mission of this vehicle is substantially different from that of the typical passenger car, and as such the previously mentioned driving cycles are not well-suited to this vehicle. The cycles considered for this vehicle are those more appropriate for refuse trucks and transit buses which generally have lower speeds with more stop and go driving. The cycles considered here are: Manhattan bus(low speed, stop and go), Artemis urban (moderate speed stop and go), CSHVR (City Suburban Heavy Vehicle Route - moderate speed stop and go), UDDS(moderate speed, more stop and go), and UDDS Truck (moderate speed, less stop and go). The speed/time profiles of these cycles are plotted in Figure 6.

The results from running the MATLAB® calculation routine with the M1081 vehicle over these cycles is presented in Table 4

As is evident from the values shown, the power and energy requirements of driving can vary substantially from vehicle to vehicle and also from cycle to cycle. This is one of the many reasons why it is important to accurately and precisely

Maximum Demanded Power[kW] or Energy[kJ]	NY City	— Urban	Artemis Cycles Extra Urban	— Highway	UDDS	EPA LA92
Propulsion Power	54.5	54.9	94.0	147	68.9	93.6
Regen Power	19.9	43.9	64.7	94.6	22.1	89.3
Avg Power(5 sec)	36.3	43.3	83.1	136	58.3	84.6
Avg Power(20 sec)	16.5	23.9	68.9	96.5	41.0	71.4
Avg Power(60 sec)	8.9	18.3	44.1	78.7	33.7	48.5
Energy (5 sec)	182	216	415	679	291	423
Energy (20 sec)	330	478	1378	1929	819	1427
Energy (60 sec)	534	1100	2644	4723	2019	2910

Table 2: Power and Energy Characteristics of Passenger Driving Cycles - Typical SUV

Model	Engine Size	Power kw [hp]	Curb Weight kg [lbs]	Fuel Economy
LMTV M1081	6.6L 6-Cyl Diesel	168 [225]	7900 [17500]	5.55 mpg

Table 3: Baseline Heavy Vehicle Specifications

define the target vehicle and driving cycle requirements before beginning any simulation or design studies.

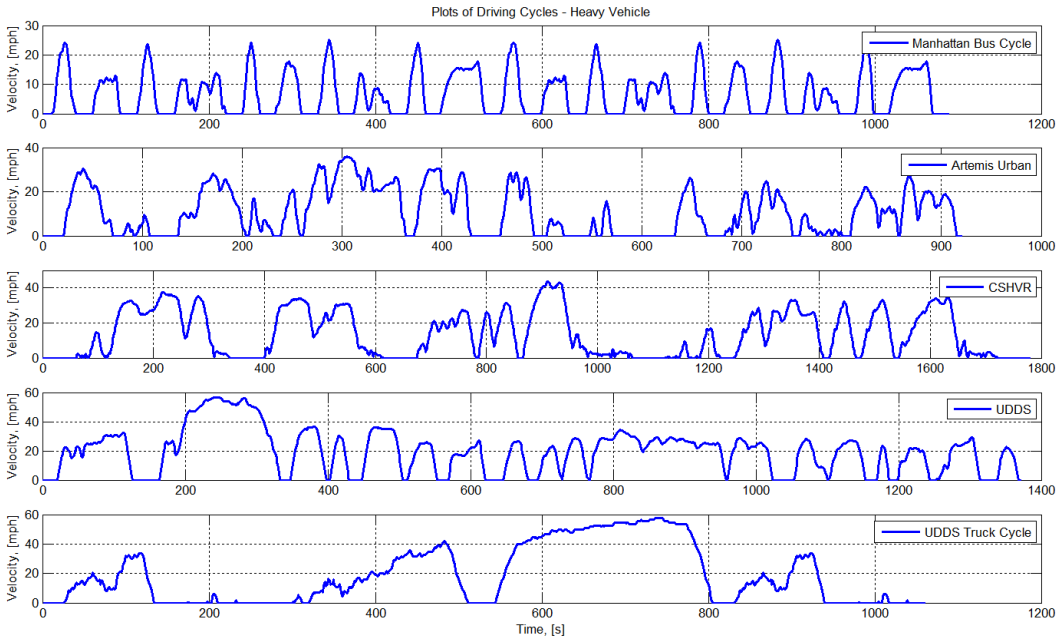


Figure 7: Heavy Vehicle Driving Cycles

Maximum Demanded Power[kW] or Energy[kJ]	Manhattan Bus	Artemis-Urban	CSHVR	UDDS	UDDS Truck
Propulsion Power	198	275	204	383	353
Regen Power	79.7	205	62.2	125	99.5
Avg Power(5 sec)	154	217	165	220	297
Avg Power(20 sec)	59.8	130	143	167	215
Avg Power(60 sec)	33.2	99.4	115	154	182
Energy (5 sec)	773	1088	825	1100	1487
Energy (20 sec)	1196	2604	2875	3340	4300
Energy (60 sec)	1992	5963	6910	9288	11000

Table 4: Power and Energy Characteristics of Heavy Vehicle Driving Cycles - M1081

Appendix C Simulation Data

This Appendix contains supplementary results from the DP program.

C.1 Baseline Simulation Cycle Data - DP Program

This section contains the baseline DP simulation results for the Central Business District (CBD) Truck and Manhattan Bus cycles, both of which are low speed stop and go cycles of similar character to the vehicles general purpose. The DP program is not able to run over the UDDS cycle due to the highly demanding nature of the cycle; A few separate times in the cycle higher power levels are demanded than the vehicle is capable of meeting.

The CBD Truck cycle is a very simple 2.17 mile stop and go cycle that is a short repeated stop-and-go sequence over 851 seconds. The input velocity and power demands, and the output engine and motor torque, as well as the SOC of the ultracapacitor can be seen below in Figure 8.

On this cycle, which is admittedly not very aggressive, the resulting fuel economy found (nominal configuration) is 9.67 mpg. Comparing this to the baseline simulation on the same cycle, which achieved 5.10 mpg, yields a 89% improvement. The engine torque is only needed for short spurts and the engine is able to be off for 669 out of the 851 seconds of the cycle, or 79% of the time, thus saving substantial amounts of fuel.

The Manhattan Bus driving schedule is a bit more aggressive, and the same results plotted for this cycle can be seen below in Figure 9.

While the Manhattan cycle looks similar in character to the CBD cycle, but is only a double repetition of a 570 second stop-and-go cycle. The results from this cycle were a fuel economy of 9.29 mpg, which is a 112% improvement over the baseline ICE simulation result. The cycle demands are such that the engine is only needed to be on for 230 out of the 1090 seconds of the cycle, which helps

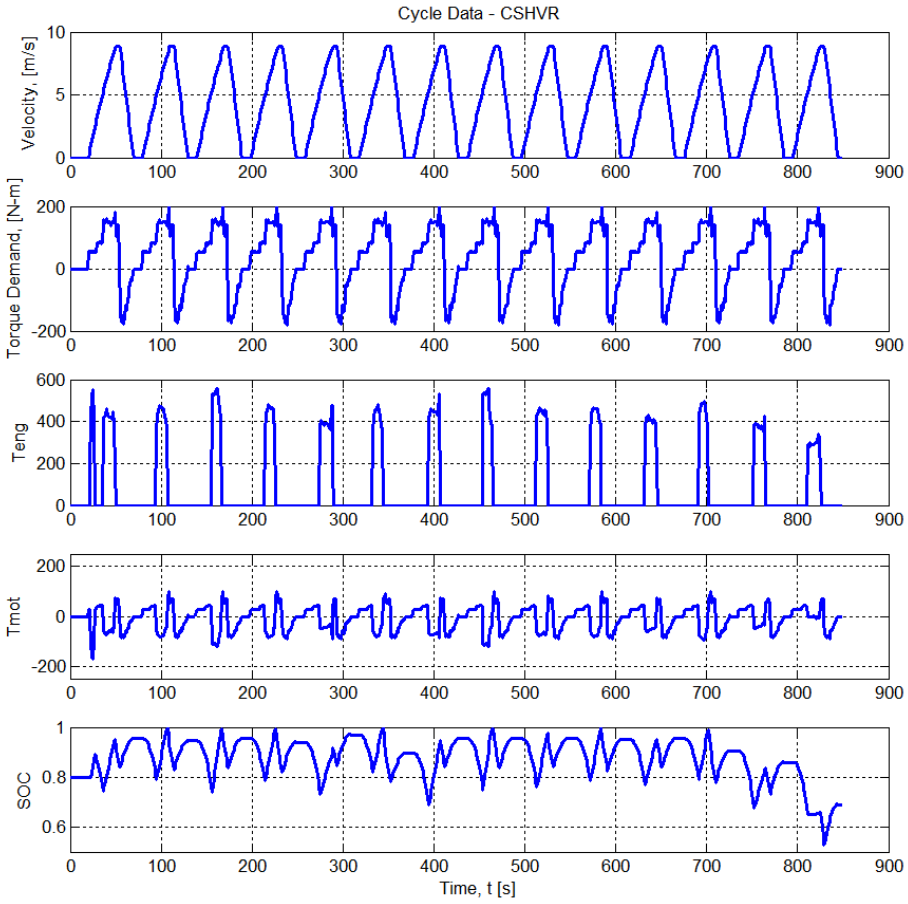


Figure 8: CBD Truck Cycle Inputs and Outputs

explain why the fuel economy improvement is so high. Regardless, this shows the potential benefits that engine on-off strategy can have on the fuel economy improvement of a hybrid vehicle.

Both of the above cycles also show that use of the full available SOC range is not necessarily needed to achieve the best possible fuel economy. This result agrees with the results for the baseline case, which also rarely approaches the constraint boundary.

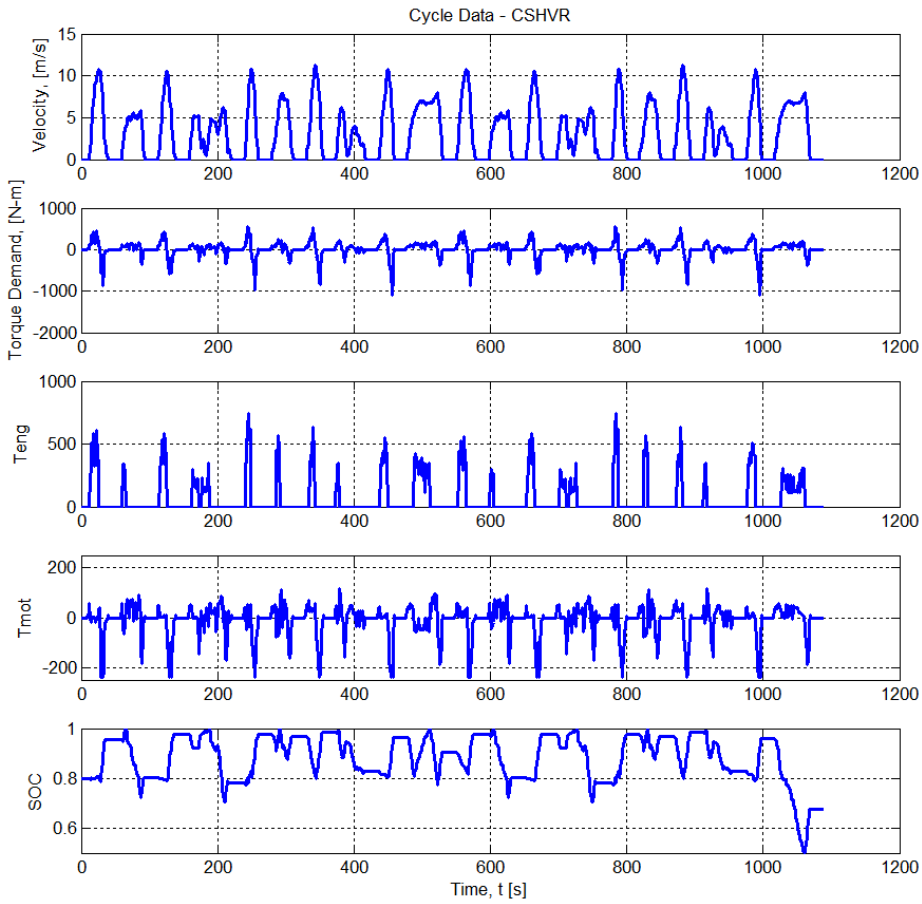


Figure 9: Manhattan Bus Cycle Inputs and Outputs

Appendix D MATLAB® Programs for DP and MPC

This Appendix contains the MATLAB® code for the DP and MPC implementations used for this thesis.

D.1 Dynamic Programming and Vehicle Initialization

The DP code is shown first, along with the vehicle initialization data definitions for the heavy vehicle model.

```
% DP_fuel_cons_final.m
% Seneca Schepmann                                          Spring 2010

% DP code which calculates the optimal inputs to minimize the fuel
% consumption (mass_fuel_rate) over the driving cycle. All units are SI.

% This program first initializes the hybrid vehicle model from PSAT-
% extracted data sets for the engine, motor, gearbox, UC, etc.
% The fixed speed shifting strategy is defined and the shifting points are
% calculated.
% The desired cycle velocity profile is input, power and torque
% requirements are calculated based on a simple vehicle model.
% These requirements at the wheel are transferred to demands at the engine
% - via the shifting strategy, known gear ratios, and efficiencies.

% The Dynamic Programming method then calculates the "optimal" fuel economy
% over the cycle using these inputs and the DP method.
% To run this program on the CSHVR cycle (1780 seconds long, designed for
% heavy urban vehicles like trucks) with a discretization of n = 50, the
% simulation takes approx 100seconds.

% Initialization sections define (from PSAT data):
% -Engine fuel rate map: Indexed by eng.spd_fuel_hot_index and
% eng.trq_fuel_hot_index
% -Engine output constrained to: eng.spd_max_hot_index used with
% eng.trq_max_hot_map and eng.pwr_max_hot_map

% -Motor efficiency map: Indexed by mc.spd_eff_index and mc.trq_eff_index
% efficiency numbers in  mc.eff_trq_map
% -Motor input constraints:
% Voltage range of 60-250V (mc.volt_min - mc.volt_max)
% Max current of 300A (mc.curr_max)
% -Motor output constraints:
% Propulsion Mode:
% Continuous duty:  mc.spd_prop_cont_index    mc.trq_prop_cont_map  mc.pwr_prop_cont_map
% Max duty:  mc.spd_prop_max_index  mc.trq_prop_max_map   mc.pwr_prop_max_map
% Regeneration Mode:
% Continuous: mc.spd_reg_cont_index  mc.trq_reg_cont_map    mc.pwr_reg_cont_map
% Max: mc.spd_reg_max_index   mc.trq_reg_max_map     mc.pwr_reg_max_map

% Fixed speed shifting strategy, should upshift and downshift at same speeds
```

```
% Current gear is indexed by throttle position shift.acc_dn_index
% and speed shift.veh_spd_upshift_index with gear ratio map given by shift.gear_upshift_map

clc; clear all; close all; format short; format compact;
%-------------------------------------------------------------------------%
%%%%%% Vehicle definition (~M1081 - Heavy Military Transport Vehicle)%%%%%%
%-------------------------------------------------------------------------%
tic;
% Vehicle - from 'veh_6000_546_075_M1081'
veh.body_mass               = 6000;     % vehicle mass without powertrain
veh.cargo_mass              = 2000;      % can be up to 2200kg(~5k lbs)
veh.frontal_area            = 5.46;
veh.coeff_drag              = 0.75;
veh.wheelbase               = 3.9;
veh.cg_height               = 1.5;      % approx
veh.ratio_wt_frt            = 0.4;      % ratio of weight on the front wheels
veh.ratio_wt_rear           = (1-veh.ratio_wt_frt);

% Wheel - from 'wh_05738_P395_85_R20_M1081'
wheel.mass                  = 100;
wheel.num                   = 4;        % Number of wheels
wheel.radius                = 0.5738;   % loaded vehicle radius
wheel.brake_trq_max         = 12000;    % N-m
% Rolling Resistance as a function of speed:  Frr = (CRR1 + CRR2*w)* m*g
wheel.CRR1                  = 0.025;
wheel.CRR2                  = 0.0003;

% Final drive - from 'fd_39' and 'trc_2'
fd.trc_ratio                = 2;       % transfer case ratio
fd.trc_eff                  = 0.96;    % transfer case efficiency
fd.ratio                    = 3.9;     % differential ratio
fd.eff                      = 0.97;    % final drive efficiency
fd.mass                     = 120;     % both the diff and trc masses, plus mech access
fd.tot_ratio = fd.trc_ratio*fd.ratio;  % overall ratio
fd.tot_eff = fd.trc_eff*fd.eff;        % overall efficiency

% Gearbox - from 'gb_7_au_564_066_AllisonMDD7_M1081'
gb.mass                     = 200;
gb.gear_nbr                 = 7;
gb.shift_time               = 0.6; % seconds to blend across the shift
gb.gear_index               = [0, 1, 2, 3, 4, 5, 6, 7];
gb.ratio_map                = [0, 5.64, 3.45, 1.84, 1.39, 1.0, 0.76, 0.66];
gb.eff_map                  = [0, 0.90, 0.92, 0.94, 0.96, 0.98, 0.98, 0.98];
% since efficiencies do not change with speed - these are constant values %

% Torque Coupling for motor:
tc.ratio = 2;
tc.mass = 20;

% Engine clutch
cpl.eff = 0.90;     % efficiency of clutch extracted from PSAT outputs - only affects engine torque

%-------------------------------------------------------------------------%
% Engine - from 'eng_ci_7300_171_Detroit_Diesel_S30'
eng.eng_mass                = 720;  % eng.init.mass_block + eng.init.mass_radiator...
 + eng.init.mass_vol + eng.init.tank_mass;
eng.fuel_mass               = 100;
eng.mass = eng.eng_mass + eng.fuel_mass;

eng.time_response           = 0.3;
eng.spd_idle                = 650*(2*pi/60);
%*** no fuel map at this point *** PSAT uses extrapolation *** see below
eng.spd_max                 = 2280*(2*pi/60);
eng.pwr_max                 = 171000; % Watts
```

```
eng.pwr_loss                = 4000;   % ***Due to Mechanical & Electrical accessories*** - Watts
eng.ex_gas_heat_cap         = 1089;   % J/kgK  ave sens heat cap of exh gas (SAE #890798)

eng.fuel_density_val        = 0.835;          % kg/L
eng.fuel_heating_val        = 42500000;       % (J/kg)Specific LHV
eng.fuel_carbon_ratio       = 12/13.8;        % (kg/kg) ref:Dr. Rob Thring

eng.displ_init              = 7300;     % cc
eng.inertia                 = 0.13;     % kg-m^2 - approximate value
eng.spd_str                 = 10;       % speed level (rad/s) the engine crank has to reach to start

% maximum curves at each speed (closed and wide open throttle)
% hot max wide open throttle curves
eng.spd_max_hot_index = [0 800 1200 1300 1320 1340 1360 1380 1400 1420 1440 1460 1480 1500 1520...
1540 1560 1580 1600 1620 1640 1660 1680 1700 1800 1900 2000 2100 2200 2300 2400 2500].*(2*pi/60);
eng.trq_max_hot_map= [0 432  614  637  644  655  661  672  677  702  727  750  774  796 799 803...
806  809  812  811  809  808  807  806  800  806  790  763  734  702   0   0];

% Mid speed (was used in logic to limit closed and wide open torque curves)
eng.spd_avg = 0.5*(eng.spd_max_hot_index(1)+ eng.spd_max_hot_index(length(eng.spd_max_hot_index)));

% hot max closed throttle curves
eng.spd_min_hot_index   = eng.spd_max_hot_index;
eng.trq_min_hot_map= [0 -5*ones(1,size(eng.spd_min_hot_index,2)-1)];

% fuel consumption table
eng.spd_fuel_hot_index = [800 1200 1300 1320 1340 1360 1380 1400 1420 1440 1460 1480 1500 1520...
1540 1560 1580 1600 1620 1640 1660 1680 1700 1800 1900 2000 2100 2200 2300].*(2*pi/60);
eng.trq_fuel_hot_index = [84  168  252  336  420  504  588  672  756];

% Rows represent speed (rad/s).  Columns represent torque (N-m).  Table is fuel rate (kg/s)
eng.fuel_hot_map =1e-3*[
    0.4037    1         1.4       1.8       2.2       2.6       3         3.4       3.8
    %approximated values for idle speed
    0.7098    1.2905    1.8214    2.3639    2.9109    3.4667    4.0137    4.5637    5.1077
    0.7991    1.3980    1.9731    2.5514    3.1257    3.7127    4.2926    4.8804    5.4762
    0.8175    1.4195    2.0035    2.5887    3.1682    3.7611    4.3473    4.9426    5.5488
    0.8361    1.4410    2.0338    2.6260    3.2104    3.8093    4.4018    5.0044    5.6211
    0.8549    1.4626    2.0642    2.6632    3.2525    3.8572    4.4558    5.0658    5.6930
    0.8739    1.4841    2.0946    2.7003    3.2945    3.9048    4.5095    5.1268    5.7646
    0.8931    1.5056    2.1249    2.7374    3.3362    3.9521    4.5629    5.1874    5.8358
    0.9100    1.5299    2.1553    2.7737    3.3830    4.0076    4.6281    5.2615    5.9192
    0.9270    1.5542    2.1856    2.8100    3.4298    4.0629    4.6933    5.3356    6.0025
    0.9442    1.5787    2.2160    2.8462    3.4765    4.1183    4.7585    5.4097    6.0859
    0.9615    1.6032    2.2463    2.8823    3.5232    4.1736    4.8237    5.4838    6.1693
    0.9789    1.6278    2.2767    2.9183    3.5699    4.2289    4.8888    5.5579    6.2527
    0.9964    1.6525    2.3070    2.9542    3.6166    4.2842    4.9540    5.6320    6.3360
    1.0140    1.6772    2.3374    2.9901    3.6633    4.3394    5.0192    5.7061    6.4194
    1.0317    1.7020    2.3678    3.0259    3.7099    4.3947    5.0844    5.7802    6.5028
    1.0496    1.7270    2.3981    3.0616    3.7565    4.4498    5.1496    5.8543    6.5861
    1.0676    1.7519    2.4285    3.0972    3.8030    4.5050    5.2148    5.9284    6.6695
    1.0869    1.7770    2.4612    3.1399    3.8535    4.5649    5.2827    6.0089    6.7600
    1.1063    1.8022    2.4940    3.1826    3.9041    4.6248    5.3507    6.0895    6.8507
    1.1259    1.8274    2.5268    3.2255    3.9548    4.6849    5.4188    6.1702    6.9415
    1.1456    1.8527    2.5598    3.2685    4.0055    4.7450    5.4870    6.2511    7.0325
    1.1655    1.8781    2.5927    3.3115    4.0563    4.8053    5.5552    6.3322    7.1237
    1.2670    2.0061    2.7584    3.5283    4.3114    5.1077    5.8974    6.7399    7.5824
    1.3374    2.1640    2.9674    3.7894    4.6322    5.4751    6.3226    7.2258    8.1290
    1.4078    2.3268    3.1822    4.0572    4.9616    5.8512    6.7580    7.7234    8.6888
    1.4885    2.5047    3.4183    4.3217    5.2481    6.2362    7.1677    8.1917    9.2157
    1.5983    2.6896    3.6706    4.6407    5.6355    6.6964    7.6968    8.7963    9.8959
    1.7128    2.8822    3.9334    4.9729    6.0389    7.1759    8.2478    9.4261   10.6043]*1.0195;

%---------- This stuff has been changed slightly ---------%
```

```
% Expanding the map at low torque (0 10 20)
%%% AND high torque (775 812) to allow upper torque values to be reached %%
for i=1:length(eng.spd_fuel_hot_index)
    tmp_map(i,:)=interp1(eng.trq_fuel_hot_index,eng.fuel_hot_map(i,:),[0 10 20...
     eng.trq_fuel_hot_index 775 812],'linear','extrap');
end
eng.trq_fuel_hot_index=[0 10 20 eng.trq_fuel_hot_index 775 812];

%Maps negative values are replaced by the average of non negative closest values
for i =1:length(eng.spd_fuel_hot_index)
    for j=1:length(eng.trq_fuel_hot_index)
        if tmp_map(i,j)<0 && i>1
            tmp_map(i,j)=abs(0.5*(tmp_map(i-1,j)+tmp_map(i+1,j)));
        end
        if tmp_map(i,j)<0 && i==1
           tmp_map(i,j)=abs(0.5*(tmp_map(i,j+1)+tmp_map(i+1,j)));
        end
    end
end

eng.fuel_hot_map=tmp_map;
clear tmp_map i j

% --- Expanding the map down to IDLE speed --- (Added) %
for i=1:length(eng.trq_fuel_hot_index)
    tmp_map(:,i)=interp1(eng.spd_fuel_hot_index,eng.fuel_hot_map(:,i),[eng.spd_idle...
      eng.spd_fuel_hot_index],'linear','extrap');
end
eng.spd_fuel_hot_index=[eng.spd_idle eng.spd_fuel_hot_index];

eng.fuel_hot_map=tmp_map;
eng.fuel_hot_map(1,1) = eng.fuel_hot_map(2,1);
clear tmp_map i j
%-------------------------------------------------%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DIESEL ENGINE
% Friction Torque for a 1.7 L Engine Based on Heywood. "Internal Combustion Fundamentals,"
% Section 13.6.1 Figure 13-14 (b) P726 Polynomial interp at low speed... Linear interp at high
% speed. Otherwise 6000 rpm equals 150 N-m of negative torque
eng.spd_zero_fuel_hot_index =(0:20:max(eng.spd_min_hot_index));
eng.tmp.half_way_idx = floor(length(eng.spd_zero_fuel_hot_index)/2);
eng.tmp.half_way_idx(eng.tmp.half_way_idx<2)=2;
eng.tmp.fmep_hot_map_part1 =  polyval([(60/2/pi)^2*1.5625e-005  (60/2/pi)*0.02875 86],...
eng.spd_zero_fuel_hot_index(1:eng.tmp.half_way_idx-1));
eng.tmp.fmep_slope_part2 =  polyval([2*(60/2/pi)^2*1.5625e-005 (60/2/pi)*0.02875],...
eng.spd_zero_fuel_hot_index(eng.tmp.half_way_idx));
eng.tmp.fmep_intercept_part2 = polyval([(60/2/pi)^2*1.5625e-005  (60/2/pi)*0.02875 86],...
eng.spd_zero_fuel_hot_index(eng.tmp.half_way_idx)) - eng.tmp.fmep_slope_part2.*...
eng.spd_zero_fuel_hot_index(eng.tmp.half_way_idx);
eng.tmp.fmep_hot_map_part2 =  polyval([eng.tmp.fmep_slope_part2...
eng.tmp.fmep_intercept_part2],eng.spd_zero_fuel_hot_index(eng.tmp.half_way_idx:end));
eng.fmep_hot_map = [eng.tmp.fmep_hot_map_part1 eng.tmp.fmep_hot_map_part2];
eng.fmep_hot_map(1) =0;
eng.trq_zero_fuel_hot_index = -eng.fmep_hot_map * eng.displ_init/1000 / 4/ pi;

%Calculations
% maximum and minimum calculations
eng.trq_hot_max = max(eng.trq_max_hot_map); % N-m
[eng.pwr_hot_max,Idx]  = max(eng.spd_max_hot_index.* eng.trq_max_hot_map); % W
eng.pwr_max_hot_map = eng.spd_max_hot_index.* eng.trq_max_hot_map; % W

%Code to compute maximum speed of the engine. Speed at 80% of peak power.
eng.spd_max =   eng.spd_max_hot_index(Idx);
```

```
if Idx < length(eng.pwr_max_hot_map)
    eng.spd_max = min(interp1(eng.pwr_max_hot_map(Idx:end)+1e-6*(1:...
    length(eng.pwr_max_hot_map(Idx:end))),...
        eng.spd_max_hot_index(Idx:end),eng.pwr_hot_max * 0.80,'linear','extrap')...
        ,max(eng.spd_max_hot_index));
end

% Calculate the max engine efficiency in within the max torque curve
eng.eff_hot_map    = eng.spd_fuel_hot_index'*eng.trq_fuel_hot_index/...
eng.fuel_heating_val./(eng.fuel_hot_map);
eng.tmp.max_trq         = interp1(eng.spd_max_hot_index,eng.trq_max_hot_map...
,eng.spd_fuel_hot_index);
eng.tmp.max_trq         = eng.tmp.max_trq(:)*ones(1,length(eng.trq_fuel_hot_index));
eng.tmp.max_trq         = (eng.trq_fuel_hot_index(:) * ones(1,...
length(eng.spd_fuel_hot_index)))' > eng.tmp.max_trq;
eng.eff_hot_map(eng.tmp.max_trq) = 0;
eng.eff_max        = max(max(eng.eff_hot_map));

eng = rmfield(eng,'tmp'); clear Idx

%-------------------------------------------------------------------------%
% Energy Storage - from 'ess_ultracap_Maxwell_model'
ess.num_module_parallel = 1;
ess.num_module_series = 3;

ess.soc_init    = 0.8;
ess.soc_min     = 0.5;              % no units(0.5 Vmax, 0.25 Emax)
ess.soc_max     = 1;                % no units( Vmax, Emax)(uses 75% of total energy)
ess.mass_cell              = 0.55;  % 0.55Kg for 1 cell
ess.packaging_factor       = 1.43;  % 18 cells at 0.55kg = 9.9kg, module wt 14.2kg

ess.cell_volt_abs_max = 2.7;                              % max cell voltage
ess.cell_volt_max_surge = 2.8;                               % max surge OC cell voltage
ess.cell_volt_min = ess.cell_volt_abs_max*ess.soc_min;    % min cell voltage allowed
ess.cell_volt_max = ess.cell_volt_abs_max*ess.soc_max;    % max cell voltage allowed
ess.cell_volt_init = ess.cell_volt_abs_max*ess.soc_init;  % initial cell voltage
ess.chg_eff = 0.92;      % Charging efficiency, good correlation to PSAT

ess.curr_chg_max = 500*ess.num_module_parallel;     % 500A per string of cells,
ess.curr_dis_max = 500*ess.num_module_parallel;     % from Maxwell spec sheet
ess.i2t_1sec_rating = 1000000;                      % i^2t rating for 500A*1sec

% Total Line and Connection resistance for UC to motor connections - IMPORTANT!!
ess.res_line = 0.05;

%Choose type of cell or Module:
ess.cell_module = 'BMOD0165P048';

% Specify Bin Number - capacitance ratings sorted by bins - ~2% difference
% between UCs sorted into different bins (1-7)
ess.bin = 1;
ess.num_cell = 18;

% 'BMOD0165P048' - 3000P Cells
        ess.rconn = 70e-6;
        ess.lconn = 63.7e-9;
        ess.rs = 0.3e-3;
        ess.rp = 643.333;
        ess.rth = 3.2;
        ess.cth = 588;

        ess.co_v = 0:0.2:2.8;
        ess.co_i = [-500 -400 -300 -200 -100 -75 -50 -40 -30 -20 -10 10 20 30 40 50...
         75 100 200 300 400 500];
```

```
        ess.co_c = (1+(ess.bin-3)*.02)*[
            2281.5 2367.1 2453.2 2548.7 2654.7 2762.7 2866.1 2960.0 3040.8...
             3106.8 3157.9 3195.4 3222.4 3257.0 3288.0
            2300.7 2378.1 2448.6 2539.8 2642.3 2748.4 2851.4 2946.3 3029.2...
             3097.8 3151.0 3189.1 3213.8 3245.9 3276.6
            2314.2 2390.9 2459.4 2550.9 2655.0 2762.8 2867.1 2962.4 3044.8...
             3112.3 3164.3 3202.1 3228.5 3261.2 3292.1
            2321.9 2396.7 2468.9 2560.7 2662.8 2767.3 2867.8 2959.4 3038.7...
             3103.7 3154.2 3191.1 3217.1 3245.9 3274.6
            2331.7 2407.1 2473.7 2563.7 2666.0 2771.4 2872.4 2963.8 3042.3...
             3106.4 3156.6 3195.6 3227.8 3259.6 3282.0
            2335.0 2408.5 2474.7 2564.7 2666.9 2771.4 2870.9 2959.9 3035.4...
             3096.3 3143.9 3181.5 3214.8 3251.5 3267.2
            2348.0 2418.6 2491.5 2584.8 2687.9 2792.3 2891.1 2979.4 3054.2...
             3114.2 3159.9 3193.8 3220.3 3245.3 3274.0
            2343.7 2415.8 2483.0 2573.2 2674.7 2778.3 2876.5 2964.3 3038.5...
             3098.2 3144.6 3180.8 3212.2 3246.3 3263.0
            2340.3 2413.6 2480.5 2570.3 2671.4 2774.6 2872.6 2960.4 3034.8...
             3095.2 3142.5 3180.3 3213.8 3250.6 3263.9
            2357.7 2427.6 2496.0 2585.9 2686.6 2788.9 2886.0 2972.8 3046.1...
             3104.7 3149.5 3183.2 3210.3 3237.5 3259.4
            2364.4 2432.0 2499.6 2588.4 2687.5 2788.3 2883.7 2968.8 3040.3...
             3097.1 3139.8 3170.9 3194.8 3217.9 3241.0
            2255.8 2338.4 2420.9 2524.9 2637.2 2747.8 2850.3 2941.4 3021.3...
             3093.3 3164.2 3244.3 3346.8 3488.7 3630.5
            2255.0 2339.3 2423.6 2528.9 2642.2 2753.8 2857.1 2948.8 3028.5...
             3099.5 3167.8 3242.9 3337.3 3467.0 3596.6
            2246.7 2331.8 2416.9 2522.6 2636.0 2747.2 2849.5 2939.5 3017.0...
             3085.1 3149.8 3220.5 3310.0 3433.9 3557.8
            2240.5 2329.7 2418.9 2526.8 2641.1 2752.3 2854.4 2944.0 3021.2...
             3089.0 3153.3 3223.4 3311.4 3432.6 3553.8
            2243.4 2332.7 2422.0 2531.7 2648.8 2763.4 2869.0 2962.0 3042.3...
             3112.6 3179.1 3251.0 3340.8 3464.0 3587.3
            2228.1 2321.8 2415.4 2525.0 2639.6 2750.5 2852.3 2941.8 3019.0...
             3086.2 3148.8 3214.8 3294.7 3402.1 3509.5
            2237.3 2325.0 2412.8 2521.1 2637.0 2750.8 2855.5 2947.5 3026.2...
             3094.0 3156.6 3222.4 3303.1 3413.6 3524.1
            2208.6 2309.3 2410.0 2523.8 2640.4 2752.3 2854.1 2943.2 3019.4...
             3085.0 3145.0 3206.5 3279.7 3376.7 3473.8
            2190.0 2296.5 2402.9 2519.6 2637.5 2749.9 2852.2 2942.0 3019.1...
             3085.5 3145.6 3205.8 3274.6 3343.5 3412.4
            2167.2 2280.2 2393.3 2512.0 2628.9 2738.5 2837.3 2923.6 2997.7...
             3061.9 3120.3 3179.0 3245.9 3312.9 3379.8
            2149.4 2272.2 2395.1 2517.9 2637.0 2747.2 2845.3 2930.3 3002.8...
             3065.6 3123.2 3182.3 3251.5 3320.6 3389.7
            ];


    % Total number of cells, max and min pack voltages
    ess.num_cell_series = ess.num_module_series*ess.num_cell;
    ess.mass = ess.mass_cell*ess.num_cell_series*ess.packaging_factor;

    % Max charge and discharge power per cell
    ess.pwr_chg      = ess.cell_volt_max*ess.curr_chg_max + ess.rs*ess.curr_chg_max^2; %per cell
    ess.pwr_dis      = ess.cell_volt_max*ess.curr_dis_max - ess.rs*ess.curr_dis_max^2; %per cell

    % Energy storage density
    ess.pwr_density    = ess.pwr_dis/ess.mass_cell;
    ess.energy_density = max(mean(0.5*ess.co_c'*ess.cell_volt_max^2))/ess.mass_cell;
    %calcs the average capacitance at each current level(over all voltages), then takes max
    ess.energy_capacity = ess.energy_density*ess.num_cell_series*ess.mass_cell;

    % Simplify the data for simple RC model (per pack)
    ess.pack_cap = mean(ess.co_c)/ess.num_cell_series;  % now cell cap only indexed wrt voltage
```

```
ess.pack_v = ess.co_v*ess.num_cell_series;
ess.pack_polyfit = polyfit(ess.pack_v,ess.pack_cap,1);  % linear polyfit for pack wrt voltage
ess.pack_vmax = ess.cell_volt_max*ess.num_cell_series;  % max pack voltage allowed
ess.pack_vmax_surge = ess.cell_volt_max_surge*ess.num_cell_series;  % max pack surge voltage
ess.pack_vmin = ess.cell_volt_min*ess.num_cell_series;    % min pack voltage allowed
ess.pack_vinit = ess.cell_volt_init*ess.num_cell_series;  % initial pack voltage
% total pack resistance - Important to be accurate here
ess.pack_res = (ess.rconn + ess.rs)*ess.num_cell_series/ess.num_module_parallel + ess.res_line;
% max energy content of the pack - Emax = 0.5*C*Vmax^2
ess.pack_Emax = 0.5*(ess.pack_vmax^2*(mean(ess.pack_cap)));

%--------------------------------------------------------------------------%
% Motor - from 'mc_pm_36_75_UQM_PowerPhase75'
mc.inertia              = 0.047;
mc.coeff_regen          = 1;
mc.volt_min             = 60;    % min voltage allowed by the controller and motor
mc.volt_max             = 250;
mc.time_response        = 0.05;
mc.t_max_trq            = 180;   % Time the motor can remain at max torque
mc.motor_mass           = 47.7;
mc.controller_mass      = 13.1;
mc.mass = mc.motor_mass + mc.controller_mass;

mc.curr_max             = 300;           % A, maximum current allowed by the controller and motor
mc.spd_base             = 3000*(2*pi/60); % rad/s

mc.spd_cont_index       = [0 1200 1500 2000 2500 3000 3500 4000 4500 5000 6000 7000...
 7500 7700 7800]*(2*pi/60);
mc.trq_cont_map         = [150 150 150 150 137 112 90 76 65 60 46 42 42 0 0]; % (N*m)

mc.spd_max_index        = [0 1200 1500 2000 2500 3000 3500 4000 4500 5000 6000 7000...
 7500 7700]*(2*pi/60);
mc.trq_max_map          = [240 240 240  240 240 238 204 179 159 143 119 102 95 0];

mc.spd_min_index       = mc.spd_max_index; % rad/s
mc.trq_min_map         = -mc.trq_max_map;

mc.spd_eff_index        = [0 1200 1500 2000 2500 3000 3500 4000 4500 5000 6000 7000 7500]*(2*pi/60);
mc.trq_eff_index        = [0 5 25 50 75 100 125 150 175 200 240];
mc.eff_trq_map          =  0.01*[...
60 80.5 85   88   88   86   85   86   80   79   79
60 80.5 85   88   88   86   85   86   80   79   79
60 81   85   89   89   88   87   86   82   80   79.5
60 80   87   90   90.5 90   89   88   86   85   83
60 78   88   91.5 91.5 91   91   90   89   86   85
60 79   91   92.5 93   92.5 92   91   90   87   87
60 79   93   94   93.5 93   93   92   90   90   90
60 86   93   94   94   93   92.5 92   92   92   92
60 79   90   92   93   92   91   91   91   91   91
60 79   86   91   92   92   92   92   92   92   92
60 77.5 86   91   92   92   92   92   92   92   92
60 77.5 83   90   90.5 90.5 90.5 90.5 90.5 90.5 90.5
60 77.5 81   90   90   90   90   90   90   90   90];

mc.spd_prop_cont_index = [-fliplr(mc.spd_cont_index(2:end)) -eps 0 eps mc.spd_cont_index(2:end)];
mc.trq_prop_cont_map   = [-fliplr(mc.trq_cont_map(2:end))  -mc.trq_cont_map(2) mc.trq_cont_map(2) ...
mc.trq_cont_map(2) mc.trq_cont_map(2:end)];
mc.pwr_prop_cont_map = mc.spd_prop_cont_index.*mc.trq_prop_cont_map;

mc.spd_prop_max_index  = [-fliplr(mc.spd_max_index(2:end)) -eps 0 eps mc.spd_max_index(2:end)];
mc.trq_prop_max_map    = [-fliplr(mc.trq_max_map(2:end))   -mc.trq_max_map(2) mc.trq_max_map(2) ...
mc.trq_max_map(2) mc.trq_max_map(2:end)];
mc.pwr_prop_max_map    =  mc.spd_prop_max_index.*mc.trq_prop_max_map;
```

```
mc.spd_reg_cont_index  = [-fliplr(mc.spd_cont_index(2:end)) -eps 0 eps mc.spd_cont_index(2:end)];
mc.trq_reg_cont_map    = [fliplr(mc.trq_cont_map(2:end))  mc.trq_cont_map(2) -mc.trq_cont_map(2) ...
-mc.trq_cont_map(2)  -mc.trq_cont_map(2:end)];
mc.pwr_reg_cont_map = mc.spd_reg_cont_index.*mc.trq_reg_cont_map;

mc.spd_reg_max_index   = [-fliplr(mc.spd_max_index(2:end)) -eps 0 eps mc.spd_max_index(2:end)];
mc.trq_reg_max_map     = [fliplr(mc.trq_max_map(2:end))    mc.trq_max_map(2) -mc.trq_max_map(2) ...
-mc.trq_max_map(2) -mc.trq_max_map(2:end)];
mc.pwr_reg_max_map     = mc.spd_reg_max_index.*mc.trq_reg_max_map;

mc.spd_eff_index       = [-fliplr(mc.spd_eff_index(2:end)) mc.spd_eff_index];
mc.trq_eff_index       = [-fliplr(mc.trq_eff_index(2:end)) mc.trq_eff_index];
mc.eff_trq_map         = [flipud(fliplr(mc.eff_trq_map(2:end,2:end))) flipud(mc.eff_trq_map(2:end,:));
fliplr(mc.eff_trq_map(:,2:end)) mc.eff_trq_map];

%Power Losses at zero speed (modified from mc_pre_calculation and mc_calculation files):
mc.pwr_mech_map    = mc.spd_eff_index' * mc.trq_eff_index;

tmp.position_zero_spd           = find(mc.spd_eff_index==0);
tmp.position_zero_trq           = find(mc.trq_eff_index==0);

% Power fraction applied to zero speed case for losses
mc.pwr_frac_locked_rotor   = 0.05;
mc.pwr_elec_loss_map = mc.pwr_mech_map .*(1-mc.eff_trq_map);

mc.pwr_elec_loss_map(tmp.position_zero_spd,:) = mc.pwr_elec_loss_map(tmp.position_zero_spd+1,:)...
*mc.pwr_frac_locked_rotor;
% Always positive power loss!
mc.pwr_elec_loss_map = abs(mc.pwr_elec_loss_map);

mc.pwr_elec_map = mc.pwr_mech_map + mc.pwr_elec_loss_map;

%--------------------------------------------------------------------------%
% Total Mass Computation:
mass = veh.body_mass + veh.cargo_mass + wheel.mass*wheel.num + fd.mass ...
    + gb.mass + eng.mass + ess.mass + mc.mass + tc.mass;

%--------------------------------------------------------------------------%
% shifting map - from 'tx_stf_shift_n_gen_eng_fixed_spd'
shift.percent_eng_spd_idle                = 1.1;
shift.percent_eng_spd_max                 = 0.98;
shift.percent_between_opt_and_max_speed   = 0.15;
shift.percent_between_idle_and_opt_speed  = 0.25;

shift.min_time_for_shift_dmd              = 1;    % [s] minimal time the up- or down- shifting...
 request has to be constant before performing the shifting
shift.acc_below_no_timer                = 0;  % [0->1] if the accelerator pedal position is...
 below this value, the timer is by-passed

shift.tmp.trs_eng_spd_veh_start = 2*eng.spd_idle;

shift.tmp.gear_ratio=gb.ratio_map;

if min(gb.ratio_map)==0
    shift.tmp.gear_ratio(1)='';
end

%Vehicle speed below which neutral gear can be requested (for manual tx)
shift.veh_spd_max_neutro=(eng.spd_idle*shift.percent_eng_spd_idle+10)*wheel.radius/fd.ratio/...
shift.tmp.gear_ratio(1)/fd.trc_ratio;

% Shift table accelerator command - Parallel Hybrid
    [shift.tmp.spd_index, shift.tmp.trq_index]=find(max(max(eng.eff_hot_map))==eng.eff_hot_map);
    shift.tmp.spd_index = shift.tmp.spd_index(1); %In case there is more than one optimal point.
```

```
    % Just take the first one.
    shift.tmp.trq_index = shift.tmp.trq_index(1); %In case there is more than one optimal point.
    shift.tmp.spd_op_max = eng.spd_fuel_hot_index(shift.tmp.spd_index)...
        + shift.percent_between_opt_and_max_speed*(max(eng.spd_max_hot_index(1:end-2))- ...
        eng.spd_fuel_hot_index(shift.tmp.spd_index));
    shift.tmp.spd_op_min = eng.spd_idle + shift.percent_between_idle_and_opt_speed...
     *(eng.spd_fuel_hot_index(shift.tmp.spd_index)- eng.spd_idle);
%4WD
    shift.tmp.veh_spd_op_max=shift.tmp.spd_op_max*wheel.radius/fd.ratio/...
    fd.trc_ratio./shift.tmp.gear_ratio;
    shift.tmp.veh_spd_op_min=shift.tmp.spd_op_min*wheel.radius/fd.ratio/...
    fd.trc_ratio./shift.tmp.gear_ratio;

cpt2 = 1;
for cpt=length(shift.tmp.gear_ratio):-1:2
    shift.acc_up_index(cpt2) = shift.tmp.gear_ratio(cpt)/shift.tmp.gear_ratio(1);
    cpt2=cpt2+1;
end
clear cpt cpt2
shift.acc_up_index=[0 shift.acc_up_index];
shift.acc_up_index(end+1)=1;

cpt2 = 1;
for cpt=length(shift.tmp.gear_ratio):-1:2
    shift.acc_dn_index(cpt2) = shift.tmp.gear_ratio(cpt)/shift.tmp.gear_ratio(1);
    cpt2=cpt2+1;
end
clear cpt cpt2
shift.acc_dn_index= [0 shift.acc_dn_index];
shift.acc_dn_index(end+1)=1;

shift.veh_spd_upshift_index(1)=0;

shift.veh_spd_upshift_index(2)=max((eng.spd_idle)*wheel.radius/fd.ratio/...
fd.trc_ratio./shift.tmp.gear_ratio(2) ,(shift.tmp.trs_eng_spd_veh_start+50)...
*wheel.radius/fd.ratio/fd.trc_ratio./shift.tmp.gear_ratio(1));

cpt2 = 1;
for cpt=3:(length(shift.tmp.gear_ratio)+2)
    shift.veh_spd_upshift_index(cpt) = max(shift.tmp.veh_spd_op_max(cpt2),...
    shift.veh_spd_upshift_index(cpt-1)+2);
    cpt2 = cpt2 + 1;
end
clear cpt cpt2

shift.veh_spd_dnshift_index(1)=0;
for cpt=2:(length(shift.tmp.gear_ratio)+1)
    shift.veh_spd_dnshift_index(cpt) = shift.tmp.veh_spd_op_min(cpt-1);
end
clear cpt
shift.veh_spd_dnshift_index(end+1) = 2*shift.tmp.veh_spd_op_min(end)-shift.tmp.veh_spd_op_min(end-1);


shift.tmp.gear_num = 1:length(shift.tmp.gear_ratio);
shift.gear_upshift_map = [];
for cpt = 1:length(shift.acc_up_index)
    if cpt <=2
        shift.gear_upshift_map =[shift.gear_upshift_map [shift.tmp.gear_num...
         shift.tmp.gear_num(end) shift.tmp.gear_num(end)]'];
    else
        shift.gear_upshift_map =[shift.gear_upshift_map [1 shift.tmp.gear_num...
         shift.tmp.gear_num(end)]'];
    end
end
```

```
shift.acc_up_index(cpt)=2; % added to keep gear interpolation from going out of accel range
clear cpt

shift.gear_dnshift_map = [];
for cpt = 1:length(shift.acc_dn_index)
    if cpt >=(length(shift.acc_dn_index)-1)
        shift.gear_dnshift_map =[shift.gear_dnshift_map [1 1 shift.tmp.gear_num]'];
    else
        shift.gear_dnshift_map =[shift.gear_dnshift_map [1 shift.tmp.gear_num...
         shift.tmp.gear_num(end)]'];
    end
end
shift.acc_dn_index(cpt)=2; % added to keep gear interpolation from going out of accel range
clear cpt

shift.gear_upshift_map = shift.gear_upshift_map';
shift.gear_dnshift_map = shift.gear_dnshift_map';

if exist('ptc','var')
    if isfield(shift,'tmp')
        shift = rmfield(shift,'tmp');
    end
end
clear tmp

% This results in a 2-D map for upshifting and one for downshifting
% that is indexed on vehicle accel and speed (dims a x v)


%-------------------------------------------------------------------------%
%%%%%%%%                 Driving Cycle Demands                   %%%%%%%%%
%-------------------------------------------------------------------------%
% Load the appropriate driving cycle - then calculate demands with it
cycle = 'cshvr' ;    % 'cshvr' or 'manhattan' or 'cbd_truck' or 'udds'
load(cycle)         % City Suburban Heavy Vehicle Route

% Define the velocity and position vectors driving cycle imported above
t = sch_cycle(:,1);     % Cycle time counter (seconds)
v = sch_cycle(:,2);     % Cycle velocity profile (m/s)
slope = t-t;  %currently set to zero%          % Cycle slope profile (rad)
dt = t(4)-t(3);         % Time step, in seconds
ttot = length(t);
g = 9.81;               % gravity (m/s^2)
rho = 1.23;             % air density (kg/m^3)

% Parameters, aA, aR, aI, aS used in the equation:
% (can be calculated from parameters in other init files)  %All units SI
% T_wheel = aA*v^2 + aR + aI*a + aS*r             T in N-m
% P_wheel = aA*v^3 + aR*v + aI*v*a + aS*r*v       P in Watts
% Aero coeff - constant
% aA = 0.5*rho*Cd*A
  aA = 0.5*rho*veh.frontal_area*veh.coeff_drag;          % ~3.2
% Rolling coeff as a function of wheel angular velocity
% aR = (CRR_1 + CRR_2*v/r)*m*g*cos(slope)
  aR = (wheel.CRR1 + wheel.CRR2.*v/wheel.radius)*mass*g.*cos(slope);  % ~2000 + ~0.1v
% Intertia coeff - constant
% aI = ~1.1*mass
  aI = 1.1*mass;                                         %~8000
% Slope coeff - can be
% aS = m*g*sin(slope)
  aS = mass*g*sin(slope);

%Initialize:
a=v-v; T_dmd=a; P_dmd=a; gear_conv=a; gear=a; weng=a; wmot=a; ovrspd=a;
```

```
gear_up_next=a; gear_down_next=a; gear_up_weng=a; gear_down_weng=a;

a = diff(v);   % calculate acceleration
a(ttot) = 0;   % keep it the same length

% Driving cycle statistics - just for reference at this point
a_min = min(a); a_max = max(a);
v_min = min(v); v_max = max(v);

% Calculate torque demand at the wheel
T_dmd = (aA*v.^2 + aR + aI.*a + aS)*wheel.radius.*(1-(v==0&a==0));
% term of (1-(v==0&a==0)) ensures no torque demand if a=v=0)
P_dmd = T_dmd.*v/wheel.radius;

% Initialize Gear Shifting Variables:
gear(1:5) = 1;  % Start in first gear
gear_conv(1:5) = wheel.radius/fd.tot_ratio/gb.ratio_map(gear(1)+1);
weng(1:5) = eng.spd_idle;
gear_up_next(1:5) = 1;
gear_up_weng(1:5) = eng.spd_idle;
gear_down_next(1:5) = 1;
gear_down_weng(1:5) = eng.spd_idle;

% Calculate gear choices and engine and motor speeds
for k=5:ttot

  % Calculate upshift or downshift gears, and corresponding eng speed
  gear_up_next(k) = interp2(shift.veh_spd_upshift_index,shift.acc_up_index,...
  shift.gear_upshift_map,v(k),max(0,a(k)),'linear',7);
  gear_up_next(k) = floor(gear_up_next(k));
  gear_up_weng(k) = v(k)/(wheel.radius/fd.tot_ratio/gb.ratio_map(gear_up_next(k)+1));

  gear_down_next(k) = interp2(shift.veh_spd_dnshift_index,shift.acc_dn_index,...
  shift.gear_dnshift_map,v(k),max(0,a(k)),'linear',7);
  gear_down_next(k) = floor(gear_down_next(k));
  gear_down_weng(k) = v(k)/(wheel.radius/fd.tot_ratio/gb.ratio_map(gear_down_next(k)+1));

  % Check to prevent overspeeding or underspeeding the engine
  if (gear(k-1) < 7) && (v(k)/gear_conv(k-1)) > (shift.percent_eng_spd_max*eng.spd_max)
      gear(k) = gear(k-1) + 1;
      ovrspd(k) = 1;
  elseif (gear(k-1) > 1) && (v(k)/gear_conv(k-1)) < (shift.percent_eng_spd_idle*eng.spd_idle)
      gear(k) = gear(k-1) - 1;
      ovrspd(k) = 1;
  % Smoothing of gear shifting - to prevent excessive shifting we only
  % allow a shift every 3 seconds
  elseif (gear(k-3) ~= gear(k-1)) && (ovrspd(k) == 0)
      gear(k) = gear(k-1);
      ovrspd(k) = 1;
  % Shift if that command is acceptable and doesnt cause overspeed
  elseif gear_up_next(k) > gear(k-1) && gear_up_weng(k) > (shift.percent_eng_spd_idle...
  *eng.spd_idle) && ovrspd(k)==0;
      gear(k) = gear_up_next(k);
  elseif gear_down_next(k) < gear(k-1) && gear_down_weng(k) < (shift.percent_eng_spd_max...
  *eng.spd_max) && ovrspd(k)==0;
      gear(k) = gear_down_next(k);
  else % if none of these cases have required a change in gear, stay in gear
      gear(k) = gear(k-1);
  end

   % Calculate engine and motor speeds
    gear_conv(k) = wheel.radius/fd.tot_ratio/gb.ratio_map(gear(k)+1);
    weng(k) = max(eng.spd_idle,v(k)/gear_conv(k));
    wmot(k) = v(k)*tc.ratio/gear_conv(k);
```

```
end
% Modify demands based on gearbox, final drive, and clutch efficiencies:
P_dmd = P_dmd./(gb.eff_map(gear+1)*fd.tot_eff)';
T_dmd = T_dmd./(gb.eff_map(gear+1)*fd.tot_eff)';

%Torque demand at the engine (not including engine losses)
T_dmd_eng = T_dmd.*gear_conv;

P_max = max(P_dmd);

toc;  % outputs simulation time needed to initialize

%--------------------------------------------------------------------------%
%%%%%%%%%%%%%      Dynamic Programming Implementation        %%%%%%%%%%%%%
%--------------------------------------------------------------------------%
% State variable here is SOC... based on voltage: SOC = CV/CVmax = V/Vmax
% The DP loop calculates using an engine on/off cost, which allows the more
% realistic simulation (without the rapid on/off "bang-bang" cycles that
% would result otherwise.
% After the DP loop, the extra time of engine idling is removed to further
% enforce the minimum of 5 seconds on and off cycling.

% Known power demand, velocities, acceleration, and vehicle parameters
clear k h j l
% Initialize variables:
n = 101;                  % Discretization steps for engine torque and SOC
DP.SOC = linspace(ess.soc_min,ess.soc_max,n);          % possible pack SOCs
% Also constrains SOC to given limits from init file
DP.largecost = 100;            % large cost value for m_dot (kg/s)
DP.eng_on_off_cost = 15e-4;    % Engine on/off cost value
kmax = ttot;                   % steps for timer loop - whole cycle
hmax = n; jmax = hmax;         % other loop steps (k (h (j ) ) )

% Initialize loop variables for speed:
DP.costmin=DP.largecost*ones(hmax,kmax); DP.costmin(:,kmax)=0;
DP.Tengmin=zeros(hmax,kmax); DP.SOCdotmin=DP.Tengmin;
DP.SOCnextmin=ones(hmax,kmax); DP.Tmotmin=DP.Tengmin;
DP.Tbrakemin=DP.Tengmin; DP.mdotmin=DP.Tengmin; DP.EngineOn=DP.Tengmin;
% Initialize optimal costs, control vectors
Tengopt=zeros(1,kmax);SOCdotopt=Tengopt;
SOCnextopt=Tengopt;SOCopt=Tengopt;Tmotopt=Tengopt;mdotopt=Tengopt;

% Assumptions:
ess.pack_cap = 3000/ess.num_cell_series;    % Assumed constant pack capacitance

% Constraints:
% Engine torque, constrained at each time step, where
% Possible values for Teng calculated within loop below at each time
cstr.Teng_max = min(interp1q(eng.spd_max_hot_index',eng.trq_max_hot_map',weng),...
max(eng.trq_fuel_hot_index));
      % min fcn prevents NAN's for being outside of mdot table range
% Engine torque losses
cstr.Teng_loss = eng.pwr_loss./weng + interp1q(eng.spd_zero_fuel_hot_index',...
-eng.trq_zero_fuel_hot_index',weng);

% Max/min motor torque - symmetric (also motor power constraint)
cstr.Tmot_max_min = interp1q(mc.spd_max_index',mc.trq_max_map',wmot);
% Max regen torque calculated in loop...
% Motor / ESS current:
cstr.mot_curr = min(mc.curr_max,ess.curr_chg_max);

% Constraint on UC power capability, indexed for each SOC
cstr.Puc_max = (DP.SOC*ess.pack_vmax).^2/(4*ess.pack_res);
```

```
cstr.Puc_max(1) = 0;   % No positive current at low SOC
cstr.Puc_min = ess.pack_vmax_surge*(DP.SOC*ess.pack_vmax - ess.pack_vmax_surge)/ess.pack_res;
cstr.Puc_min(n) = 0;   % No negative current at high SOC

% UC SOC constraints
cstr.SOC_initial = 0.8;
cstr.SOC_min = ess.soc_min;
cstr.SOC_max = ess.soc_max;

% Calculations inside DP loop start with Input
% Engine Torque/Speed (gives fuel rate) -> Road Torque demand ->
% Motor torque -> ESS power -> UC SOC change -> Next SOC

for k=kmax-1:-1:1

  % Possible Torques for engine at each given speed (constrains Teng)
    DP.Teng = [0 linspace(0,cstr.Teng_max(k)-cstr.Teng_loss(k),n-1)+cstr.Teng_loss(k)];
  % Fuel Rates for each engine torque - same for all SOC's
    DP.mdot = interp2(eng.trq_fuel_hot_index,eng.spd_fuel_hot_index,eng.fuel_hot_map,DP.Teng,weng(k));
    DP.mdot(1) = 0;  % engine off - mdot = 0
  % Max Regen torque (indexed over SOC)
    cstr.Tmot_max_regen = max(-(mc.curr_max^2*ess.pack_res+...
        mc.curr_max*DP.SOC*ess.pack_vmax),-cstr.Tmot_max_min(k));

  for h=hmax:-1:1            % Indexes possible states(SOC) at (k-1)th timestep
 % Initialize Cost = 0 - modified if constraint violated, then mdot and
 % other costs added at the end
   DP.cost = zeros(1,n);

 % Calculate Motor Torque - max/min prevents extra torque from being demanded
   DP.Tmot = (T_dmd_eng(k) - (DP.Teng -(DP.Teng>0).*cstr.Teng_loss(k)).*cpl.eff)./tc.ratio;

 % Max regen condition
   if T_dmd_eng(k)/tc.ratio < -cstr.Tmot_max_min(k)
       DP.Tmot(1) = max(cstr.Tmot_max_regen(h),cstr.Puc_min(h)/wmot(k));
       % max regen torque value possible for given SOC... approximate
       DP.cost(1) = 0;
   end

 % Constraints on Motor Torque/Power:
   if h==hmax
      DP.cost(DP.Tmot < 0) = DP.largecost;
      DP.Tmot(DP.Tmot < 0) = 0;
   elseif h==1
      DP.cost(DP.Tmot > 0) = DP.largecost;
      DP.Tmot(DP.Tmot > 0) = 0;
   end
   DP.cost(DP.Tmot > cstr.Tmot_max_min(k)) = DP.largecost;
   DP.Tmot(DP.Tmot > cstr.Tmot_max_min(k)) = cstr.Tmot_max_min(k);
   DP.cost(DP.Tmot < -cstr.Tmot_max_min(k)) = DP.largecost;
   DP.Tmot(DP.Tmot < -cstr.Tmot_max_min(k)) = -cstr.Tmot_max_min(k);


% ESS Electrical Power lookup - from motor data
% (can add a static or time-varying load to this for electrical power loss)
   DP.Puc = interp2(mc.trq_eff_index,mc.spd_eff_index,mc.pwr_elec_map,DP.Tmot,wmot(k));

% Ultracapacitor power constraints
% Max power UC discharge:
   DP.cost(DP.Puc > cstr.Puc_max(h)) = DP.largecost;
   DP.Puc(DP.Puc > cstr.Puc_max(h)) = cstr.Puc_max(h);
   DP.Tmot(DP.Puc > cstr.Puc_max(h)) = 0;

% Max power UC charging:
```

```
   DP.cost(DP.Puc < cstr.Puc_min(h)) = DP.largecost;
   DP.Puc(DP.Puc < cstr.Puc_min(h)) = cstr.Puc_min(h);
   DP.Tmot(DP.Puc < cstr.Puc_min(h)) = 0;

% ESS current out is negative, so into UC is positive
   DP.Iuc = (-DP.SOC(h)*ess.pack_vmax + sqrt((DP.SOC(h)*ess.pack_vmax)^2 ...
       - 4*ess.pack_res*DP.Puc)) / (2*ess.pack_res);

%  Motor current draw constraint
   DP.cost(abs(DP.Iuc) >= cstr.mot_curr) = DP.largecost;
   DP.Iuc(DP.Iuc > cstr.mot_curr) = cstr.mot_curr;
   DP.Iuc(DP.Iuc < -cstr.mot_curr) = -cstr.mot_curr;

% Charging efficiency consideration - from Maxwell model
   DP.Iuc(DP.Iuc > 0) = DP.Iuc(DP.Iuc > 0)*ess.chg_eff;

% SOC change
   DP.SOCdot = DP.Iuc./(ess.pack_cap*ess.pack_vmax);

% Next SOC - to help check constraints
   DP.SOCnext = DP.SOC(h) + DP.SOCdot*dt;

% Ultracapacitor SOC constraints
   DP.cost(DP.SOCnext > cstr.SOC_max) = DP.largecost;
   DP.SOCdot(DP.SOCnext > cstr.SOC_max) = (ess.soc_max - DP.SOC(h))/dt;
   DP.Iuc(DP.SOCnext > cstr.SOC_max) = DP.SOCdot(DP.SOCnext > cstr.SOC_max).*...
   (ess.pack_cap.*ess.pack_vmax);
   DP.Puc(DP.SOCnext > cstr.SOC_max) = (DP.Iuc(DP.SOCnext > cstr.SOC_max).^2).*...
   ess.pack_res+ DP.Iuc(DP.SOCnext > cstr.SOC_max).*DP.SOC(h).*ess.pack_vmax ;
   DP.Tmot(DP.SOCnext > cstr.SOC_max) = DP.Puc(DP.SOCnext > cstr.SOC_max)./wmot(k);
   DP.SOCnext(DP.SOCnext > cstr.SOC_max) = ess.soc_max;

   DP.cost(DP.SOCnext < cstr.SOC_min) = DP.largecost;
   DP.SOCdot(DP.SOCnext < cstr.SOC_min) = (ess.soc_min - DP.SOC(h))/dt;
   DP.Iuc(DP.SOCnext < cstr.SOC_min) = DP.SOCdot(DP.SOCnext < cstr.SOC_min).*...
   (ess.pack_cap.*ess.pack_vmax);
   DP.Puc(DP.SOCnext < cstr.SOC_min) = (DP.Iuc(DP.SOCnext < cstr.SOC_min).^2).*ess.pack_res+...
          DP.Iuc(DP.SOCnext < cstr.SOC_min).*DP.SOC(h).*ess.pack_vmax ;
   DP.Tmot(DP.SOCnext < cstr.SOC_min) = DP.Puc(DP.SOCnext < cstr.SOC_min)./wmot(k);
   DP.SOCnext(DP.SOCnext < cstr.SOC_min) = ess.soc_min;

% Braking torque - only to catch the rest of the negative torque
 % no engine on while braking
 if T_dmd_eng(k) < 0
   DP.Tbrake = T_dmd_eng(k) - DP.Tmot.*tc.ratio - (DP.Teng - (DP.Teng>0)*...
   cstr.Teng_loss(k)).*cpl.eff;
   DP.cost(DP.Tbrake > 0.1) = DP.largecost;
   DP.cost(DP.Tmot>0) = DP.largecost;
 else
   DP.Tbrake = zeros(1,n);
 end

% Torque balance constraint - did we constrain to not meet the cycle demands?:
DP.Trq_error = T_dmd_eng(k) - DP.Tbrake - DP.Tmot*tc.ratio - (DP.Teng -...
 cstr.Teng_loss(k).*(DP.Teng>0)).*cpl.eff;

DP.cost(abs(DP.Trq_error) > 0.1) = DP.largecost;

% If all constraints pass, the cost is the unmodified lookup-tabled fuel
% rate for that choice of Teng

% If all engine torques end up NOT meeting constraints, try and select the
% one that MINIMIZES the torque demand error
DP.Trq_error_min = min(DP.Trq_error);
```

```
if length(nonzeros(DP.cost==DP.largecost))==n
DP.cost(abs(DP.Trq_error)>abs(DP.Trq_error_min+1)) = 2*DP.largecost;
end

%-------------------------------------------------------------------------%
%    for j=jmax:-1:1         % Indexes possible engine torques we can apply
%
%           % all commands removed to speed up program
%
%    end                     % of j counter for torques
%-------------------------------------------------------------------------%

% Define cost to go for each j point as well, and pick the min of those!
   DP.nextcost = interp1(DP.SOC',DP.costmin(:,k+1),real(DP.SOCnext'))';
   DP.next_eng_on = round(interp1(DP.SOC',DP.EngineOn(:,k+1),real(DP.SOCnext')))';
   if T_dmd_eng(k) == 0
     DP.cost_eng_on_off = 0;
   else
     DP.cost_eng_on_off = DP.eng_on_off_cost.*((DP.Teng>0)~=DP.next_eng_on);
   end

   % Add fuel rate and engine on/off cost into costs
   DP.cost = DP.cost + DP.mdot + DP.cost_eng_on_off;
   % Add next costs to get cost to go
   DP.costtogo = DP.cost + DP.nextcost; % Total cost for each control decision

   DP.costtogomin = min(DP.costtogo);
   [row,col] = find(DP.costtogo==DP.costtogomin);
   if length(row)>1,  row=1; col=min(col);  % chooses lowest engine torque
   end
   % Store 'Optimal' values... for variables of concern
   DP.costmin(h,k) = DP.costtogomin;        % Cost-to-go at h,k
   DP.Tengmin(h,k) = DP.Teng(col);          % Engine torque input
   DP.SOCdotmin(h,k) = DP.SOCdot(col);      % Rate of change of SOC
   DP.SOCnextmin(h,k) = DP.SOCnext(col);    % Next SOC state
   DP.Tmotmin(h,k) = DP.Tmot(col);          % Motor torque
   DP.Tbrakemin(h,k) = DP.Tbrake(col);      % Braking torque
   DP.mdotmin(h,k) = DP.mdot(col);          % Fuel rate
   DP.EngineOn(h,k) = DP.Teng(col)>0;       % Engine on/off

% Can add other outputs here as well

  end % of h counter for each SOC possible

end % of k counter for each time step

% Start the forward lookup of the Optimal Cost Path:

% Initialize optimal costs, control vectors
Tengopt=zeros(1,kmax);SOCdotopt=Tengopt;
SOCnextopt=Tengopt;SOCopt=Tengopt;Tmotopt=Tengopt;
Tbrakeopt=Tengopt;mdotopt=Tengopt;

% If using initial SOC constraint:
SOCopt(1) = cstr.SOC_initial;

% If not using initial SOC constraint:
% mininitcost = min(DP.costmin(:,1));
% [row,col] = find(DP.costmin(:,1)==mininitcost);
% SOCopt(1) = DP.SOC(row);

Tengopt(1) = interp1(DP.SOC,DP.Tengmin(:,1),SOCopt(1));
SOCdotopt(1) = interp1(DP.SOC,DP.SOCdotmin(:,1),SOCopt(1));
```

```
SOCnextopt(1) = interp1(DP.SOC,DP.SOCnextmin(:,1),SOCopt(1));
Tmotopt(1) = interp1(DP.SOC,DP.Tmotmin(:,1),SOCopt(1));
Tbrakeopt(1) = interp1(DP.SOC,DP.Tbrakemin(:,1),SOCopt(1));
mdotopt(1) = interp1(DP.SOC,DP.mdotmin(:,1),SOCopt(1));

for k=2:kmax
    if k==kmax
        SOCopt(k)=SOCnextopt(k-1);
        Tengopt(k) = 0;
        SOCdotopt(k) = 0;
        SOCnextopt(k) = 0;
        Tmotopt(k) = 0;
        Tbrakeopt(k) = 0;
        mdotopt(k) = 0;
    else
        SOCopt(k) = SOCnextopt(k-1);
        Tengopt(k) = interp1(DP.SOC,DP.Tengmin(:,k),SOCopt(k),'nearest');
        SOCdotopt(k) = interp1(DP.SOC,DP.SOCdotmin(:,k),SOCopt(k),'nearest');
        SOCnextopt(k) = interp1(DP.SOC,DP.SOCnextmin(:,k),SOCopt(k),'nearest');
        Tmotopt(k) = interp1(DP.SOC,DP.Tmotmin(:,k),SOCopt(k),'nearest');
        Tbrakeopt(k) = interp1(DP.SOC,DP.Tbrakemin(:,k),SOCopt(k),'nearest');
        mdotopt(k) = interp1(DP.SOC,DP.mdotmin(:,k),SOCopt(k),'nearest');
    end
end

% Engine idle selection - to ensure a minimum of 5 second on/off cycling
for k=6:kmax-5
  if sum(Tengopt(k-5:k-1) > 0)==5 && sum(Tengopt(k:k+4)- cstr.Teng_loss(k:k+4)' < 0.1)==5
     Tengopt(k:k+4) = 0 ;
     mdotopt(k:k+4) = 0 ;
  elseif sum(Tengopt(k+1:k+5) > 0)==5 && sum(Tengopt(k-4:k) - cstr.Teng_loss(k-4:k)' < 0.1)==5
     Tengopt(k-4:k) = 0 ;
     mdotopt(k-4:k) = 0 ;
  end
end

TorqueError = T_dmd_eng' - Tbrakeopt - Tmotopt*tc.ratio - Tengopt + cstr.Teng_loss'.*(Tengopt>1);

simutime=toc;  % stores the final simulation time

% Calculate fuel economy
cycle_length_m = sum(v.*dt);
fuel_mass_kg = sum(mdotopt*dt);
fuel_vol_l = fuel_mass_kg/eng.fuel_density_val;
fuel_vol_gal = fuel_vol_l/3.78541178;

cycle_length_mi = cycle_length_m/1000*0.621371192;

FuelEconomy = cycle_length_mi/fuel_vol_gal;
% Baseline mpgs - from DP baseline sims
switch lower(cycle)
  case('cshvr')
    baseline_FE = 6.945;
  case('manhattan')
    baseline_FE = 4.390;
  case('cbd_truck')
    baseline_FE = 5.105;
  case('udds')
    baseline_FE = 7.572;
end
percent_improvement = (FuelEconomy-baseline_FE)/baseline_FE*100;

% Statistics on driving cycle:
time_engoff = length(nonzeros(Tengopt==0));
```

```
time_regenpower = length(nonzeros(P_dmd<=0));

% Percent of Regen Braking energy recovered
brake_energy_total = sum(T_dmd_eng.*(T_dmd_eng<0).*weng.*dt);
brake_energy_captured = sum((Tmotopt*tc.ratio+Tengopt).*(Tmotopt<0).*(T_dmd_eng'<0).*weng'.*dt);
percent_brake_energy = brake_energy_captured/brake_energy_total*100 ;

% Torque error
trq_balance = T_dmd_eng' - Tbrakeopt - Tmotopt*tc.ratio - (Tengopt - cstr.Teng_loss'.*...
(Tengopt>1)).*cpl.eff;
trq_balance_error = sum(trq_balance);

% Times the cycle demands were not able to be met
error_number = sum(abs(trq_balance)>0.1);

% Display values for the simulation:
fprintf('\n   DP Simulation Parameters:\n')
fprintf(' The Simulation has been run on the %s cycle\n',cycle)
fprintf(' with a resolution of n = %5.0f samples \n',n)
fprintf(' and a simulation time of %6.1f seconds\n', simutime)

fprintf(' The Fuel economy is %6.3f mpg \n',FuelEconomy)
fprintf(' For an improvement of %5.2f %% \n',percent_improvement)
fprintf(' The engine remained off for %6.1f seconds\n', time_engoff)
fprintf(' The motor can handle %4.1f %% of the braking energy \n',percent_brake_energy)
fprintf(' There are %2.0f ''errors'' - points the vehicle \n did not meet the cycle...
 demands\n',error_number)
fprintf(' for a total torque balance error of %6.1e N-m. \n',trq_balance_error)

fprintf('\n Vehicle Parameters: \n')
fprintf(' Vehicle Mass = %6.0f kg \n',mass)
fprintf(' Torque converter ratio = %4.2f \n',tc.ratio)

fprintf('\n Ultracapacitor Pack Parameters: \n')
fprintf(' Initial SOC = %5.2f \n',SOCopt(1))
fprintf(' Cells in Series = %4.0f \n', ess.num_cell_series)
fprintf(' Max Voltage = %6.2f Volts\n', ess.pack_vmax)
fprintf(' Min Voltage = %6.2f Volts\n', ess.pack_vmin)
fprintf(' Capacitance = %6.2f Farads\n', ess.pack_cap(1))
fprintf(' Resistance = %6.4f Ohms \n', ess.pack_res)
```

D.2 Model Predictive Control Coding

The coding used for the MPC implementation is included here. It runs with the same initialization and vehicle demands calculations as the DP coding, so those have been omitted for brevity.

```
%-------------------------------------------------------------------------%
%%%%%%%%%%%%%    Model Predictive Control Implementation     %%%%%%%%%%%%%
%-------------------------------------------------------------------------%
% State variable here is SOC... based on voltage: SOC = CV/CVmax = V/Vmax
```

```
% MPC method proceeds forward through the cycle, and calculates a "DP loop"
% at each time step - trying to minimize the cost over this short time,
% then executes the first of these controls and proceeds to the next step.

% Known power demand, velocities, acceleration, and vehicle parameters
clear k h j l
q = waitbar(0,'MPC Simulation Progress');
% Initialize variables:
n = 101;         % Discretization steps for engine torque and SOC (101)
nc = 1;          % Control horizon (unused)
np = 3;          % Prediction horizon (used) = 1780 for DP comparison
% can be up to 55 w/o affecting zero-dmd tail of cycle

MPC.SOC = linspace(ess.soc_min,ess.soc_max,n);          % possible pack SOCs
% Also constrains SOC to given limits from init file
MPC.largecost = 100;            % large cost value
MPC.eng_on_off_cost = 15e-4;    % Engine on/off cost value
load data_DP_SOC_costs SOC_cost_one
% Decay function for SOC cost that decreases as horizon length increases
SOC_curve = [0 0 8e-3 5.83e-3  4.5e-3  3.5e-3  3.2e-3  2.8e-3  2.4e-3...
    2.1e-3  1.9e-3  1.8e-3  1.7e-3  1.6e-3  1.5e-3  1.4e-3...
    1.013e-3  1.005e-3  9.97e-4  9.91e-4  9.857e-4  9.812e-4  9.774e-4 ...
    9.74e-4  9.71e-4  9.68e-4  9.663e-4  9.64e-4  9.62e-4  0];
MPC.SOC_cost = SOC_cost_one*SOC_curve(np);    % Low SOC cost

% Import decay prediction parameters:
load data_decay_rates_cshvr lambda_vals torque_vals
lambda = zeros(1,ttot);

kmax = ttot;                    % steps for timer loop - whole cycle
hmax = n; jmax = hmax;          % other loop steps (k (h (j ) ) )

% Initialize loop variables for speed:
MPC.costmin=MPC.largecost*ones(hmax,np+1); MPC.costmin(:,np+1)=0;
MPC.Tengmin=zeros(hmax,np+1); MPC.SOCdotmin=MPC.Tengmin;
MPC.SOCnextmin=ones(hmax,np+1); MPC.Tmotmin=MPC.Tengmin;
MPC.Tbrakemin=MPC.Tengmin; MPC.mdotmin=MPC.Tengmin; MPC.EngineOn=MPC.Tengmin;
% Initialize optimal costs, control vectors
Tengopt=zeros(1,kmax);SOCdotopt=Tengopt;SOCnextopt=Tengopt;SOCopt=Tengopt;
Tmotopt=Tengopt;Tbrakeopt=Tengopt;mdotopt=Tengopt;EngOnopt=Tengopt;Costopt=Tengopt;
% Initialize the engine "off" control signal
EngOff_Sig=Tengopt; EngOff_Sig(1:3)=1; % Engine starts "off"

% Assumptions:
ess.pack_cap = 3000/ess.num_cell_series;    % Assumed constant pack capacitance

% Constraints:
% Engine torque, constrained at each time step, where
% Possible values for Teng calculated within loop below at each time
cstr.Teng_max = min(interp1q(eng.spd_max_hot_index',eng.trq_max_hot_map',weng),...
max(eng.trq_fuel_hot_index));
      % min fcn prevents NAN's for being outside of mdot table range
% Engine torque losses
cstr.Teng_loss = eng.pwr_loss./weng + interp1q(eng.spd_zero_fuel_hot_index',...
-eng.trq_zero_fuel_hot_index',weng);

% Max/min motor torque - symmetric (also motor power constraint)
cstr.Tmot_max_min = interp1q(mc.spd_max_index',mc.trq_max_map',wmot);
% Max regen torque calculated in loop...
% Motor / ESS current:
cstr.mot_curr = min(mc.curr_max,ess.curr_chg_max);

% Constraint on UC power capability, indexed for each SOC
cstr.Puc_max = (MPC.SOC*ess.pack_vmax).^2/(4*ess.pack_res);
```

```
cstr.Puc_max(1) = 0;   % No positive current at low SOC
cstr.Puc_min = ess.pack_vmax_surge*(MPC.SOC*ess.pack_vmax - ess.pack_vmax_surge)/ess.pack_res;
cstr.Puc_min(n) = 0;   % No negative current at high SOC

% UC SOC constraints
cstr.SOC_initial = 0.8;
cstr.SOC_min = ess.soc_min;
cstr.SOC_max = ess.soc_max;

% Calculations inside MPC loop start with Input
% Engine Torque/Speed (gives fuel rate) -> Road Torque demand ->
% Motor torque -> ESS power -> UC SOC change -> Next SOC

for k=1:kmax-np  % advance forward through the cycle
    waitbar(k/kmax)

  % Choose a value for time constant(=1/lambda) based on the value of Torque
  if T_dmd(k) > torque_vals(1)
      lambda(k)= lambda_vals(1);
  else
    for kp = 1:length(lambda_vals)-1
      if T_dmd(k)< torque_vals(kp) && T_dmd(k) > torque_vals(kp+1)
         lambda(k) = lambda_vals(kp+1);
         break
      end
    end
  end

% Generate predicted future torques (first term is actual torque)
  MPC.Tpred = T_dmd_eng(k)*exp(((1:np)-1)*lambda(k));

  for m = np:-1:1  % for each time do receding horizon DP

      % Possible Torques for engine at each given speed (constrains Teng)
    MPC.Teng = [0 linspace(0,cstr.Teng_max(k+m-1)-cstr.Teng_loss(k+m-1),n-1)+...
    cstr.Teng_loss(k+m-1)];
  % Fuel Rates for each engine torque - same for all SOC's
    MPC.mdot = interp2(eng.trq_fuel_hot_index,eng.spd_fuel_hot_index,...
    eng.fuel_hot_map,MPC.Teng,weng(k+m-1));
    MPC.mdot(1) = 0;  % engine off - mdot = 0
  % Max Regen torque (indexed over SOC)
    cstr.Tmot_max_regen = max(-(mc.curr_max^2*ess.pack_res+...
        mc.curr_max*MPC.SOC*ess.pack_vmax),-cstr.Tmot_max_min(k+m-1));

  for h=hmax:-1:1            % Indexes possible states(SOC) at (k-1)th timestep
 % Initialize Cost = 0 - modified if constraint violated, then mdot and
 % other costs added at the end
   MPC.cost = zeros(1,n);

 % Calculate Motor Torque - max/min prevents extra torque from being demanded
   MPC.Tmot = (MPC.Tpred(m) - (MPC.Teng -(MPC.Teng>0).*cstr.Teng_loss(k+m-1)).*cpl.eff)./tc.ratio;

 % Max regen condition
   if MPC.Tpred(m)/tc.ratio < -cstr.Tmot_max_min(k+m-1)
       MPC.Tmot(1) = max(cstr.Tmot_max_regen(h),cstr.Puc_min(h)/wmot(k+m-1));
       % max regen torque value possible for given SOC... approximate
       MPC.cost(1) = 0;
   end

 % Constraints on Motor Torque/Power:
   if h==hmax
      MPC.cost(MPC.Tmot < 0) = MPC.largecost;
      MPC.Tmot(MPC.Tmot < 0) = 0;
   elseif h==1
```

```
      MPC.cost(MPC.Tmot > 0) = MPC.largecost;
      MPC.Tmot(MPC.Tmot > 0) = 0;
   end
   MPC.cost(MPC.Tmot > cstr.Tmot_max_min(k+m-1)) = MPC.largecost;
   MPC.Tmot(MPC.Tmot > cstr.Tmot_max_min(k+m-1)) = cstr.Tmot_max_min(k+m-1);
   MPC.cost(MPC.Tmot < -cstr.Tmot_max_min(k+m-1)) = MPC.largecost;
   MPC.Tmot(MPC.Tmot < -cstr.Tmot_max_min(k+m-1)) = -cstr.Tmot_max_min(k+m-1);


% ESS Electrical Power lookup - from motor data
% (can add a static or time-varying load to this for electrical power loss)
   MPC.Puc = interp2(mc.trq_eff_index,mc.spd_eff_index,mc.pwr_elec_map,MPC.Tmot,wmot(k+m-1));

% Ultracapacitor power constraints
% Max power UC discharge:
   MPC.cost(MPC.Puc > cstr.Puc_max(h)) = MPC.largecost;
   MPC.Puc(MPC.Puc > cstr.Puc_max(h)) = cstr.Puc_max(h);
   MPC.Tmot(MPC.Puc > cstr.Puc_max(h)) = 0;

% Max power UC charging:
   MPC.cost(MPC.Puc < cstr.Puc_min(h)) = MPC.largecost;
   MPC.Puc(MPC.Puc < cstr.Puc_min(h)) = cstr.Puc_min(h);
   MPC.Tmot(MPC.Puc < cstr.Puc_min(h)) = 0;

% ESS current out is negative, so into UC is positive
   MPC.Iuc = (-MPC.SOC(h)*ess.pack_vmax + sqrt((MPC.SOC(h)*ess.pack_vmax)^2 ...
       - 4*ess.pack_res*MPC.Puc)) / (2*ess.pack_res);

%  Motor current draw constraint
   MPC.cost(abs(MPC.Iuc) >= cstr.mot_curr) = MPC.largecost;
   MPC.Iuc(MPC.Iuc > cstr.mot_curr) = cstr.mot_curr;
   MPC.Iuc(MPC.Iuc < -cstr.mot_curr) = -cstr.mot_curr;

% Charging efficiency consideration - from Maxwell model
   MPC.Iuc(MPC.Iuc > 0) = MPC.Iuc(MPC.Iuc > 0)*ess.chg_eff;

% SOC change
   MPC.SOCdot = MPC.Iuc./(ess.pack_cap*ess.pack_vmax);

% Next SOC - to help check constraints
   MPC.SOCnext = MPC.SOC(h) + MPC.SOCdot*dt;

% Ultracapacitor SOC constraints
   MPC.cost(MPC.SOCnext > cstr.SOC_max) = MPC.largecost;
   MPC.SOCdot(MPC.SOCnext > cstr.SOC_max) = (ess.soc_max - MPC.SOC(h))/dt;
   MPC.Iuc(MPC.SOCnext > cstr.SOC_max) = MPC.SOCdot(MPC.SOCnext > cstr.SOC_max).*...
   (ess.pack_cap.*ess.pack_vmax);
   MPC.Puc(MPC.SOCnext > cstr.SOC_max) = (MPC.Iuc(MPC.SOCnext > cstr.SOC_max).^2).*...
   ess.pack_res+ MPC.Iuc(MPC.SOCnext > cstr.SOC_max).*MPC.SOC(h).*ess.pack_vmax ;
   MPC.Tmot(MPC.SOCnext > cstr.SOC_max) = MPC.Puc(MPC.SOCnext > cstr.SOC_max)./wmot(k+m-1);
   MPC.SOCnext(MPC.SOCnext > cstr.SOC_max) = ess.soc_max;

   MPC.cost(MPC.SOCnext < cstr.SOC_min) = MPC.largecost;
   MPC.SOCdot(MPC.SOCnext < cstr.SOC_min) = (ess.soc_min - MPC.SOC(h))/dt;
   MPC.Iuc(MPC.SOCnext < cstr.SOC_min) = MPC.SOCdot(MPC.SOCnext < cstr.SOC_min).*...
   (ess.pack_cap.*ess.pack_vmax);
   MPC.Puc(MPC.SOCnext < cstr.SOC_min) = (MPC.Iuc(MPC.SOCnext < cstr.SOC_min).^2).*...
   ess.pack_res+ MPC.Iuc(MPC.SOCnext < cstr.SOC_min).*MPC.SOC(h).*ess.pack_vmax ;
   MPC.Tmot(MPC.SOCnext < cstr.SOC_min) = MPC.Puc(MPC.SOCnext < cstr.SOC_min)./wmot(k+m-1);
   MPC.SOCnext(MPC.SOCnext < cstr.SOC_min) = ess.soc_min;

% Braking torque - only to catch the rest of the negative torque
% no positive motor torque while braking
if MPC.Tpred(m) < 0
```

```
   MPC.Tbrake = MPC.Tpred(m) - MPC.Tmot.*tc.ratio -(MPC.Teng -(MPC.Teng>0)*...
   cstr.Teng_loss(k+m-1)).*cpl.eff;
   MPC.cost(MPC.Tbrake > 0.1) = MPC.largecost;
   MPC.cost(MPC.Tmot>0) = MPC.largecost;
else
   MPC.Tbrake = zeros(1,n);
end

% Torque balance constraint - did we constrain to not meet the cycle demands?:
MPC.Trq_error = MPC.Tpred(m) - MPC.Tbrake - MPC.Tmot*tc.ratio - (MPC.Teng - ...
cstr.Teng_loss(k+m-1).*(MPC.Teng>0)).*cpl.eff;

MPC.cost(abs(MPC.Trq_error) > 0.1) = MPC.largecost;

% If all constraints pass, the cost is the unmodified lookup-tabled fuel
% rate for that choice of Teng

% If all engine torques end up NOT meeting constraints, try and select the
% one that MINIMIZES the torque demand error
MPC.Trq_error_min = min(MPC.Trq_error);

if length(nonzeros(MPC.cost==MPC.largecost))==n
MPC.cost(abs(MPC.Trq_error)>abs(MPC.Trq_error_min+1)) = 2*MPC.largecost;
end

%-------------------------------------------------------------------------%
%    for j=jmax:-1:1         % Indexes possible engine torques we can apply
%
%           % all commands removed to speed up program
%
%    end                     % of j counter for torques
%-------------------------------------------------------------------------%

% Define cost to go for each j point as well, and pick the min of those!
   MPC.nextcost = interp1(MPC.SOC',MPC.costmin(:,m+1),real(MPC.SOCnext'))';
% Engine on/off cost:
   MPC.next_eng_on = round(interp1(MPC.SOC',MPC.EngineOn(:,m+1),real(MPC.SOCnext')))';
   if MPC.Tpred(m) == 0
     MPC.cost_eng_on_off = 0;
   else
     MPC.cost_eng_on_off = MPC.eng_on_off_cost.*((MPC.Teng>0)~=MPC.next_eng_on);
   end
% Eng on/off cost from previous control decision:
   if k > 1 && m == 1
     MPC.prev_eng_on_off = MPC.eng_on_off_cost.*((MPC.Teng>0)~=EngOnopt(k-1));
     MPC.cost_eng_on_off = MPC.cost_eng_on_off + MPC.prev_eng_on_off;
   end
% Engine on/off constraint - to ensure engine idles if not demanded on for
% a long enough time (attempt minimum of 5 second on/off cycles):
MPC.cost_eng_idle = MPC.Teng*0; %Initialize cost
   if k > 5 && m == 1
  MPC.next_next_eng_on = round(interp1(MPC.SOC',MPC.EngineOn(:,m+2),...
  real(interp1(MPC.SOC',MPC.SOCnextmin(:,m+1),real(MPC.SOCnext')))'))';
      % conditions if engine was just on
      if  EngOnopt(k-1)==1 && sum(EngOnopt(k-5:k-1)==1)< 5
      MPC.cost_eng_idle(1) = MPC.largecost; % high cost to turn the engine off
      elseif EngOnopt(k-1)==1 && MPC.next_eng_on(1)==1
      MPC.cost_eng_idle(1) = MPC.largecost;
      elseif EngOnopt(k-1)==1 && MPC.next_next_eng_on(1)==1
      MPC.cost_eng_idle(1) = MPC.largecost;
      % conditions if engine was just off (less than three seconds)
      elseif EngOnopt(k-1)==0 && sum(EngOnopt(k-5:k-1)==0)< 5
      MPC.cost_eng_idle = MPC.largecost*(MPC.Teng>0);
      end
```

```
   end

% SOC cost - to simulate DP performance and help prevent errors due to
% short-sighted prediction horizon
   MPC.SOCnext_cost = interp1(MPC.SOC,MPC.SOC_cost,MPC.SOCnext);

% Add fuel rate and engine on/off cost into costs
   MPC.cost = MPC.cost + MPC.mdot + MPC.cost_eng_on_off + MPC.cost_eng_idle + MPC.SOCnext_cost;
% Add next costs to get cost to go
   MPC.costtogo = MPC.cost + MPC.nextcost; % Total cost for each control decision

   MPC.costtogomin = min(MPC.costtogo);
   [row,col] = find(MPC.costtogo==MPC.costtogomin);
   if length(row)>1,  row=1; col=min(col);  % chooses lowest engine torque
   end
   % Engine on/off modification for Predicted MPC information
   if k>2 && m==1 && col==1 && length(nonzeros(EngOff_Sig(k-2:k-1)==1))<2;
     % If we first request engine off, wait two seconds before we actually turn it off
       MPC.EngineOff(h,m) = 1;
       MPC.costtogomin = min(MPC.costtogo(2:n));
       [row,col] = find(MPC.costtogo==MPC.costtogomin);
    if length(row)>1,  row=1; col=min(col);  % chooses lowest engine torque
    end
   % If we have requested the engine off for the last two seconds, the
   % engine can be turned off if it needs to be
   elseif k>2 && m==1 && col==1 && length(nonzeros(EngOff_Sig(k-2:k-1)==1))==2;
       MPC.EngineOff(h,m) = 1;
   elseif k>2 && m==1
       MPC.EngineOff(h,m) = 0;
   else
       MPC.EngineOff(h,m) = 1;
   end

   % Store 'Optimal' values... for variables of concern
   MPC.costmin(h,m) = MPC.costtogomin;        % Cost-to-go at h,k
   MPC.Tengmin(h,m) = MPC.Teng(col);          % Engine torque input
   MPC.SOCdotmin(h,m) = MPC.SOCdot(col);      % Rate of change of SOC
   MPC.SOCnextmin(h,m) = MPC.SOCnext(col);    % Next SOC state
   MPC.Tmotmin(h,m) = MPC.Tmot(col);          % Motor torque
   MPC.Tbrakemin(h,m) = MPC.Tbrake(col);      % Braking torque
   MPC.mdotmin(h,m) = MPC.mdot(col);          % Fuel rate
   MPC.EngineOn(h,m) = MPC.Teng(col)>0;       % Engine on/off

% Can add other outputs here as well

  end % of h counter for each SOC possible

  end % of m counter for the MPC horizon loop

% Choose the "optimal" control input based on the MPC horizon - and other quantities
if k==1
    SOCopt(k) = cstr.SOC_initial;
else
    SOCopt(k) = SOCnextopt(k-1);
end
    idx=interp1(MPC.SOC,1:n,SOCopt(k),'nearest');
    Costopt(k) = MPC.costmin(idx,1);
    Tengopt(k) = MPC.Tengmin(idx,1);
    SOCdotopt(k) = MPC.SOCdotmin(idx,1);
    SOCnextopt(k) = MPC.SOCnextmin(idx,1);
    Tmotopt(k) = MPC.Tmotmin(idx,1);
    Tbrakeopt(k) = MPC.Tbrakemin(idx,1);
    mdotopt(k) = MPC.mdotmin(idx,1);
    EngOnopt(k) = MPC.EngineOn(idx,1);
```

```
    EngOff_Sig(k) = MPC.EngineOff(idx,1);

end % of k counter for each time step

close(q)    % close waitbar

% Demand steady control values for the rest of the cycle
Tengopt(k+1:kmax) = 0;
SOCdotopt(k+1:kmax) = 0;
SOCnextopt(k+1:kmax) = SOCopt(k);
SOCopt(k+1:kmax) = SOCopt(k);
Tmotopt(k+1:kmax) = 0;
Tbrakeopt(k+1:kmax) = 0;
mdotopt(k+1:kmax) = 0;
EngOnopt(k+1:kmax) = 0;

simutime=toc;  % stores the final simulation time

% Calculate fuel economy
cycle_length_m = sum(v.*dt);
fuel_mass_kg = sum(mdotopt*dt);
fuel_vol_l = fuel_mass_kg/eng.fuel_density_val;
fuel_vol_gal = fuel_vol_l/3.78541178;

cycle_length_mi = cycle_length_m/1000*0.621371192;

FuelEconomy = cycle_length_mi/fuel_vol_gal;
% Baseline mpgs - from DP baseline sims
switch lower(cycle)
  case('cshvr')
    baseline_FE = 6.945;
  case('manhattan')
    baseline_FE = 4.390;
  case('cbd_truck')
    baseline_FE = 5.105;
  case('udds')
    baseline_FE = 7.572;
end
percent_improvement = (FuelEconomy-baseline_FE)/baseline_FE*100;

% Statistics on driving cycle:
time_engoff = length(nonzeros(Tengopt==0));
time_regenpower = length(nonzeros(P_dmd<=0));

% Percent of Regen Braking energy recovered
brake_energy_total = sum(T_dmd_eng.*(T_dmd_eng<0).*weng.*dt);
brake_energy_captured = sum((Tmotopt*tc.ratio+Tengopt).*(Tmotopt<0).*...
(T_dmd_eng'<0).*weng'.*dt);
percent_brake_energy = brake_energy_captured/brake_energy_total*100 ;

% Torque error
trq_balance = T_dmd_eng' - Tbrakeopt - Tmotopt*tc.ratio - (Tengopt - ...
cstr.Teng_loss'.*(Tengopt>1)).*cpl.eff;
trq_balance_error = sum(trq_balance);

% Times the cycle demands were not able to be met
error_number = sum(abs(trq_balance)>0.1);

% Display values for the simulation:
fprintf('\n  MPC Exponential Decay Simulation Parameters: \n')
fprintf(' The Simulation has been run on the %s cycle \n',cycle)
fprintf(' with a resolution of n = %5.0f samples \n',n)
fprintf(' a control horizon of nc = %5.0f seconds \n',nc)
fprintf(' a prediction horizon of np = %5.0f seconds \n',np)
```

```
fprintf(' an Engine On/Off cost of %7.1e \n',MPC.eng_on_off_cost)
fprintf(' a SOC penalty cost of %7.1e \n',max(MPC.SOC_cost))
fprintf(' simulation time of %6.1f seconds \n', simutime)

fprintf('\n The Fuel economy is %6.3f mpg \n',FuelEconomy)
fprintf(' For an improvement of %5.2f %% \n',percent_improvement)
fprintf(' The engine remained off for %6.1f seconds\n', time_engoff)
fprintf(' The motor can handle %4.1f %% of the braking energy \n',percent_brake_energy)
fprintf(' There are %2.0f ''errors'' - points the vehicle \n did not meet the...
 cycle demands\n',error_number)
fprintf(' for a total torque balance error of %6.1e N-m. \n',trq_balance_error)

fprintf('\n Vehicle Parameters: \n')
fprintf(' Vehicle Mass = %6.0f kg \n',mass)
fprintf(' Torque converter ratio = %4.2f \n',tc.ratio)

fprintf('\n Ultracapacitor Pack Parameters: \n')
fprintf(' Initial SOC = %5.2f \n',SOCopt(1))
fprintf(' Cells in Series = %4.0f \n', ess.num_cell_series)
fprintf(' Max Voltage = %6.2f Volts\n', ess.pack_vmax)
fprintf(' Min Voltage = %6.2f Volts\n', ess.pack_vmin)
fprintf(' Capacitance = %6.2f Farads\n', ess.pack_cap(1))
fprintf(' Resistance = %6.4f Ohms \n', ess.pack_res)
```

Bibliography

[1] 2010 Toyota Prius-6063 Hybrids BOT Battery Test Results. Technical report, U.S. DOE - Office of Energy Efficiency and Renewable Energy. Tested on 15 July 2009, URL: http://avt.inel.gov/pdf/hev/batterygenIIIprius6063.pdf.

[2] Technology Road Map for the 21st Century Truck Program[21CT-001], December 2000.

[3] USABC Energy Storage System Goals, 2002. [Online; accessed 28-Sept-2008] URL: http://www.uscar.org/guest/article_view.php?articles_id=85.

[4] Progress Report on Clean and Efficient Automotive Technologies Under Development at EPA. Interim Technical Report EPA420-R-04-002, U.S. EPA - Office of Transportation and Air Quality, January 2004.

[5] A. Stefanopoulou A. Vahidi and H. Peng. Current Management in a Hybrid Fuel Cell Power System: A Model-Predictive Control Approach. *IEEE Transactions on Control Systems Technology*, November 2006.

[6] Argonne National Labs. *Powertrain Systems Analysis Toolkit*. [Software documentation by A. Rousseau, P. Sharer, S. Pagerit] More info at: http://www.transportation.anl.gov/modeling_simulation/PSAT/index.html.

[7] B. Asadi and A. Vahidi. Predictive Use of Traffic Signal State for Fuel Saving. *Proceedings of the 12th IFAC Symposium on Control in Transportation Systems*, September 2009.

[8] A. Bemporad. Model Predictive Control Design: New Trends and Tools. *IEEE Conference on Decision and Control*, December 2006.

[9] J. Besenhard, editor. *Handbook of Battery Materials*. Wiley VCH, 1999.

[10] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, second edition, 2003.

[11] D. Cross and C. Brockbank. Mechanical Hybrid System Comprising a Flywheel and CVT for Motorsport and Mainstream Automotive Applications. *SAE Paper*, (2009-01-1312), 2009.

[12] A. Vahidi D. Rotenberg and I. Kolmanovsky. Ultracapacitor Assisted Powertrains: Modeling, Control, Sizing, and The Impact on Fuel Economy. *American Control Conference*, 2008.

[13] Departments of the Army and Air Force. *TM_9-2320-365-10 [Technical Manual-2.5Ton 4x4 LMTVs]*, June 1998. [Online; accessed 2-Feb-2010] URL: http://www.combatindex.com/store/tech_man/Sample/M1078/TM_9-2320-365-10.pdf.

[14] S. Dhameja. *Electric Vehicle Battery Systems*. Elsevier Butterworth-Heinemann, 2002.

[15] H. Douglas and P. Pillay. Sizing Ultracapacitors for Hybrid Electric Vehicles. *IEEE IECON, 31st Annual Conference*, November 2005.

[16] Eaton Corporation. *V-FIFI-MC003-E [Hydraulic Accumulators Catalog]*. [Online; accessed 6-Feb-2009] URL: www.eaton.com.

[17] EIG - Energy Innovation Group. *Product Specification - C020 Cells*. [Online; accessed 22-June-2009] URL: www.eigbattery.com.

[18] M. Fodor F. Borrelli, A. Bemporad and D. Hrovat. An MPC/Hybrid System Approach to Traction Control. *IEEE Transactions on Control Systems Technology*, May 2006.

[19] J. Fenton and R. Hodkinson. *Lightweight Electric/Hybrid Vehicle Design*. Elsevier Butterworth-Heinemann, 2001.

[20] M. Fetcenko. Advanced Materials for Next Generation NiMH Batteries, March 2008. [Online; accessed 29-May-2009] URL: http://www.ovonic.com/PDFs/ovonic-materials/Ovonic-Fetcenko-2008-Wolsky-Seminar.pdf.

[21] F. Assadian C. Dextreit S. Di Cairano G. Ripaccioli, A. Bemporad and I. Kolmanovsky. Hybrid Modeling, Identification, and Predictive Control: An Application to Hybrid Electric Vehicle Energy Management. *HSCC*, 2009.

[22] Garmin Ltd. *Owner Manuals, Garmin nuvi 1690 and Oregon 450*. [Online; accessed June-2010] URL: http://www.garmin.com.

[23] L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Springer, 2005.

[24] A. Phillips M. Kuang I. Kolmanovsky H. Borhan, A. Vahidi. Predictive Energy Management of a Power-Split Hybrid Electric Vehicle. *American Control Conference*, June 2009.

[25] S. Cui H. Yu and T. Wang. Simulation and Performance Analysis on an Energy Storage System for Hybrid Electric Vehicle Using Ultracapacitor. *IEEE Vehicle Power and Propulsion Conference*, September 2008.

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

[27] I. Husain. *Electric and Hybrid Vehicles: Design Fundamentals*. CRC Press, 2003.

[28] J. W. Dixon J. Moreno, M. E. Ortuzar. Energy-Management System for a Hybrid Electric Vehicle, Using Ultracapacitors and Neural Networks. *IEEE Transactions on Industrial Electronics*, April 2006.

[29] J. Kargul. Hydraulic hybrids: Cost effective clean urban vehicles, March 2006. Michigan Clean Fleet Conference.

[30] D. Kirk. *Optimal Control Theory: An Introduction*. Dover Publishers (2004), 1970.

[31] M. Kutz. *Environmentally Conscious Transportation*. John Wiley and Sons, 2008.

[32] W. H. Kwon and S. Han. *Receding Horizon Control: Model Predictive Control for State Models*. Springer, 2005.

[33] R. A. Dougal L. Gao and S. Liu. A New Battery/Ultracapacitor Energy Storage System Design and Its Motor Drive Integration for Hybrid Electric Vehicles. *IEEE Transactions on Power Electronics*, January 2005.

[34] B. Egardt L. Johannesson, S. Pettersson. Predictive Energy Management of a 4QT Series-Parallel Hybrid Electric Bus. *Control Engineering Practice*, 17, 2009.

[35] S. Gay A. Emadi M. Ehsani, Y. Gao. *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design*. CRC Press, 2005.

[36] Wahyudi R. Muhida M.A.S. Kamal, Raisuddin. Comprehensive Driving Behavior Model for Intelligent Transportation Systems. *Proceedings of the International Conference on Computer and Communication Engineering*, May 2008.

[37] Ultracapacitor Model: Designed to Simulate Electrical and Thermal Performance. White Paper 3JMM, Maxwell Technologies, Inc., April 2009. URL: www.maxwell.com.

[38] Maxwell Technologies, Inc. *Maxwell UC Comparison - 31 August 2009*. [Online; accessed 10-Feb-2010] URL: www.maxwell.com.

[39] J. M. Miller. How to size ultracapacitors. *EE Times - India [Online: http://www.eetindia.com]*, July 2008.

[40] Tesla Motors. Increasing Energy Density Means Increasing Range. *Online: http://www.teslamotors.com/roadster/technology/battery*, 2010.

[41] R. Nine N. Clark, J. Daley and C. Atkinson. Application of the New City-Suburban Heavy Vehicle Route (CSHVR) to Truck Emissions Characterization. *SAE Paper*, (1999-01-1467), 1999.

[42] Y. Li Q. Gong and Z. Peng. Optimal Power Management of Plug-in HEV with Intelligent Transportation System. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, September 2007.

[43] S.J. Qin and T.A. Badgwell. A Survey of Industrial Model Predictive Control Technology. *Control Engineering Practice*, 11, 2003.

[44] A. Bollig R. Beck and D. Abel. Comparison of Two Real-Time Predictive Strategies for the Optimal Energy Management of a Hybrid Electric Vehicle. *Oil & Gas Science and Technology*, 63(4), 2007.

[45] F. Richert A. Bollig K. Neib T. Scholt K. E. Noreikat D. Abel R. Beck, S. Saenger. Model Predictive Control of a Parallel Hybrid Vehicle Drivetrain. *IEEE Conference on Decision and Control and European Control Conference*, December 2005.

[46] Freightliner Press Release. Freightliner Trucks Launches Run-Smart Predictive Cruise for Cascadia. pages Online, URL: http://www.freightlinertrucks.com, March 2009.

[47] M. Ferdowsi S. Lu, K. A. Corzine. A New Battery/Ultracapacitor Energy Storage System Design and Its Motor Drive Integration for Hybrid Electric Vehicles. *IEEE Transactions on Vehicular Technology*, July 2007.

[48] H. Sams. *Handbook of Electronics Tables and Formulas*. H. W. Sams, sixth edition, 1986.

[49] C. Manzie T. Kim and R. Sharma. Model Predictive Control of Velocity and Torque Split in a Parallel Hybrid Vehicle. *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, October 2009.

[50] L. A. Viterna. Ultra-Capacitor Energy Storage in a Large Hybrid Electric Bus. *14th International Electric Vehicle Symposium and Exposition*, December 1997. NASA Document Number 97-206319.

[51] A. Vahidi W. Greenwell. A Decentralized Model Predictive Control Approach to Power Management of a Fuel Cell-Ultracapacitor Hybrid. *American Control Conference*, 2007.

[52] R.J. Collingridge W. Millar and D.A. Ward. Consumer Vehicle Telematics An Emerging Market Where Web Services Offer Benefits. *BT Technology Journal*, 22(1), Jan 2004.

[53] M. Westbrook. *The Electric Car: Development and Future of Battery, Hybrid, and Fuel Cell Cars*. Institution of Electrical Engineers, 2005.

[54] V. Winstead and I. Kolmanovsky. Estimation of Road Grade and Vehicle Mass via Model Predictive Control. *IEEE Conferece on Control Applications*, 2005.

[55] www.dictionary.com. Definitions, Telematics and Informatics. Accessed June 2010.

[56] C. Zhang and A. Vahidi. Role of Terrain Preview in Energy Management of Hybrid Electric Vehicles. *Transactions on Vehicular Technology*, March 2010.