2016

INCORPORATING DRIVER'S BEHAVIOR INTO PREDICTIVE POWERTRAIN ENERGY MANAGEMENT FOR A POWER-SPLIT HYBRID ELECTRIC VEHICLE

Lei Feng
Michigan Technological University, leifeng@mtu.edu

INCORPORATING DRIVER'S BEHAVIOR INTO PREDICTIVE POWERTRAIN ENERGY MANAGEMENT FOR A POWER-SPLIT HYBRID ELECTRIC VEHICLE

By

Lei Feng

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Mechanical Engineering – Engineering Mechanics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2016

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Mechanical Engineering-Engineering Mechanics

Department of Mechanical Engineering – Engineering Mechanics

Dissertation Advisor: *Dr. Bo Chen*

Committee Member: *Dr. Jeffrey D. Naber*

Committee Member: *Dr. Jeffrey B. Burl*

Committee Member: *Dr. John E. Beard*

Department Chair: *Dr. William W. Predebon*

Table of Contents

List of Figures

List of Tables

Preface

Included in Chapter 3, Chapter 4 and Chapter 5 are a journal article and two conference proceedings respectively, reproduced in their entirety. I have written all three publications as the lead contributor. As one of the co-author for Chapter 3, Dr. Wenjia Liu has provided Matlab code for pattern recognition algorithm. Ming Cheng has provided ideas about advanced battery models for Chapter 4. As the co-author of all three chapters, Dr. Bo Chen has provided important guidance and advises.

Acknowledgements

First of all, I would like to provide the greatest appreciation to my research advisor, Dr. Bo Chen, for her consistent guidance, support and encouragement. Her dedication and passion will keep inspiring me for a very long time. She helped me develop the three most important qualities of being a PhD, perseverance, cogency, and tenacity by setting herself as an example.

I am grateful to my committee members, Dr. Jeffrey D. Naber, Dr. Jeffrey B. Burl, and Dr. John E. Beard for their help and advices. Dr. Naber's and Dr. Beard's HEDV courses captivated my interests in hybrid electric vehicles.

My sincere thanks to the department of Mechanical Engineering for their consistent financial support.

It has been a great pleasure to be part of the laboratory of Intelligent Mechatronic and Embedded Systems. All the discussion, idea sharing, debating and care that my colleagues have provided to me consisted of a major part of the priceless memory of my five-year PhD life. Great thanks to Wenjia Liu, Wei Luo, Ming Cheng, Yang Li, Zicheng Ge, Hao Wu, Chong Cao, Xin Wang, Luting Wang and Chuanliangzi Liu.

At last, I would like to express my extreme gratitude to my loving parents. Their full-hearted support, encouragement is instrumental to my success of obtaining the PhD degree. Their presence for my dissertation defense and graduation commencement made the finale of my five-year PhD life and a memory for life.

Abstract

The goal of this series of research is to advance hybrid electric vehicle (HEV) energy management by incorporating driver's driving behavior and driving cycle information. To reduce HEV fuel consumption, the objectives of this research are divided into the following three parts.

The first part of the research investigates the impact of driver's behavior on the overall fuel efficiency of a hybrid electric vehicle and the energy efficiency of individual powertrain components under various driving cycles. Between the sticker number fuel economy and actual fuel economy, it is well known that a noticeable difference occur when a driver drives aggressively. To simulate aggressive driving, the input driving cycles are scaled up from the baseline driving cycles to higher levels of acceleration/deceleration. The simulation study is conducted using Autonomie®, a powertrain simulation and analysis software. The performance of the major powertrain components is analyzed when the HEV is operated at different level of aggressiveness.

In the second part of the study, the vehicle driving cycles affect the performance of a hybrid vehicle control strategy and the corresponding overall performance of the vehicle. By identifying the driving cycles of a vehicle, the HEV supervisor controller system will be dynamically adapt the control strategy to the changes of vehicle driving patterns. With pattern recognition method, a driving cycle is represented by feature vectors that are formed by a set of parameters to which the driving cycle is sensitive. To establish reference

driving cycle database, the representative feature vectors of four federal driving cycles are generated using feature extraction method. The performance of the presented adaptive control strategy based on driving pattern recognition is evaluated using Autonomie.

In the last part of the study, a predictive control method is developed and investigated for hybrid electric vehicle energy management in effort to improve HEV fuel economy. Model Predictive Control (MPC), a predictive control method, is applied to improve the fuel economy of a power-split HEV. The study compares the performance of MPC method and conventional rule-base control method. A parametric study is conducted to understand the influence of 3 weighting factors in MPC formulation on the performance of the vehicles.

Chapter 1 Introduction

1.1. Background

Since the first oil price crisis in 1973, US has enacted the CAFE standards to regulate the fleet fuel economy of passenger vehicles and light trucks in the United States. United States consumes 7.5 million barrels of crude oil per day, and this number is projected to be 8.96 million barrel per day in 2016. Meanwhile, the world is consuming approximately 90 million barrels per day in 2013. This number has been increasing for 30 years. In 1983, this number was 58 million barrels. The world consumption of crude oil has increased 50% since 1983. By then end of 2013, the world proved oil reserve is approximately 1,646 billion barrels. If the world keeps consuming crude oil at the speed in 2013, current oil reserve will be depleted in 50 years.

United States government has been pushing for more stringent fuel economy and emission standards since 1978. In 2014, the CAFE fuel economy standard for passenger cars is 36.5 MPG, and that for light trucks is 26.4 MPG. According to the official document by NHTSA (National Highway Traffic Safety Administration), the average fuel economy of passenger cars will increase to 56.2 – 55.3 MPG by 2025 [1, 2].

1.2. Literature review

1.2.1. Pattern recognition

An HEV has better fuel economy and less emission than a conventional internal combustion engine vehicle due to the existence of electric powertrain. The introduction of additional powertrain components, however, makes the HEV control more challenging and the performance of HEVs is more sensitive to their control strategies. To achieve maximum fuel economy and minimum emissions, researchers in the automotive community have

made significant effort to investigate the major factors impacting fuel efficiency and develop optimal power management strategies for hybrid vehicles [3-7]. Research results showed that, in addition to vehicle and fuel characteristics, driving patterns have a strong impact on the fuel consumption and exhaust emissions [8, 9]. To optimize vehicle performance, multi-mode driving control method has been proposed for the adaptive vehicle control [10, 11]. The multi-mode driving control is defined as the control strategy which is able to switch a current control algorithm to the one that is optimized to the recognized driving pattern [10]. The ability to dynamically select control algorithms based on identified driving patterns leads to adaptive vehicle control, improved energy efficiency, and reduced green gas emissions.

A driving pattern is typically defined as the driving cycle of a vehicle in a particular environment. To recognize driving patterns, it is necessary to identify a list of characteristic parameters which can be used to describe driving patterns. Although there is no consensus among researchers about what parameters can be used for driving pattern recognition, several studies have attempted to find such a list of parameters. Ericsson [8] investigated the impact of 62 driving pattern parameters on fuel economy and emissions using a large amount of testing driving cycles. The study showed that nine driving pattern parameters (four associated with power demand and acceleration, three with gear changing behavior, and two with speed level) had an important effect on fuel consumption and emissions. Lin et al. [11] selected power demand related parameters and stop time for hybrid electric truck driving pattern recognition. In addition to vehicle parameters, Jeon et al. [10] incorporated road grade parameters in the driving pattern recognition. For pattern classification method, neural network [10], support vector machine (SVM) [12], and learning vector quantization

network [13] were applied for the driving pattern classification. However, most existing driving pattern recognition methods are based on binary classification, which may cause losing of information.

The importance of driving patterns to the fuel economy and emission justifies a systematic study of driving pattern recognition. In this chapter, a supervised pattern recognition approach is studied for the classification of a real-world driving cycle to a similar driving cycle in the representative driving cycle group. Four federal driving cycles, Urban Dynamometer Driving Schedule (UDDS), Highway Fuel Economy Driving Schedule (HWFET), a high acceleration aggressive driving schedule (US06), and an air conditioning driving schedule (SC03), are selected as representative driving cycles. These driving cycles represent different street types, driver behavior, and weather condition. With pattern recognition method, driving cycles and environmental information for various driving patterns are represented by corresponding features vectors. The classification is based on the distance of a test feature vector (test driving cycle) to the representative feature vectors (representative driving patterns). The test driving pattern is classified to one of representative driving pattern with which the test driving pattern has the smallest distance. The identified driving pattern information is then used to implement adaptive control strategies. The performance of adaptive control is evaluated in a powertrain/propulsion simulation and analysis software - Autonomie.

1.2.2. Driver behavior and vehicle energy efficiency

Under the real-world driving scenario, the driver's behavior is one of the major influence factors to fuel economy of vehicles. Under normal driving circumstances, hybrid electric vehicles have a better fuel economy than conventional vehicles because electric motors are

the main traction power sources which have positive impact on the overall efficiency of the vehicle. Due to the one more degree of freedom in HEV powertrain, the engine speed or engine torque becomes independent from the driver's request. Thus, with proper control, the engine operating region could be placed within an optimal region and the better fuel economy will be achieved. However, the scenario described above only happens when the driver's power request is within a reasonable range and the most of the power request could be provided by the electric motors. If the driver's behavior shows aggressive driving style, the power request for the powertrain increases rapidly. Usually, this kind of instantaneous power request peaks will result in frequent starting and stopping of the engine and pushing the engine operation to the inefficient region.

A number of researchers have conducted pioneer study of the impact of aggressive driving on vehicle fuel consumption. [13-16] Sharer et al. [17] performed a series of HEV dynamometer tests to study why the real-world fuel economy number is different from the window sticker number by Environmental Protection Agency(EPA). The tests are conducted for both HEV and conventional vehicles: Toyota Prius and Ford Focus. Both vehicles show an increase of fuel consumption for aggressive driving. Carlson et al. [18] performed similar vehicle dynamometer tests on two Plug-in HEVs. From the experiment results, it is concluded that the PHEV is less sensitive to driver’s aggressiveness than HEVs due to its large battery capacity. However, the control strategy of the vehicle is of great influence to the sensitivity to aggressive driving. Lee and Son [19] conducted highway tests to find the relationship of real-time fuel consumption, vehicle speed, gear selection, brake/acceleration pedal usage, and steering angle. The test results showed that the fuel efficiency was significantly affected by the average depth of acceleration pedal in highway

driving in which the highest gear was engaged. The standard deviation of steering wheel angle and the fuel consumption were highly correlated.

1.2.3. Model predictive control and HEV

Fuel economy regulations and emission standards are becoming more stringent each year. According to the official document by NHTSA (National Highway Traffic Safety Administration), the average fuel economy of passenger cars will increase to 37.8 MPG (miles per gallon) by 2016 and 56.2 – 55.3 MPG by 2025, respectively [1, 2].

A power-split hybrid electric vehicle has better fuel economy and less emission than a conventional internal combustion engine vehicle due to electric powertrain. The introduction of additional powertrain components, however, makes the HEV control more challenging and the performance of HEVs is more sensitive to their control strategies. To achieve better fuel economy and less emissions, researchers have made significant effort to investigate the major factors impacting fuel efficiency and develop optimal power management strategies for hybrid vehicles [3-7]. The proposed energy management methodologies include dynamic programming (DP) [20-24], stochastic dynamic programming (SDP) [3, 25, 26], model predictive control (MPC) [27-29], Pontryagin's minimum principle [30-32], equivalent consumption minimization strategy (ECMS) [33, 34], instantaneous minimization methods [34-38], genetic algorithm [39], rule-based control [40-44], H_∞ control [45], and fuzzy logic control [46-51]. Liu et al. [52, 53] have proposed an analytical model for power-split HEVs and applied SDP method and ECMS to optimize fuel economy. On the one hand, SDP, as a combination of Markov Chain and DP, makes prediction based on states of the vehicle in the past. On the other hand, ECMS considers the current states of vehicles as well as the kinetic constraints of the planetary

gear systems. The simulation results of these two methods are both compared to that of the DP method. The DP method yields the best fuel economy for its nature as an exhaustive method. In general, SDP has better fuel economy comparing to ECMS method. For the drivability, the SDP has smoother engine power curve while the ECMS produces several power peaks.

Chapter 2 Research Environment and Vehicle Model

This dissertation research studies the impact of aggressive driving on HEV powertrain, driving cycle pattern recognition, and model predictive control for better fuel economy and performance. To perform these research tasks, a vehicle modeling and simulation software, Autonomie, has been selected as research environment for this study. A power-split HEV model, MY04 Prius, in Autonomie is used as a base vehicle model for simulation study and MPC controller evaluation. This chapter gives an introduction to Autonomie and MY04 Prius vehicle model.

2.1. Introduction to Autonomie

Autonomie is developed by the Argonne National Laboratory and it is the upgraded version of the Powertrain System Analysis Toolkit (PSAT). Autonomie runs in Matlab environment. All the models and the initialization files of the model are saved in Matlab format, thus, most Autonomie model are visible to users. Autonomie contains a number of calibrated vehicle models, for example, the Prius MY04 vehicle model. Autonomie models are forward-looking models. When a control decision is about to be made, the physical constraints in the vehicle are considered. Thus the response of the vehicle models is realistic.

Users have access to the models of individual powertrain components. Each vehicle model in Autonomie is modulized and can be divided into four major parts, Driver Controller, Vehicle Powertrain Controller, Vehicle Powertrain Architecture, and Environment as shown in Figure 1. The driver controller simulates the response of a driver by comparing the actual vehicle speed and the target speed. According to the output of the Driver

Controller, the Vehicle Powertrain Controller generates component-specific commands, for example, the engine torque and speed command. The control strategy in this controller is rule-based and vehicle-dependent. The Vehicle Powertrain Architecture simulates the response of each powertrain component and the powertrain operating information will be fed back to the two controllers described above. The Environment block provides the environmental information, for example, the ambient temperature and road grade.

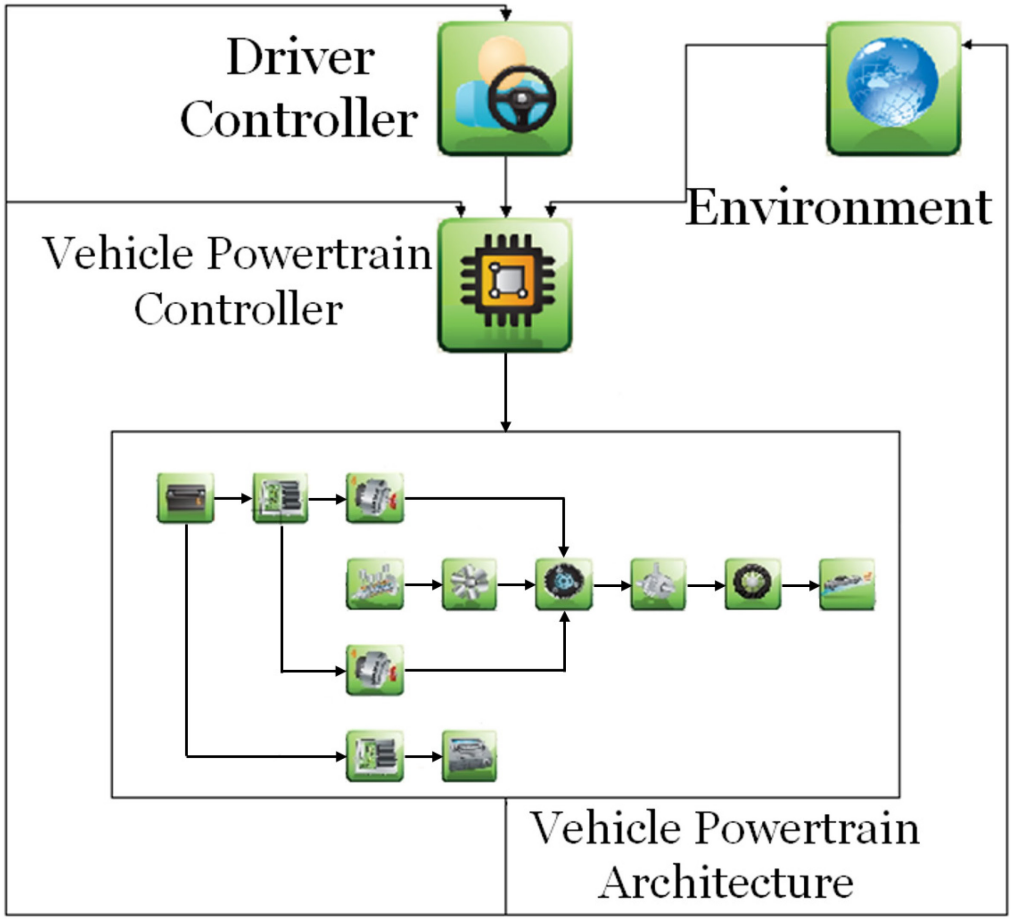

Figure 1. The Layout of Vehicle Models in Autonomie.

2.2. **Vehicle plant model**

2.2.1. Overview

The vehicle model used in this study is the MY04 Prius power-split HEV model provided by Autonomie. The key parameters of the model are listed in Table 1.

Table 1. The Major Model Parameters of MY04 Prius.

Parameter Name	Value	Unit
Max Engine Power	57	kW
Max Power of Motor/Generator 1 (Traction)	25(cont) 50(Peak)	kW
Max Power of Motor/Generator 2 (Speed balancing)	15(cont) 30(Peak)	kW
Battery Capacity	6.5	A.h
Curb Weight	1449	kg
Electrical Consumption (Urban/Highway)	64.68/49.04	W.h/mile

2.2.2. Engine

The engine model in Autonomie consists of two parts. One part is called engine controller, but it is functioning as constraint calculation. The other part of the engine model is the engine plant for calculating engine torque output and engine fuel rate.

There are three steps to determine the operating constraints for the engine in Autonomie, which are basic constraint calculation, transient behavior calculation, and load calculation. The first step calculates the maximum torque and minimum torque at current engine speed based on Wide Open Throttle (WOT) torque curve. In the second step, the transient behavior calculation determines the response of the engine to the engine torque request

from Vehicle Powertrain Controller (VPC) based on the engine status. In Autonomie, the engine operating status is categorized into three states: off state, cranking state, and running state. If the engine is at running state, the torque request from VPC will be fulfilled. Otherwise, the engine will not respond to the torque request. The last step of constraint calculation is to determine the ratio of the engine torque request to the maximum available engine torque at this time instance.

The engine plant model is to calculate torque output and instantaneous fuel rate. Engine torque and fuel rate are obtained from engine WOT torque curve and engine brake specific fuel consumption (BSFC) map, respectively. The engine model in Autonomie is considered as a quasi-static model.

The calculation of engine torque has following two steps. The first step is to estimate engine operating condition according to given engine speed and the torque request from VPC. The available engine torque depends on the engine speed. The relationships between the minimum engine torque and the WOT engine torque versus engine speed are given in Figure 2.

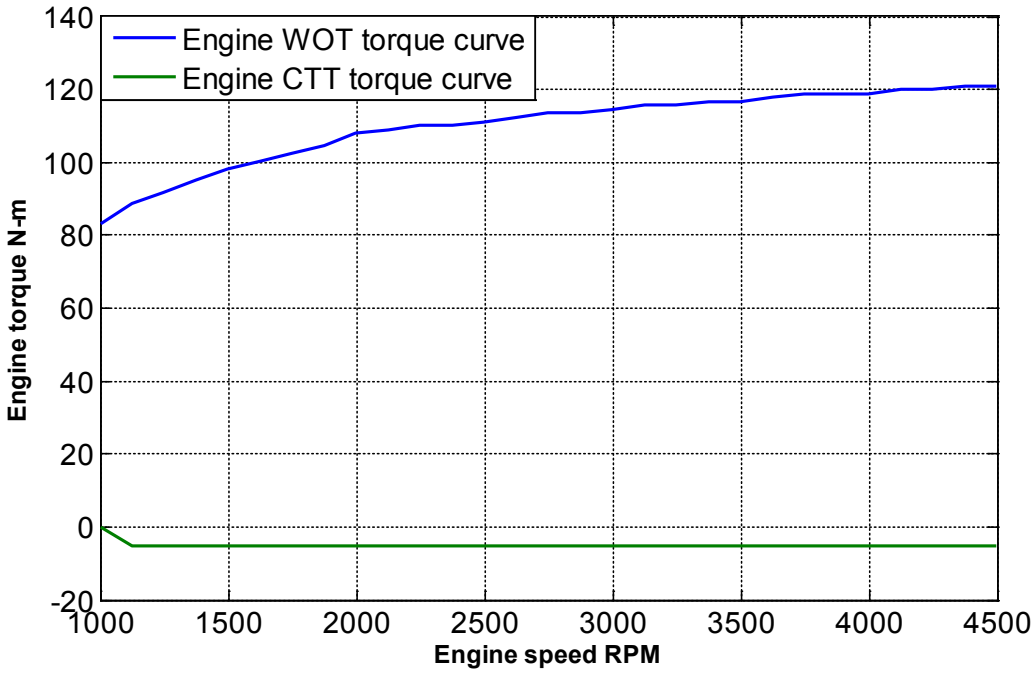

Figure 2. Engine minimum torque and maximum torque vs engine speed.

The second step is the engine fuel calculation based on BSFC map. CTT stands for closed throttle torque. The engine cranking process has been modeled in Autonomie. When engine is cranking, it needs external power source to overcome its internal friction. In the model, this internal friction is considered as negative torque output by the engine. This part of the model is based on the engine fuel consumption map, and it is given in Figure 3.

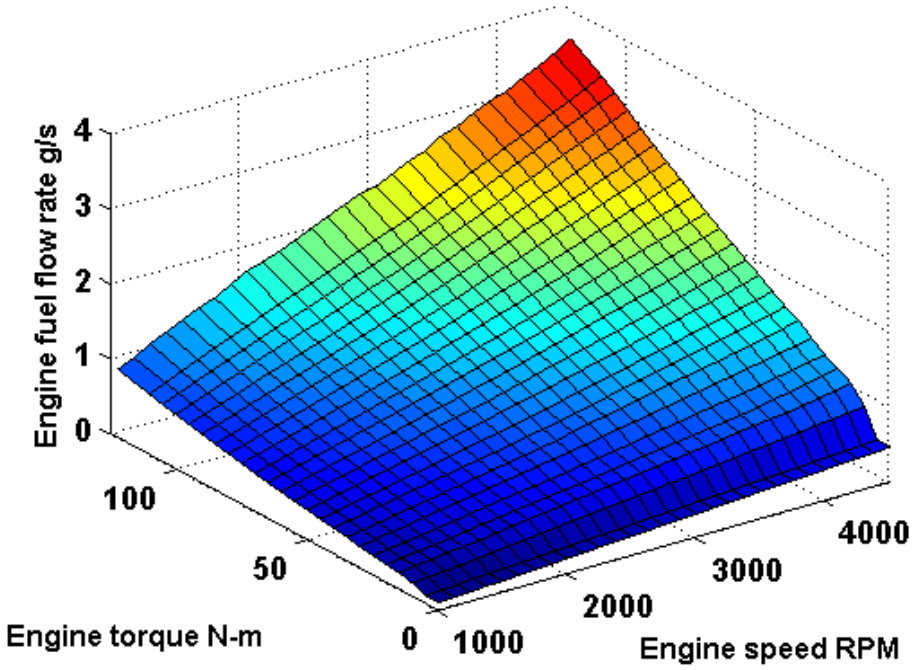

Figure 3. Engine fuel consumption map

The x axis is engine speed, and the y-axis is the engine torque. The z-axis is the engine fuel rate. The engine fuel rate is calculated based on interpolation in the engine fuel consumption map look-up table. The inputs to the look-up table are engine speed and engine torque. The output of the table is engine fuel rate. The engine fuel consumption map is obtained from dyno testing at Argonne National Laboratory.

Another important aspect of the engine model is to determine the operating status of the engine. The engine operating status has been modelled into three states (i.e. off, cranking, and running). The engine operation status are determine by three signals: engine speed, engine ON command, and engine torque. When the engine speed is greater than 800 RPM, the engine speed status is defined as “Engine running”. Otherwise, it is defined as “Engine off or cranking”. Engine ON command is the engine running request from vehicle powertrain controller. When the driver requests power, and the controller determines that

the traction motor torque is insufficient, Engine ON will be “True”. Engine torque is the amount of torque requested by the vehicle powertrain controller.

Table 2. Logic table of engine status decision.

Case	Statement 1 engine torque >0	Statement 2 engine speed>800 RPM	Statement 3 engine on = “True”	Decision
1	1	1	1	Engine running
2	1	1	0	Engine off
3	1	0	1	Engine off
4	0	1	1	Cranking
5	1	0	0	Engine off
6	0	1	0	Engine off
7	0	0	1	Engine off
8	0	0	0	Engine off

When the engine is running, the engine fuel rate is calculated by engine fuel rate look-up table with a given engine speed and a given engine torque.

When the engine is cranking, engine fuel is calculated based on the ratio between the simulated engine torque over minimum engine torque. This ratio multiplying the minimum engine fuel rate at this engine speed is the engine fuel consumption during cranking. When calculating fuel rate, the model refer to another engine map called engine internal friction torque over speed, as shown in Figure 4.

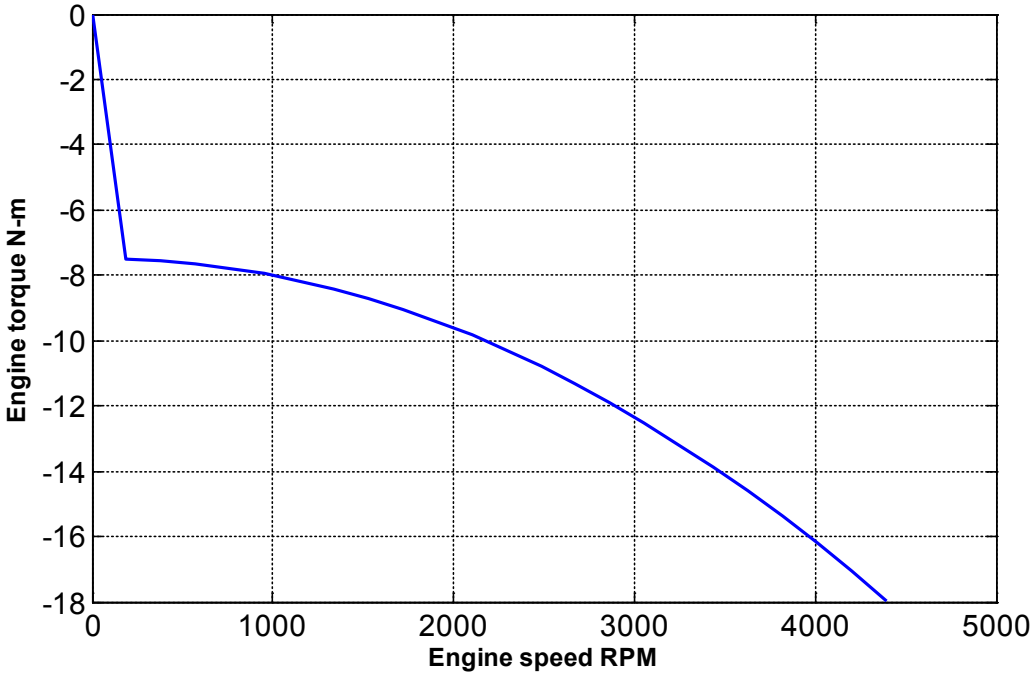

Figure 4. Engine internal friction torque vs. speed.

The engine minimum torque vs. speed map presents the amount of resistance torque in the engine when no fuel is injected and combusted in the cylinders. This is the minimum amount of torque required to crank the engine.

2.2.3. Battery

The battery model in Autonomie is an equivalent circuit model with the assumption that the performance of every cell in the battery is identical to each other. Similar to engine mode, the battery model consists of two parts: battery controller and battery plant model. The battery controller is for looking up maximum propulsion power and maximum regenerative braking power at a given SOC. The maximum charging power and maximum discharging power vs SOC are given in Figure 5.

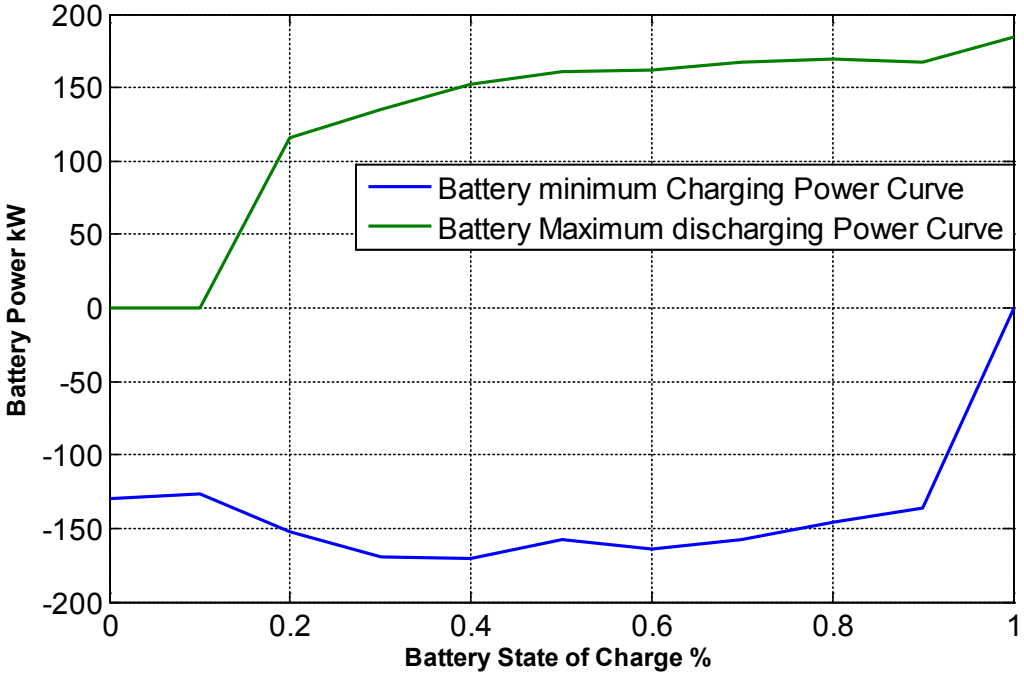

Figure 5. Battery max charging and max discharging power vs. SOC.

The battery plant model has the battery current as the only input. The battery current and battery voltage are updated each computation cycle in the battery plant mode. The battery plant model can be generally divided into two parts: SOC calculation and voltage calculation.

SOC calculation is based on the integral of the current using equation **(12)**.

$$SOC = \frac{1}{C}\int idt \qquad \textbf{(1)}$$

where C is the capacity of the battery.

Voltage calculation in Autonomie battery model is quasi-static as well. The following equation is used to calculating battery voltage.

$$V_{batt} = V_{oc} - i * R_{batt} \qquad \textbf{(2)}$$

where V_{batt} and V_{oc} are terminal voltage and open-circuit voltage of the battery, respectively. R_{batt} is the internal resistance of the battery. The hysteresis of the battery is

modeled by using two maps for R_{batt} at charging and discharging, respectively. The open-circuit voltage of the battery is also modeled by a look up table. The values of three look-up tables are shown in Figure 8.

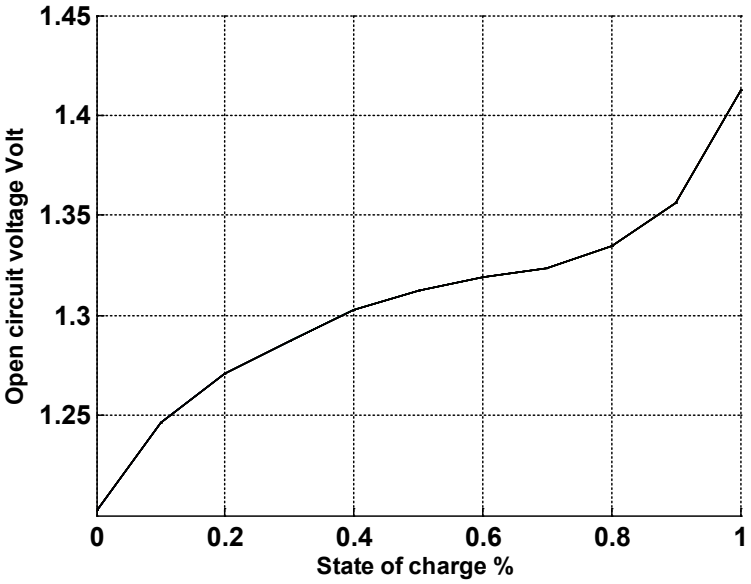

Figure 6. SOC vs V_{oc}

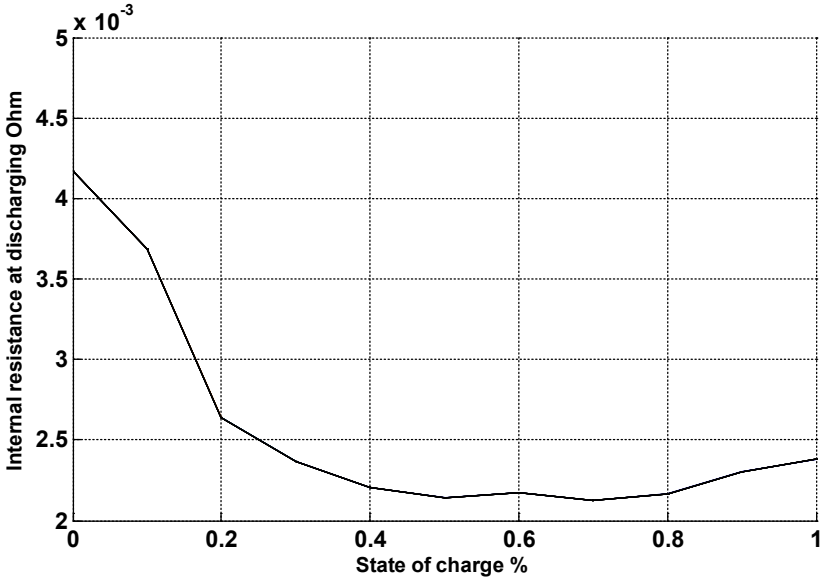

Figure 7. Internal resistance when charging vs. SOC

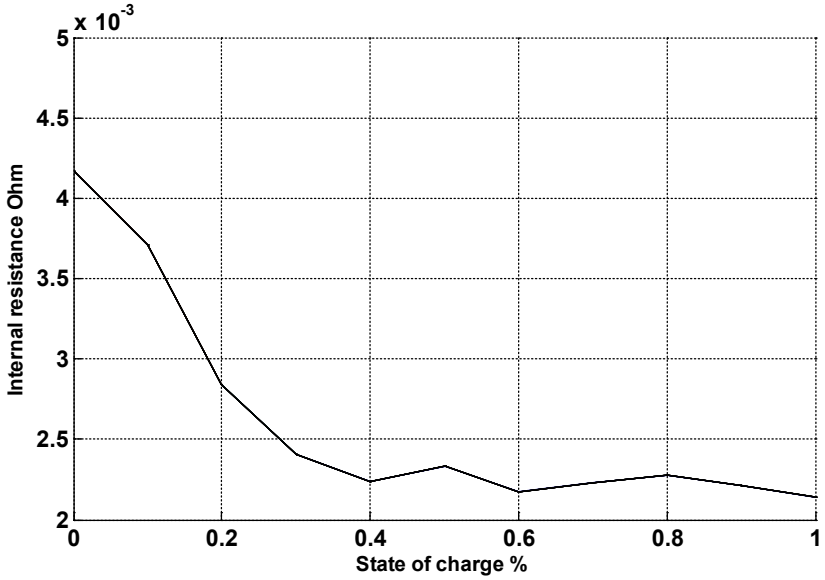

Figure 8. Internal resistance when discharging vs. SOC.

The battery current is defined to be positive when the battery is being discharged. The battery voltage at a given time is calculated based on current magnitude, current direction, and the internal resistance. The battery current is calculated based on the power output of the motors as well as the electronic power accessories.

2.2.4. Motor and Power electronics

Similar to engine and battery models, the motor model in Autonomie has both controller and plant models. The controller performs constraint calculation at a given speed. It also calculates the nominal load of the motor, varying from -1 to 1. The negative sign indicates that the motor is spinning in counter-clockwise direction.

The plant model of the motor calculates the torque output and current output. The maximum available torque from the motor is determined by the temperature of the motor. The actual temperature estimation is simplified into a heat index variable. The heat index varies from 0 to 1. When the heat index is 1, it means that the motor is heated, and it cannot output higher amount of torque than its continuous maximum. When the heat index is zero,

it means that the motor is ready to operate at its peak torque. When heat index is 0.5, the maximum torque of the motor is in between of its maximum peak torque and maximum continuous torque. In the last step, the motor plant model will calculate the current in the motor. Even though the motor is a PMAC motor, the current in the AC motor is the average root square voltage of the AC motor. The efficiency of the motor is implemented by using the following map in Figure 9.

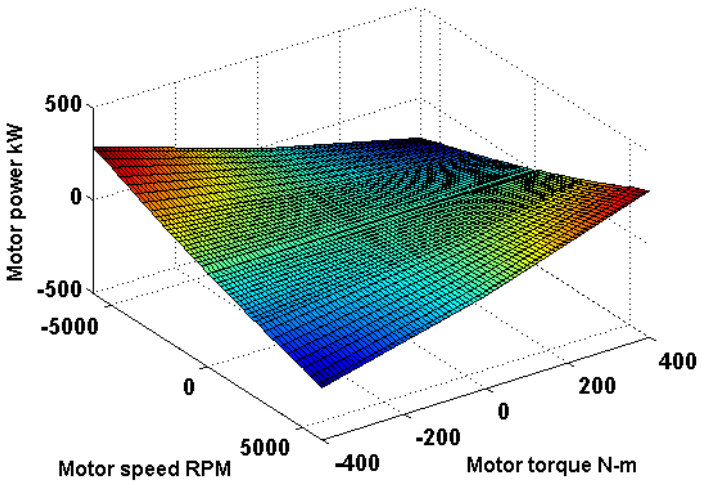

Figure 9. Motor power vs torque and RPM.

Figure 9 is calculated based on the motor speed vs. torque map with efficiency in consideration. In Autonomie, the DC to AC conversion is simplified to DC to root squared average of the AC voltage and current.

The power-electronics model in Autonomie also has a controller model and a plant model. The controller model consists of boost enabler and boost voltage calculator. The boost enabler looks into two enabling conditions. The one is regarding the powertrain power output and battery output power values. If either of these two power values are less than 10kW, the boost will be disabled. The other condition is considered true when the vehicle speed is greater than 45 Mph. If either of these two conditions is satisfied, the voltage

boosting will be enabled. The boost voltage calculator is to estimate the boosted voltage based on vehicle speed, battery output current, and battery output voltage. The power-electronics plant model is for calculating current output of the device and applying constant 95% efficiency to positive and negative voltage.

2.2.5. Planetary gear transmission

The planetary gear in Autonomie is modeled based on the physical dynamics of a planetary gear set. The ring gear is connected to the shaft of a traction motor. The carrier is connected to the engine. The sun gear is connected to a speed balancing motor. In this model, the torque output to the final drive and the speed of each components in the planetary gear set, including pinion gears, are calculated. This model also calculates the torque loss. The torque loss map is given in Figure 10. In Figure 10, x axis is the torque output to the final drive; the y axis is the speed of the ring gear of the planetary gear set; and the z axis is the amount of torque lost due to mechanical inefficiency between the ring gear of the planetary gear set and the final drive gear.

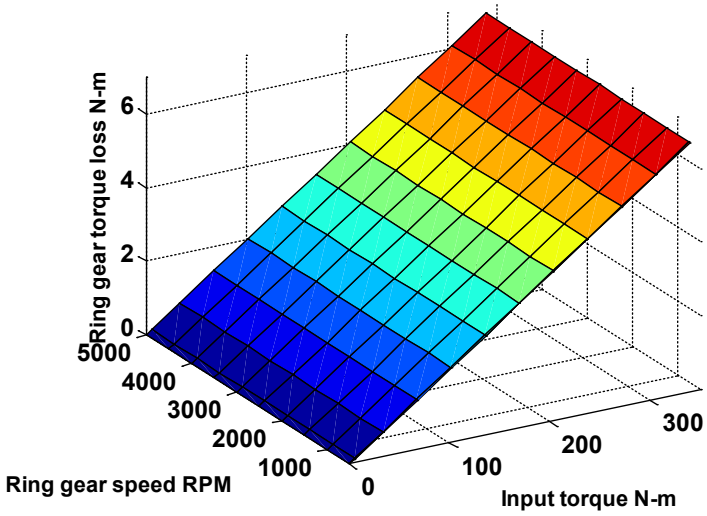

Figure 10. Torque loss map of a planetary gear set.

2.3. Vehicle Powertrain Controller model

2.3.1. Overview

To test and validate the predictive control design, the MPC controller will be developed in Simulink and integrated into Powertrain Controller in Autonomie. The block diagram of the MY04 Prius controller in Autonomie is shown in Figure 11. The signals labeled in green are the input signals. The signals labeled in red are the signals fed back from the plant models and the signals labeled in blue are the output signals from the controller. The powertrain controller with light green background will be integrated with the adaptive controller.

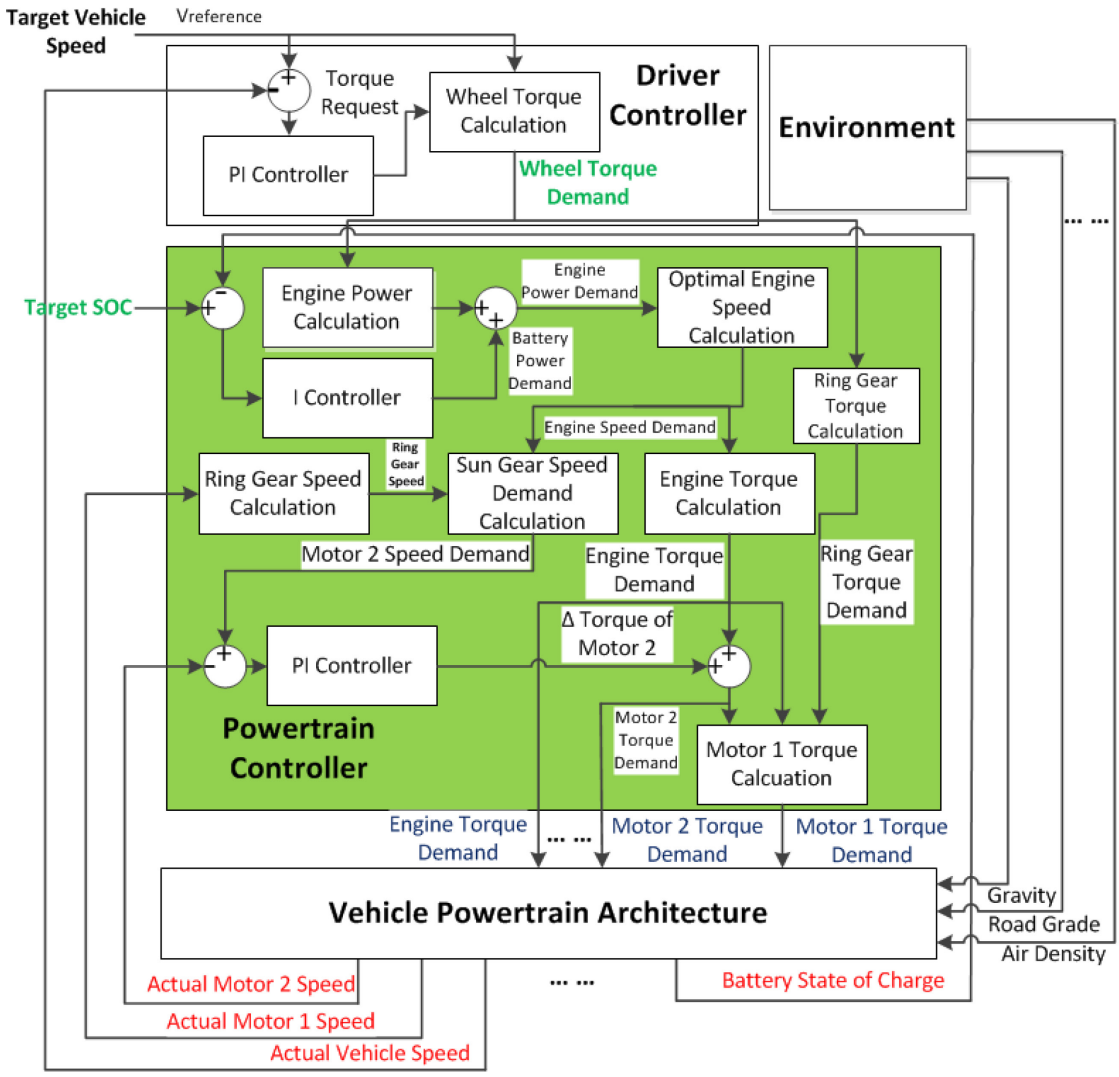

Figure 11. The Overview of the Toyota Prius Controller in Autonomie

Figure 11 gives a brief introduction of the control flow diagram in Autonomie MY04 Prius model. Driver gives wheel torque demand to VPC according to the error between the simulated vehicle speed and the target vehicle speed. VPC first calculates the engine power demand with the assumption that the wheel torque demand from the driver will be provided only by the engine. The optimal engine speed calculation gives the optimal engine speed at which the optimal engine power will produced, while the engine thermal efficiency is optimal. Based on the optimal engine torque and speed just found, finding the sun gear

speed and torque is the next step. Since the ring gear speed and engine speed are controlled by the driver and the optimal engine speed calculation, the target torque and speed of the sun gear can found by calculating the planetary gear speed constraints. However, a correction torque given by a PI controller is added to the target sun gear torque in order to bring sun gear torque to its target as soon as possible. At last, the ring gear torque will be calculated based on the planetary gear dynamic equation.

2.3.2. Engine On/Off control

The engine ON/OFF control in Autonomie consists of three criteria. If any of these three criteria is met, the engine on request will be sent out by the controller. These three criteria are given below,

1. Engine power request is greater than a threshold.
2. Acceleration pedal position is greater than a threshold.
3. Traction motor torque demand is greater than a threshold.

The three criteria for turning on the engine are engine power criteria, acceleration pedal criteria, and the motor torque demand threshold. Engine power request criteria is a calibrated value, and it is calibrated as 10kW for MY04 Prius. The APP threshold is set to 90%. This mode is also called performance mode in Autonomie. When the acceleration pedal is pushed to close the floor, the vehicle controller turns on engine to provide maximum propulsion power. The third scenario that engine is turned on is the motor saturation mode, in which the traction motor itself cannot fulfill the torque request from the driver. Thus, the engine has to start.

Engine power demand in Autonomie is calculated by adding the power demand of the driver and the SOC regulation demand from the battery together under the assumption that

the engine will provide all the traction power. The battery power demand is a function of the battery SOC as shown in Figure 12 when target SOC is 70%. In Figure 12, the x-axis is the battery SOC, and the y axis is the power request from the battery for SOC regulation. It could be inferred from the figure that the power request is positive when SOC is above 70% and is negative when SOC is less than 70%. That means when SOC is above 70%, the controller will tend to use battery for the propulsion power source. Otherwise, the controller will ask the engine to output more power in order to bring SOC back to its nominal state, 70%.

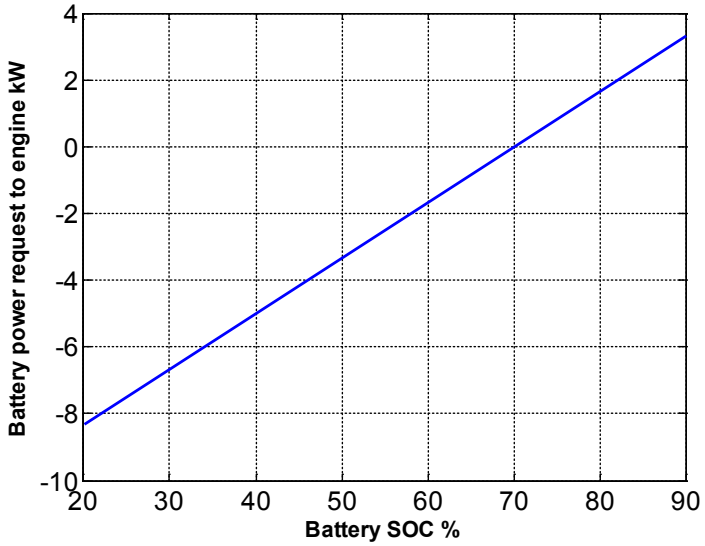

Figure 12. Battery power demand for SOC regulation.

2.3.3. Engine torque and speed calculation

Engine power demand calculation is shown in equation **(2)**.

$$P_{eng}=T_{request}*w_{request}+P_{soc\_regulation} \qquad \textbf{(3)}$$

$T_{request}$ is the wheel torque demand, and the $w_{request}$ is the target angular velocity of the vehicle. The $P_{soc\_regulation}$ is the amount of extra power that the engine needs to transfer to the battery other than satisfy the torque request of the driver.

Given the advantage that the engine speed in a power-split HEV is independent from vehicle speed, the controller is able to choose the optimal engine speed based on the power request. The optimal engine speed of a given power is provided by the following map in Figure 13 based on engine efficiency map. In Figure 13, the x-axis is the engine power, and the y-axis is the optimal engine speed for this given power. With optimal speed calculated, the optimal torque can be calculated as well. Figure 14 is an illustration about how constraints are applied in the process of calculating engine optimal speed and torque. In order to reach a possible steady-state with the capacity of the equipped speed balancing motor, the torque and speed output of the engine is limited by the torque and speed limit of the speed balancing motor.

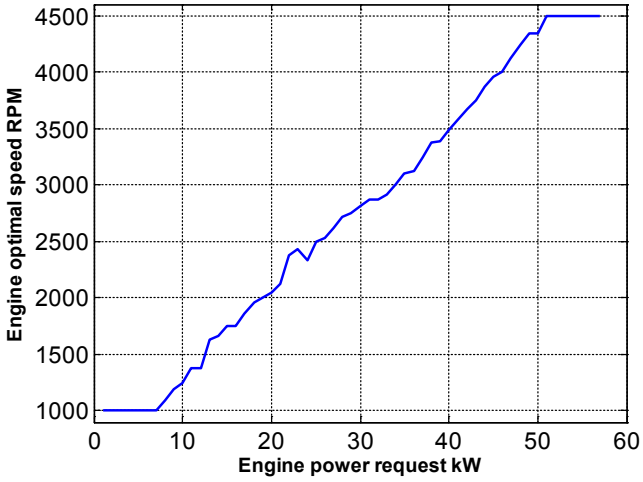

Figure 13. Engine power vs speed map.

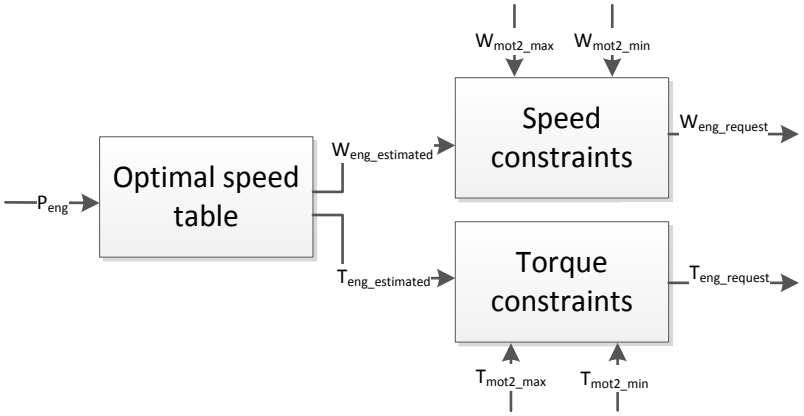

Figure 14. Constraints on engine output calculation.

2.3.4. Torque and speed estimation of motor 1 (traction motor)

Dynamics of the planetary gear sets is given in Autonomie by the following equation:

$$\begin{bmatrix} J_s & 0 & 0 & 0 & N_s & 0 \\ 0 & J_{cc} & 0 & 0 & -R_c & R_c \\ 0 & 0 & J_r & 0 & 0 & -N_r \\ 0 & 0 & 0 & J_p & R_p & R_p \\ \frac{N_s}{N_r} & -\frac{N_r+N_s}{N_r} & 1 & 0 & 0 & 0 \\ \frac{N_s}{N_r} & 0 & -1 & \frac{N_r-N_s}{N_r} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{w}_s \\ \dot{w}_c \\ \dot{w}_r \\ \dot{w}_p \\ F_{s-p} \\ F_{r-p} \end{bmatrix} = \begin{bmatrix} T_s \\ T_c \\ T_r - T_{request} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \textbf{(4)}$$

In Autonomie, the process of estimating the torque of the motor is done by omitting the angular acceleration of the ring gear, $\dot{w}_r$. By taking the inverse matrix A on the left-hand side, the matrix equation could be written as,

$$\begin{bmatrix} \dot{w}_s \\ \dot{w}_c \\ \dot{w}_r \\ \dot{w}_p \\ F_{s-p} \\ F_{r-p} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} & B_{15} & B_{16} \\ B_{21} & B_{22} & B_{23} & B_{24} & B_{25} & B_{26} \\ B_{31} & B_{32} & B_{33} & B_{34} & B_{35} & B_{36} \\ B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & B_{46} \\ B_{51} & B_{52} & B_{53} & B_{54} & B_{55} & B_{56} \\ B_{61} & B_{62} & B_{63} & B_{64} & B_{65} & B_{66} \end{bmatrix} \begin{bmatrix} T_s \\ T_c \\ T_r - T_{request} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \textbf{(5)}$$

Thus it could be inferred from the equation by calculating the third row of the inversed matrix with vector x.

$$J\dot{w}_r = B_{31}T_s + B_{32}T_c + B_{33}(T_r - T_{request}) \quad \textbf{(6)}$$

Assuming $\dot{w}_r = 0$, the torque of the traction motor could be calculated by re-arrange the equation above,

$$T_r = -\frac{B_{31}T_s + B_{32}T_c}{B_{33}} + T_{request} \quad \textbf{(7)}$$

Where T_s, T_c, and $T_{request}$ are already obtained by the previous calculations

2.3.5. Torque and speed estimation of motor 2 (speed balancing motor).

In order to control the speed of the engine, the speed balancing motor in a planetary gear set is working as the enabling part of a continuous variable transmission. Thus it is also called EVT (Electric Variable Transmission). By adjusting the speed of the sun gear, the speed of the engine connected to the carrier gear will always be greater than 0 RPM. When the engine is cranking, the speed balancing motor is working as a starter. The control of the speed balancing motor is a critical part in controlling the planetary gear transmission as well as the entire vehicle powertrain. Figure 15 shows how the speed and torque the speed balancing motor is controlled.

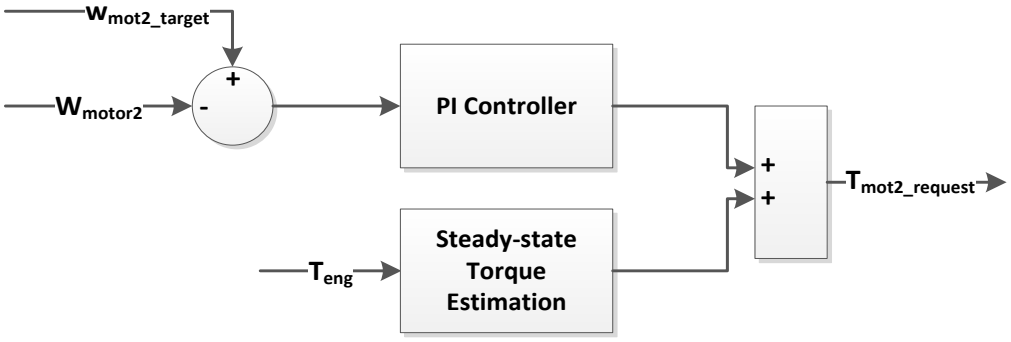

Figure 15. Speed balancing motor torque calculation.

The input to the speed balancing motor controller are current motor speed, motor speed target, and the engine torque request. The role of the speed balancing motor is to adjust the engine speed. Given the target engine torque and the engine speed, the steady state speed and torque of the motor when the target is reached can be calculated using speed constraint equation as well as dynamic equation. These two equations are given below.

$$J_s \dot{w}_s = T_s - R_s F_{s-p} \quad \textbf{(8)}$$

$$N_s w_s + N_r w_r = (N_s + N_r) w_c \quad \textbf{(9)}$$

Assume that the steady state is reachable, the target speed of the motor can be given by the following equation by re-arrange the speed constraints. Thus it can be inferred from the equation.

$$w_s = \frac{(N_s + N_r) w_c - N_r w_r}{N_s} \quad \textbf{(10)}$$

$$T_s = R_s F_{s-p} \quad \textbf{(11)}$$

Since the torque calculated from the equation above is for steady state. In order to have better transient performance, a PID controller is included in the speed balancing motor to adjust the amount of torque from the motor.

2.4. **Summary**

In Autonomie, the vehicle plant model and the controller model of MY04 Prius is realistic. Thus, it is a sound base vehicle model to perform controller design and optimization, especially for advanced control algorithm such as model predictive control and hybrid control.

Chapter 3 Driving pattern recognition for adaptive hybrid vehicle control[1] [54]

3.1. Introduction

The importance of driving patterns to the vehicle fuel economy and emission justifies a systematic study of driving pattern recognition. In this chapter, a supervised pattern recognition approach is studied for the classification of a real-world driving cycle to a similar driving cycle in the representative driving cycle group. Four federal driving cycles, HWFET, US06, and SC03, are selected as representative driving cycles. These driving cycles represent different street types, driver behavior, and weather condition. With pattern recognition method, driving cycles and environmental information for various driving patterns are represented by corresponding features vectors. The classification of feature vectors is based on the distance of a test feature vector (test driving cycle) to the representative feature vectors (representative driving patterns). The test driving pattern is classified to one of the representative driving pattern with which the test driving pattern has the smallest distance. The identified driving pattern information is then used to select appropriate controller parameters. The performance of adaptive control is evaluated by Prius MY04 model in Autonomie.

The rest of the chapter is structured as follows. Section 3.2 introduces the fundamentals of pattern-recognition-based driving cycle recognition. Section 3.3 presents the classification of real-world driving cycles using representative feature vectors of four federal driving cycles. Section 3.4 studies the impact of dissimilarity measures and feature extraction

[1] SAE©. Portions reprinted with permission, from Feng, L., W. Liu, and B. Chen, Driving Pattern Recognition for Adaptive Hybrid Vehicle Control. SAE International Journal of Alternative Powertrains, 2012. 1(1): p. 169-179. See Appendix B for documentation of permission to republish this material

methods on the performance of driving cycle pattern recognition. Section 3.5 discusses adaptive vehicle control based on driving cycle pattern recognition.

3.2. Pattern-recognition-based driving cycle recognition

3.2.1. Concept of pattern recognition

Pattern recognition is a scientific discipline whose goal is to classify objects into a number of meaningful categories or classes [55]. In pattern recognition, the patterns to be classified are usually the groups of measurements, defining points in an appropriate multidimensional space [55]. The measurements used for the classification are described by features. If p features are used f_i, i=1, 2… p, these p features can form a feature vector $F = \left(f_1, f_2, ..., f_P \right)^T$, where T denotes transposition. A feature vector is a point in P dimensional space R^P. The process of supervised pattern recognition consists of two steps, feature extraction and classification, respectively. In the feature extraction stage, a number of feature members are selected from the measurement data of the pattern. These feature members are used to form feature vectors to represent the pattern. In the pattern classification stage, the dissimilarity of the test pattern with the representative patterns is evaluated. The dissimilarity of two patterns is defined as a function of the distance between the corresponding feature vectors of the patterns. Usually, the shorter distance means higher similarity and the longer distance means lower similarity. As such, the test pattern is classified to one of representative patterns, with which the test pattern has the smallest distance. Different types of distance definitions can be used in pattern recognition. The Euclidian distance is one of the most commonly used distances. Let $X = \left(x_1, x_2, ..., x_n \right)^T$

and $Y = (y_1, y_2, ..., y_n)^T$ denote two feature vectors. The Euclidian distance between these two feature vectors is defined below:

$$d_{XY} = \sqrt{\left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)} \qquad \textbf{(12)}$$

3.2.2. Feature selection for driving cycle pattern recognition

Feature selection is application dependent. The rule of thumb for feature extraction is that the selected features can replicate most information of the original measurement data and separate feature vectors for different patterns in the feature space. To classify driving cycles, thirty-nine characteristic parameters were initially chosen based on Jeon and Ericsson's work [8, 10]. The high dimension of feature vectors, however, impedes practical application of driving cycle pattern recognition in real-time. To reduce the dimension of feature vectors, numerous simulation tests were performed to find a reduced set of feature members and the weighting factor for each feature member. The simulation work finally identified fifteen feature members and corresponding weighting factors, listed in Table 3, to form feature vectors for driving cycles as shown below:

$$f = (k_1 \times a_1,\ k_2 \times a_2,\ \cdots\cdots,\ k_i \times a_i,\ \cdots\cdots,\ k_{15} \times a_{15})^T, \qquad i \in [1, 15] \quad (13)$$

Where a_i is a feature member and k_i is a weighting factor

Table 3. Driving cycle feature members and corresponding weighting factors.

Index Number (i)	Feature Members (a)	Weighting Factors (k)
1	Average Cycle Speed (m/s)	10
2	Positive Average Acceleration (a>0.1m/s^2)	1
3	Low Speed Time (15-30 Km/h)/Total Time (%)	10
4	Mid High Speed Time (70-90 Km/h)/Total Time (%)	100
5	High Speed Time (>90 Km/h)/Total Time (%)	10
6	Extreme Deceleration Time (a<-2.5 m/s^2)/Total Time (%)	1000
7	High Deceleration Time (a<-2 & a>-2.5 m/s^2)/Total Time (%)	1
8	Maximum Cycle Acceleration (m/s^2)	100
9	Maximum Cycle Speed (Km/h)	6
10	Standard Deviation of Cycle Speed (Km/h)	1
11	Mid Deceleration Time (a<-1 & a>-1.5 m/s^2)/Total Time (%)	1000

12	Mid High Deceleration Time (a>-2 & a<-1.5 m/s^2)/Total Time (%)	1000
13	Mid Acceleration Time (a>1.5 & a<2 m/s^2)/Total Time (%)	1
14	High Acceleration Time (a>2 & a<2.5 m/s^2)/Total Time (%)	1000
15	Extreme Acceleration Time (a>2.5 m/s^2)/Total Time (%)	1000

3.2.3. Representative feature vectors for selected driving cycles

As discussed previously, a driving pattern is determined by multiple factors, including road type, driver behavior, weather and traffic conditions. To form a good representative driving cycle base which are able to reflect aforementioned features, four federal driving cycles, UDDS, HWFET, US06, and SC03, are selected as representative driving cycles. UDDS and HWFET are two driving cycles representing two road types. US06 represents the driver's behavior of aggressive driving. SC03 is chosen to represent the influence of humid weather. The profiles of these four driving cycles are shown in Figure 16-Figure 19. HWFET is a typical highway driving cycle featuring high speed and short stop time, while UDDS has the features of low average speed and long stop time. US06 has the highest average speed and presents extreme acceleration (a>2.5 m/s^2). SC03 is similar to UDDS but its acceleration is milder than that of UDDS due to the usage of air conditioner.

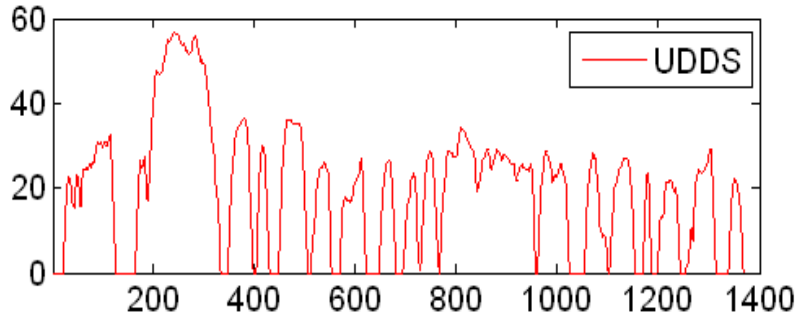

Figure 16. The driving cycle of UDDS.

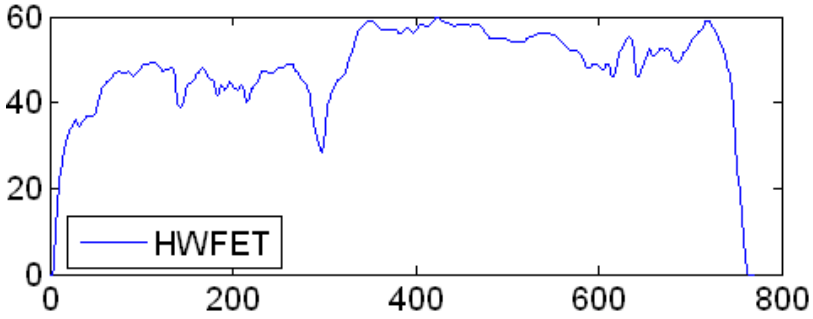

Figure 17. The driving cycle of HWFET.

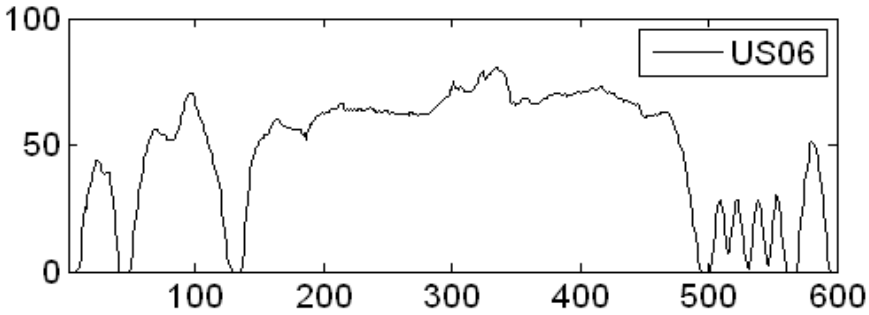

Figure 18. The driving cycle of US06.

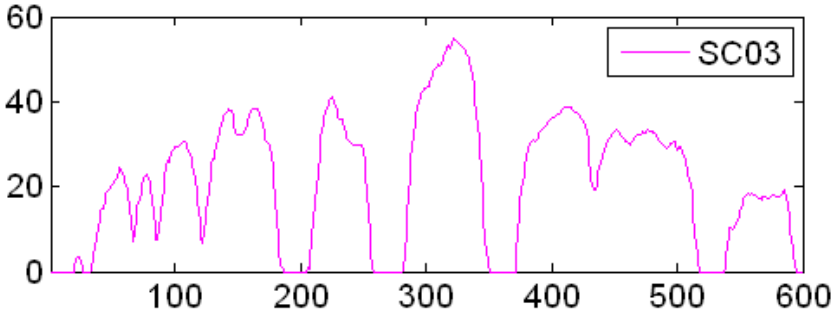

Figure 19. The driving cycle of SC03.

A feature vector of a driving cycle is calculated based on partial data points of a driving cycle for the quick recognition of driving cycle patterns, which is especially valuable in real-time applications. The number of data points is defined by the size of a sample window as shown in Figure 20. In this study, the window size is set to 450 sample points. To speed up the recognition of driving cycle patterns, the technique of sequential processing of measurement is applied. As such, the next feature vector is calculated by advancing the sample window by 50 sample points. During real-time driving cycle pattern recognition, vehicle controllers collect a number of data points defined by the size of the sample window and calculate a feature vector for the current sample window using parameters defined in Table 3. The real driving cycle is then classified to one of four representative driving cycles using classification algorithms and calculated feature vector. Once the driving cycle is recognized, the control parameters are switch to the one that is optimal to the recognized driving pattern. The time between one control decision points to the next control decision point is 50 seconds in this study if the sample rate is 1 Hz.

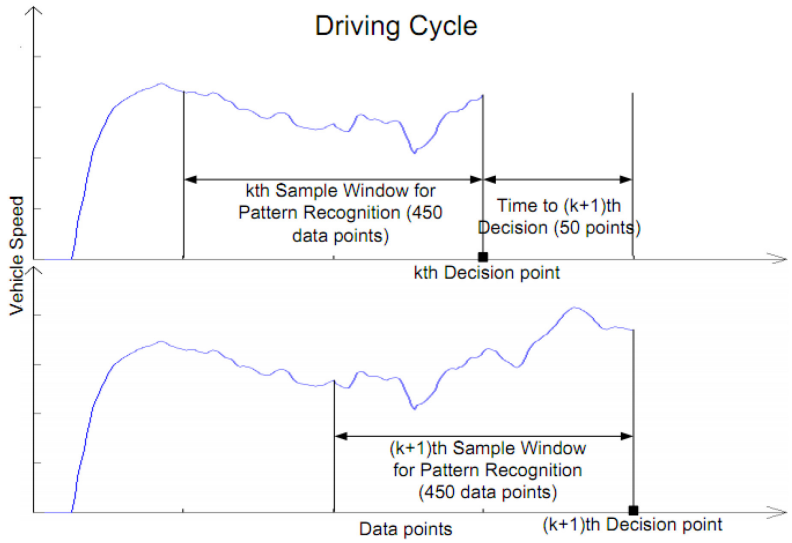

Figure 20. The definition of sample window and sequential processing of measurements.

To generate the representative feature vectors for the UDDS, HWFET, US06, and SC03, the velocity data for these driving cycles were downloaded from the U.S. Environmental Protection Agency website [56]. From each driving cycle, the most representative data segment was used to generate feature vectors for the corresponding driving cycle. For the UDDS, velocity data starting from 347 to the end of the driving cycle was used. Since these representative segments have different length, a common data length of 2000 data points was specified to ensure the same number of representative feature vectors for each driving cycle. The representative segment for each driving cycle was repeated to form a data set with a length of 2000. In each data set, the first 450 data points were used to form the first sample window and find the first feature vector using equation (13). The second feature vector was calculated by advancing the sample window by 50 data points. From equation (13) we can see that the dimension of feature vectors is 15. To display high dimensional driving cycle feature vectors, the Principal Component Analysis (PCA) algorithm was applied to the generated representative feature vectors. The first and second principal components were then used to plot representative feature vectors in 2-dimensional space. Figure 21 shows the distribution of representative feature vectors for the selected four driving cycles.

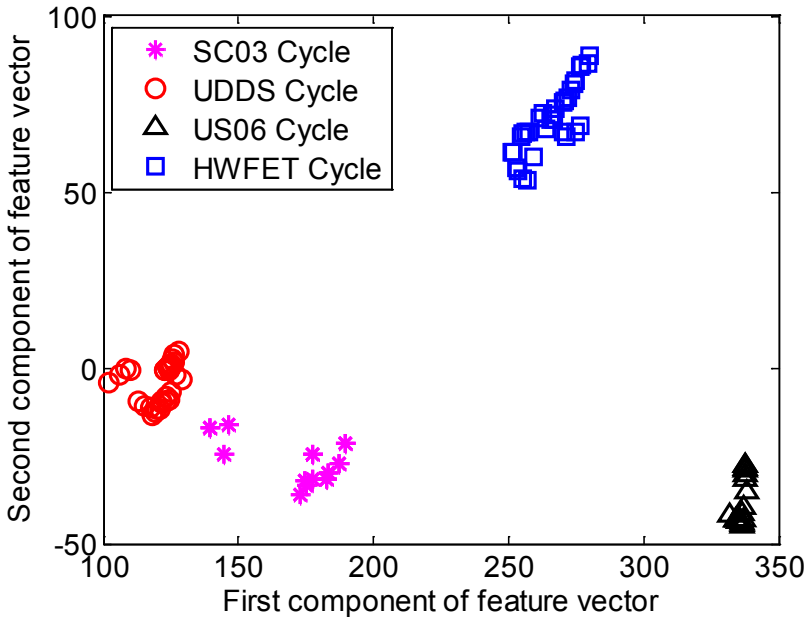

Figure 21. The distribution of the feature vectors of selected four driving cycles.

3.3. **Classify real world driving cycles using representative feature vectors**

To validate the effectiveness of representative feature vectors for real-world driving cycle pattern recognition, two real-world driving cycles were adopted for the performance test. The two real-world driving cycles were collected by a mild-hybrid Chevy Malibu driven in the urban and suburban area near the downtown Hancock. The routes of real-world city cycle (RW-CC) and real-world highway cycle (RW-HC) are shown in Figure 22 and Figure 23. RW-CC and RW-HC was logged at 1hz speed for 466 and 1046 seconds respectively. The speed profiles for these two driving cycles are shown in Figure 24 and Figure 25. To classify real-world driving cycles to one of the selected four driving cycles, the feature vectors for the RW-CC and RW-HC were generated using the feature extraction method described in Section 3.2.2. The classification is to identify to which pattern the test pattern

belongs. The k-Nearest Neighbor (*k*NN) algorithm was employed for the driving cycle classification. For a test feature vector x, the nearest neighbor rule is summarized as follows. (1) Calculate the distances of the test feature vector x to each of representative feature vectors shown in Figure 21. (2) Identify the k nearest neighbors of representative feature vectors to the vector x. The number of k is general not to be a multiple of the number of classes M. (3) Out of these k samples, identify the number of vectors, k_i, that belong to class ω_i, $i = 1, 2, \cdots\cdots, M$, $\sum_i k_i = k$. (4) Assign x to the class ω_i with the maximum number k_i of samples. In the real-world driving cycle test, the value of M is 4 and the number of k is chosen to be 13. The distances between the test feature vector and representative feature vectors was calculated by the Euclidean distance.

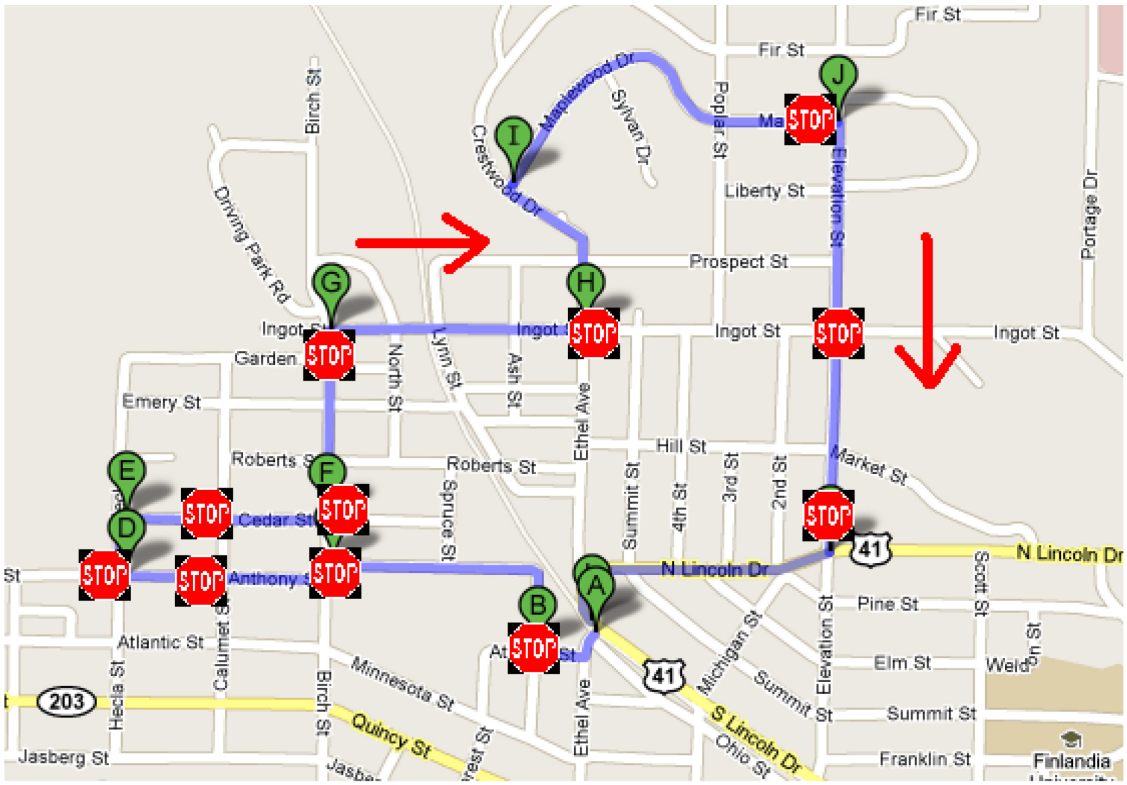

Figure 22. The route of city cycle.[57]

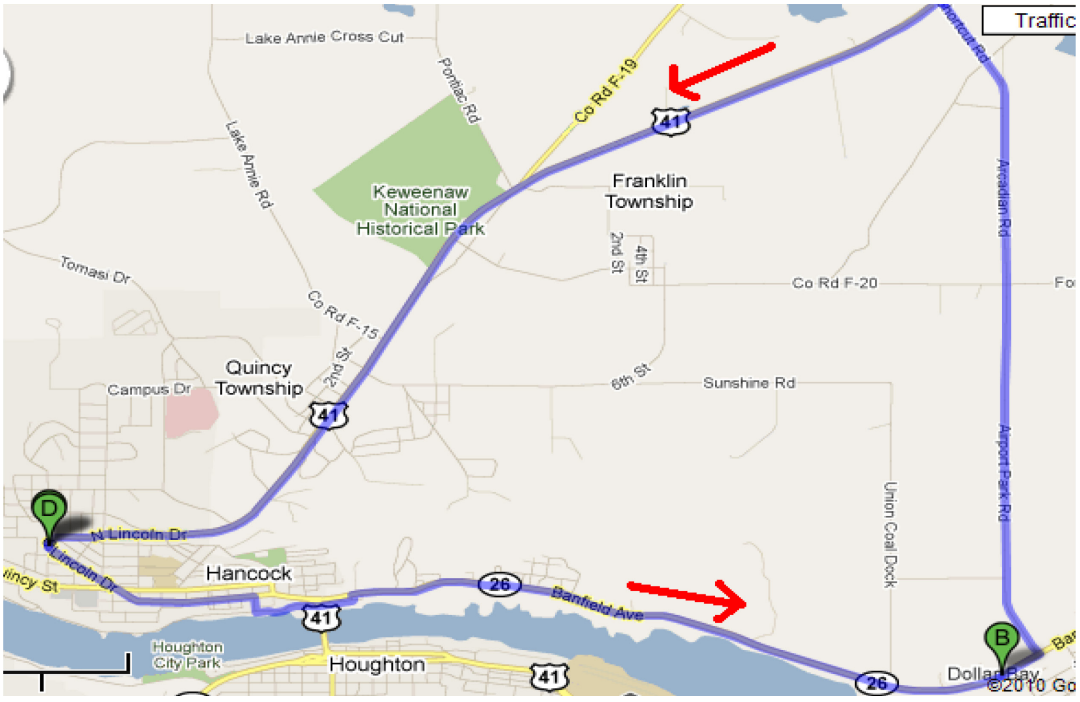

Figure 23. The route of highway cycle.[57]

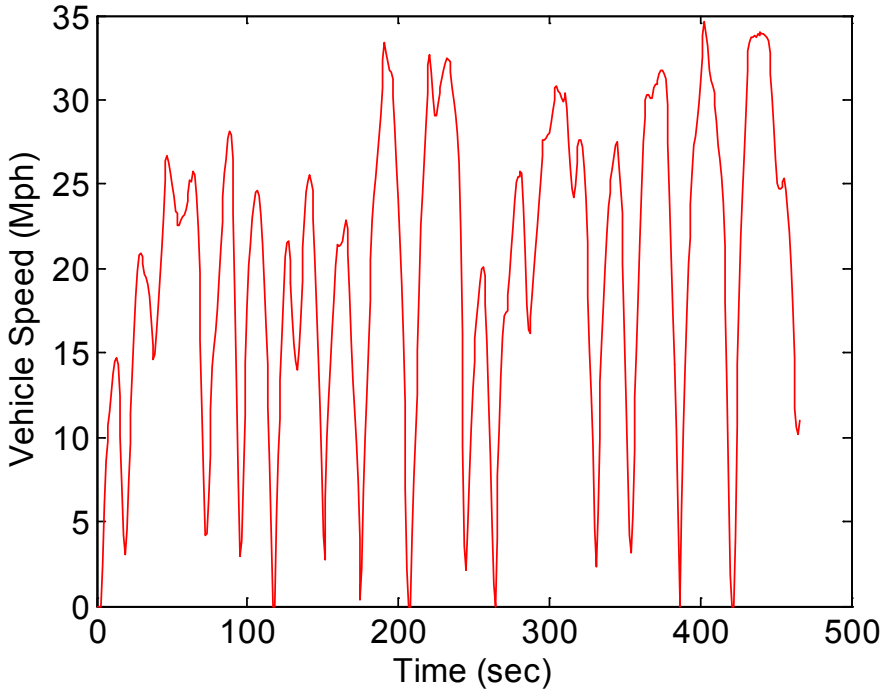

Figure 24. The speed profile of city cycle.

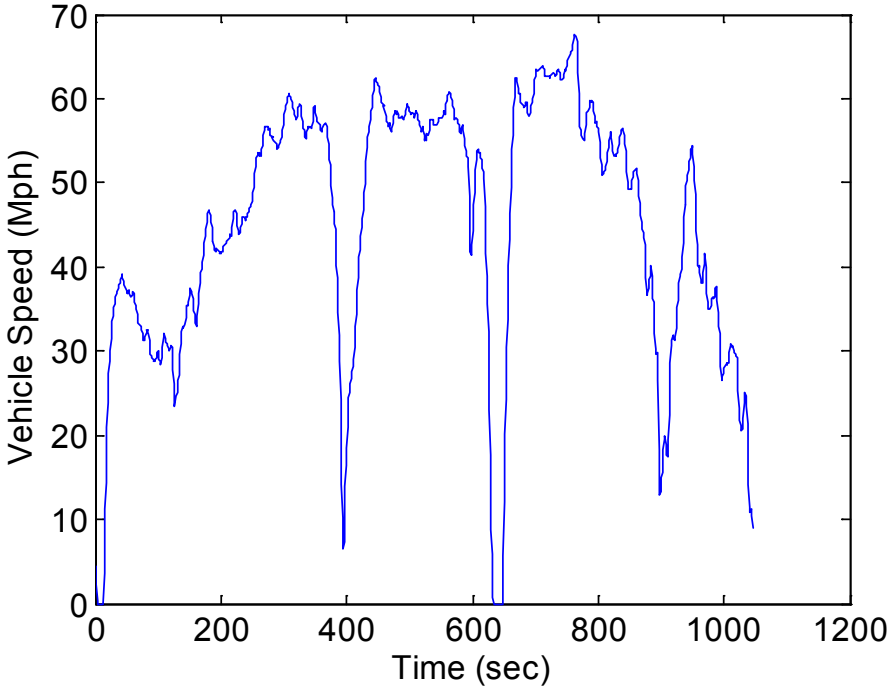

Figure 25. The speed profile of highway cycle.

The success rates of classifying the RW-CC to the UDDS category and the RW-HC to the HWFET are shown in Figure 26. In the classification test, the success rate of driving cycle pattern recognition was calculated with different size of the sample window. The tested sample window sizes include 50, 100, 150, 200, 250, 300, 350, 400, and 450. Figure 26 show that the larger size of the sample window has a higher pattern recognition success rate for both RW-CC and RW-HC. In addition, the success rates of the RW-CC are higher than RW-HC. This is due to the fact that the similarity of the RW-CC with the UDDS is higher than the similarity of the RW-HC with the HWFET.

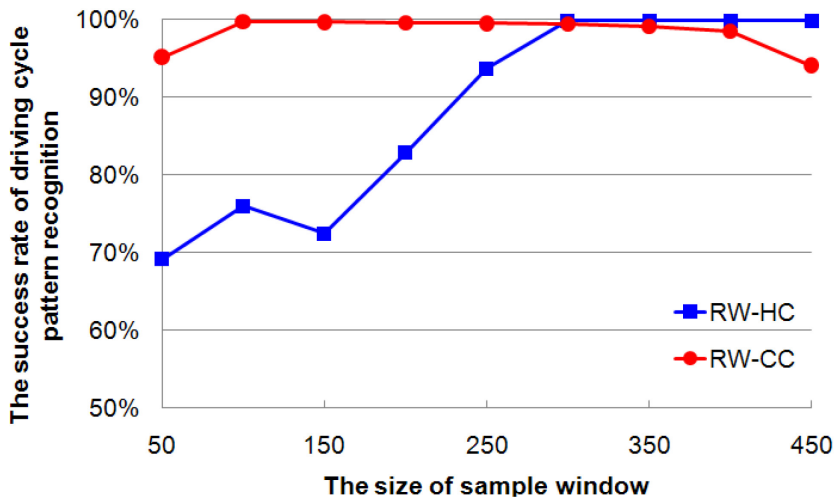

Figure 26. The success rate of real world driving cycle pattern recognition.

3.4. The impact of dissimilarity measures and feature extraction methods on driving cycle pattern recognition

This section studies the impact of the dissimilarity measures and the feature extraction methods on the quality of representative feature vectors and the performance of the driving cycle pattern recognition.

3.4.1. The impact of dissimilarity measures on pattern recognition success rate

To test the impact of the dissimilarity measure (distance between feature vectors) on the performance of the pattern recognition, a number of dissimilarity measures were tested for the driving cycle pattern recognition. The tested similarity measures include Euclidean distance, Chebyshev distance, Cosine distance, Correlation distance, and Mahalanobis distance. Let X and Y are two feature vectors with dimension *n*. The definitions of these dissimilarity measures are given below:

- Chebyshev distance:

$$d_{XY} = \max(|X_i - Y_i|), i \in n \tag{14}$$

- Cosine distance:

$$d_{XY} = 1 - \frac{XY^T}{\left(XX^T\right)^{1/2}\left(YY^T\right)^{1/2}} \tag{15}$$

- Correlation distance:

$$d_{XY} = 1 - \frac{\left(X - \overline{X}\right)\left(Y - \overline{Y}\right)^T}{\left(\left(X - \overline{X}\right)\left(X - \overline{X}\right)^T\right)^{1/2}\left(\left(Y - \overline{Y}\right)\left(Y - \overline{Y}\right)^T\right)^{1/2}} \tag{16}$$

Where $\overline{X} = \frac{1}{p}\sum_j X_j, \overline{Y} = \frac{1}{p}\sum_j Y_j$

- The Mahalanobis distance of a multivariate vector $X = (x_1, \ x_2, \ \ldots\ldots, \ x_n)^T$ from a group of values with mean $\mu = (\mu_1, \ \mu_2, \ \ldots\ldots, \ \mu_n)^T$ and covariance matrix *S* is defined as:

$$D = \sqrt{(X-\mu)^T S^{-1}(X-\mu)} \qquad (17)$$

Figure 27 shows the success rate of RW-CC pattern recognition with various dissimilarity measures using *k*NN 13 classification method. From Figure 27 we can see that the Euclidean distance is the only dissimilarity measure that has good pattern recognition performance for the RW-CC pattern recognition. Other dissimilarity measures, including Cosine distance, Correlation distance, and Mahalanobis distance have very bad performance for the RW-CC pattern recognition. The average success rate of the Chebyshev distance is only about 60% and it fluctuates significantly. Figure 28 shows the success rate of RW-HC pattern recognition with aforementioned dissimilarity measures. The Mahalanobis distance shows the best performance of the RW-HC pattern recognition. The Euclidean distance and the Chebyshev distance also show the good performance when the sample window size is larger than 300 data points.

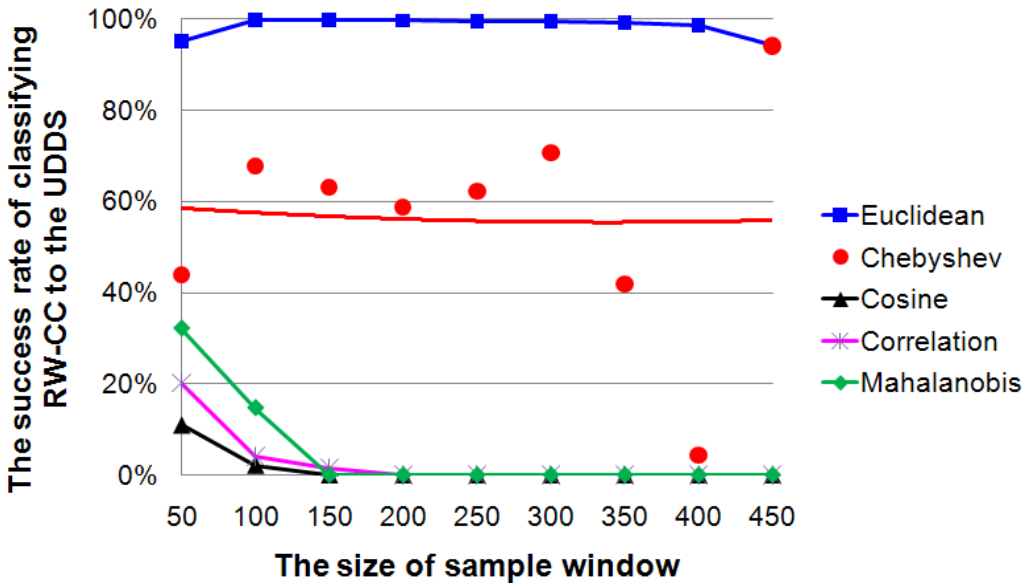

Figure 27. The success rate of real-world city cycle recognition using KNN-13 with various dissimilarity measures.

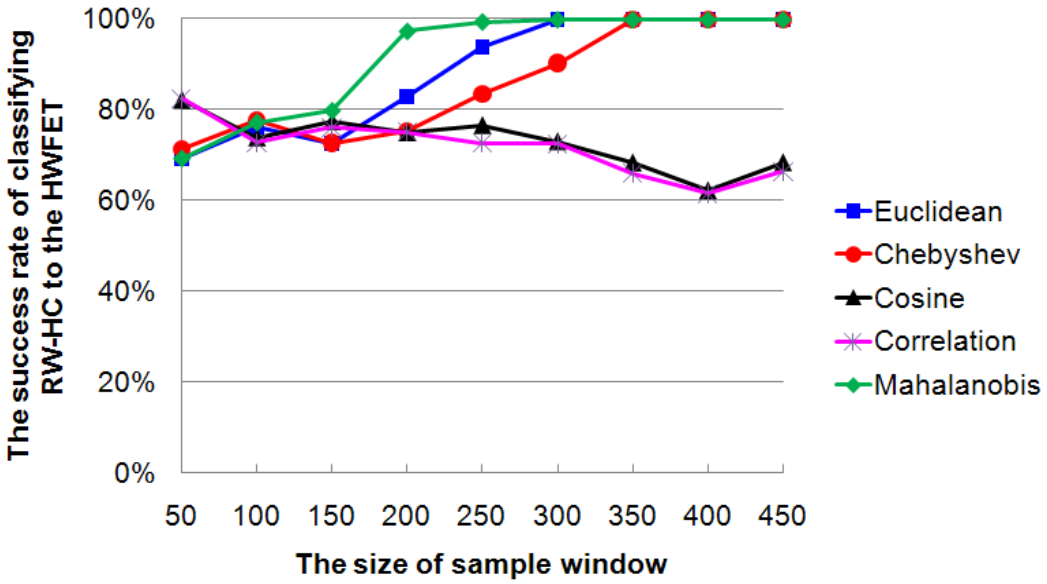

Figure 28. The success rate of real-world highway cycle recognition using KNN-13 with various dissimilarity measures.

3.4.2. The impact of feature extraction methods on the quality of representative feature vectors

Various feature extraction methods have been proposed to extract features from time series sensor data, such as Single Value Decomposition [58], Discrete Fourier Transformation (DFT) [59, 60], Discrete Wavelet Transformation (DWT) [61], Adaptive Piecewise Constant Approximation [62], Discrete Cosine Transformation [58], Chebyshev Polynomials [63], Piecewise Aggregate Approximation [62], and Symbolic Aggregate Approximation [64]. In this section, the performance of the autoregressive (AR), DFT, and DWT feature extraction methods for the driving cycle pattern recognition is studied.

3.4.2.1. Feature extraction using auto-regressive model

For the normalized driving cycle data set X , it can be fitted to an AR model of order p as

$$x_k = \sum_{i=1}^{p} \alpha_i x_{k-i} + r_k \quad k = p+1,...,n \tag{18}$$

where α_i , $i = 1,2,...,p$ is a coefficient of the AR model; r_k, $k = p+1,...,n$ is the residual between the driving cycle data and the AR model value. The vector $f(X) = \left(\alpha_1, \alpha_2, ..., \alpha_p\right)^T$ can be used as the feature vector of the normalized data X .

3.4.2.2. Feature extraction using discrete Fourier transform

Discrete Fourier Transform is one of the techniques for dimensionality reduction using spectral decomposition. In this study, the DFT coefficients of the four driving cycle data vary a lot within the frequency range of 0-0.5Hz. The frequency range of 0-0.5Hz was equally divided into 5 small ranges, each of which has 0.1Hz bandwidth. In each small range, the frequency with largest amplitude was selected as a feature member. The mean value of the DFT amplitudes in each small frequency range was also selected as a feature

member. As such, the feature vector was formed by frequencies $f_1 - f_5$ and the mean amplitudes $a_1 - a_5$ as shown below

$$f(X) = (f_1,\ k \times a_1,\ f_2,\ k \times a_2,\ f_3,\ k \times a_3,\ f_4,\ k \times a_4,\ f_5,\ k \times a_5)^T \quad (19)$$

where *k* is the weighting factor of the amplitudes.

3.4.2.3. Feature extraction using discrete wavelet transform

Discrete wavelet transform decomposes a signal into layers of coefficients. These coefficients contain both frequency and time domain information. Given a time series x with the length of n, the discrete wavelet transform of x is calculated by passing the time-series data through a series of low pass and high pass filters. The outputs from the high pass filter are called detail coefficients while the outputs from the low-pass filter are called approximation coefficients. The approximation coefficients are further decomposed in the next iteration while the detail coefficients are kept as the current level wavelet coefficients. To form feature vectors from wavelet coefficients, feature extraction method proposed in [65] was employed, which consists of two steps: cluster determination and feature determination. The cluster determination process determines the boundary of each cluster in DWT coefficients matrix, while the feature determination step calculates each element of the feature vector using the Euclidean norms of coefficients in each cluster.

3.4.3. Success rate of the pattern recognition using AR, DFT, and DWT feature extraction methods

Autoregressive, discrete Fourier transformation, and discrete wavelet transformation feature extraction methods were applied to the UDDS, HWFET, US06, and SC03 driving

cycle data. The generated representative feature vectors for these four driving cycles are shown in Figure 29, Figure 30, and Figure 31. The features vectors generated using AR, DFT, and DWT methods are not well separated. With these representative feature vectors, the success rates of driving cycle pattern recognition were tested using *k*NN-9 classification method and Euclidean distance. The test results are shown in Figure 32, Figure 33, and Figure 34. As we can see from these figures, the success rates, in generally, are lower than the pattern recognition success rate using representative feature vectors shown in Figure 21. Although the DFT feature extraction method has high success rates for the RW-CC pattern recognition, the success rates for the RW-HC pattern recognition is extremely low.

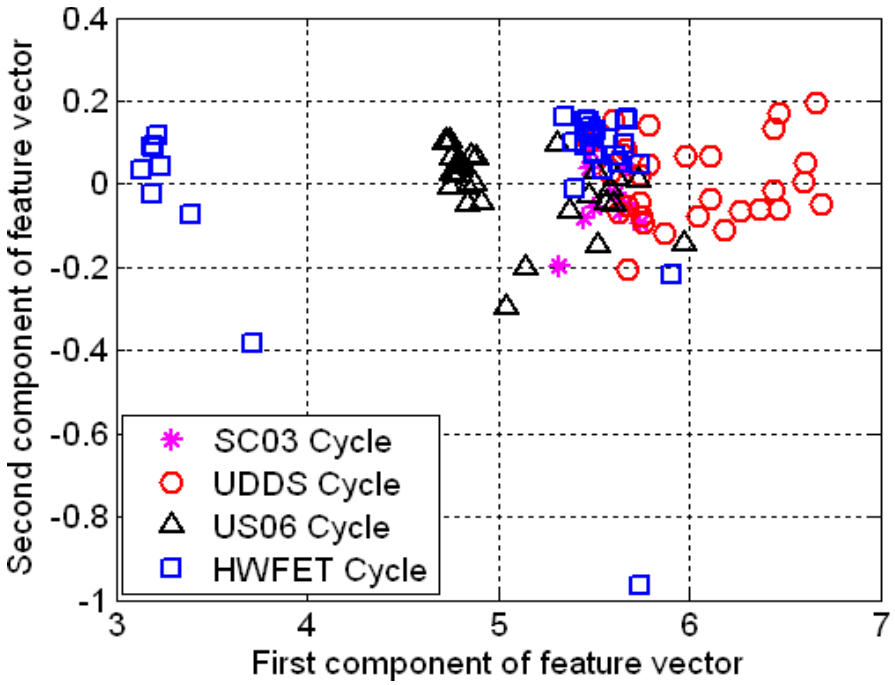

Figure 29. The distribution of DFT-based feature vectors.

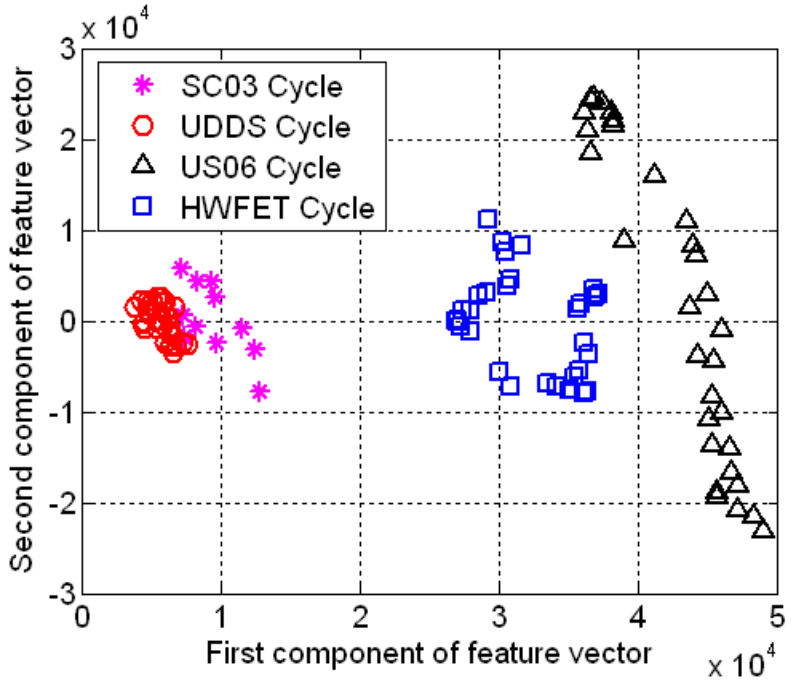

Figure 30. The distribution of DWT-based feature vectors.

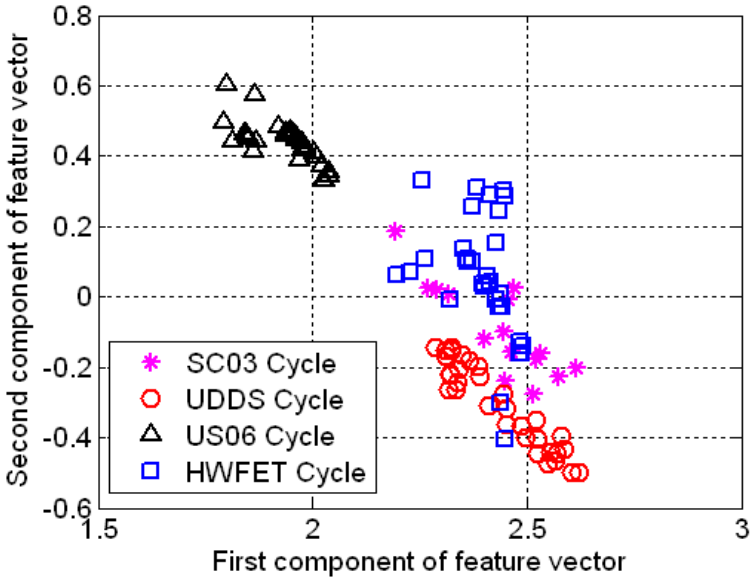

Figure 31. The distribution of AR-based feature vectors.

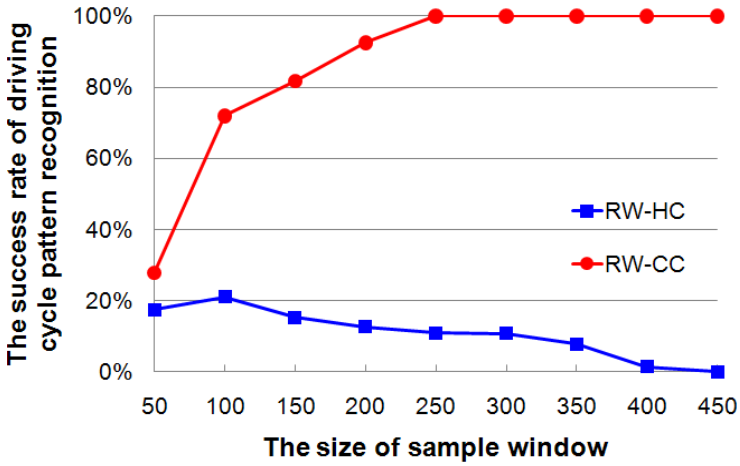

Figure 32. Success rate of driving pattern recognition using AR based feature extraction method.

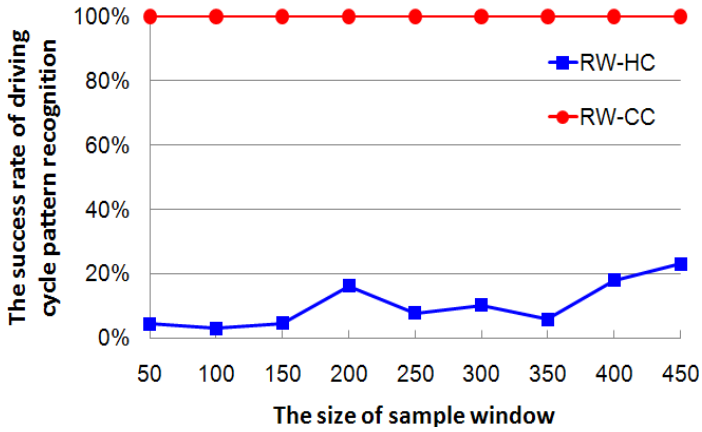

Figure 33. Success rate of pattern recognition using DFT based feature extraction method.

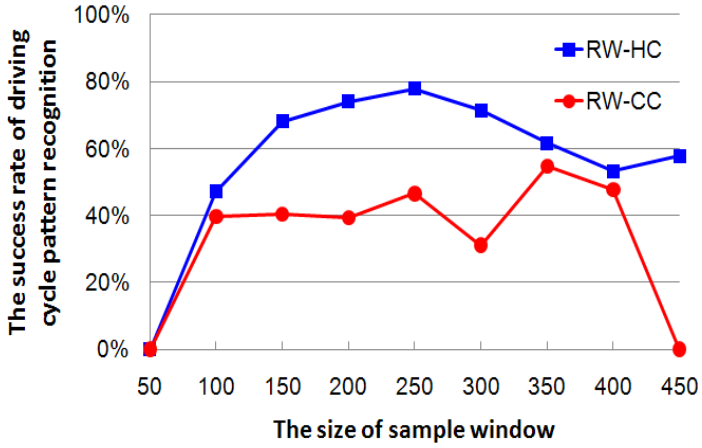

Figure 34. Success rate of driving pattern recognition using DWT based feature extraction method.

3.5. Adaptive control based on driving cycle pattern recognition

The adaptive control is achieved through the real-time driving cycle pattern recognition and dynamic change the control parameters that are optimized to the recognized driving

cycle. To implement this adaptive control strategy in a vehicle model, optimal control parameters for each representative driving cycle need to be identified. The simulation software and the vehicle model used to evaluate the improvement of fuel economy for the proposed adaptive control are the powertrain/propulsion simulation and analysis software – Autonomie [66] and the Prius MY04 model.

3.5.1. Optimized controller gains for individual driving cycles

Four controller parameters are selected to dynamically change during the fuel economy simulation for different input driving cycles. These four parameters are the proportional and integral gains of the driver controller and the proportional and integral gains of the motor 2 controller as shown in Figure 11. To find optimal controller gains for different driving cycles, 36 simulations are carried out for each representative driving cycle and 144 simulations in total are conducted for 4 federal driving cycles. When one optimal gain is being searched, the values of other 3 gains are fixed. The final optimized controller gains for four federal driving cycles are listed in Table 4.

Table 4. Default and optimized controller gains.

		Default Gains	Optimized Controller Gains			
			UDDS	US06	SC03	HIGHWAY
Controller Gains	Driver Controller Ki	0.5	0.2	0.1	0.5	0.4
	Driver Controller Kp	1000	500	500	500	500
	Motor 2 Ki	0.005	0.01	0.001	0.007	0.005
	Motor 2 Kp	0.5	0.8	0.6	0.9	0.7
Fuel Economy with default gains			72.38	43.28	69.83	63.30
Fuel Economy with optimized gains			73.56	44.03	71.67	63.35
Improved Percentage			1.63%	1.73%	2.63%	0.08%

3.5.2. The improvement of fuel economy with dynamic selection of controller gains based on driving cycle pattern recognition

To evaluate the performance of the proposed adaptive control based on the driving cycle pattern recognition, the pattern recognition algorithm is integrated with the Prius MY04 vehicle model in Autonomie. The output value of the pattern recognition algorithm for SC03, UDDS, US06, HWFET, and unclassified driving cycle (when the data points is less than 300) is defined as 1, 2, 3, 4, and 0, respectively. During simulation, a combination of three real-world city cycle (RW-CC) plus one real-world highway cycle (RW-HC) is input into the simulation. The pattern recognition result for the combined driving cycles is shown in Figure 35. As we can see from Figure 35, the pattern recognition result is 0 in the first 300 seconds (data sample rate is 1 Hz) because there is not enough data points for pattern recognition. After 300 seconds, the pattern recognition algorithm successfully classifies the input driving cycle to urban driving cycle. At 1398th second, three real-world city cycle ends and the real-world highway cycle starts. When the data points from the real-world highway cycle add to the pattern recognition data buffer, the buffer contains speed data both from real-world city cycle and real-world highway cycle. This causes the fail of pattern recognition from 1570th second to 1690th second. After 1690 seconds, most speed data in the pattern recognition data buffer are from the real-world highway cycle, as a result, the pattern recognition algorithm is able to successfully classify the current driving cycle to highway driving cycle. The output of the pattern recognition algorithm is then used to select new controller gains to which the recognized driving pattern has the highest fuel economy.

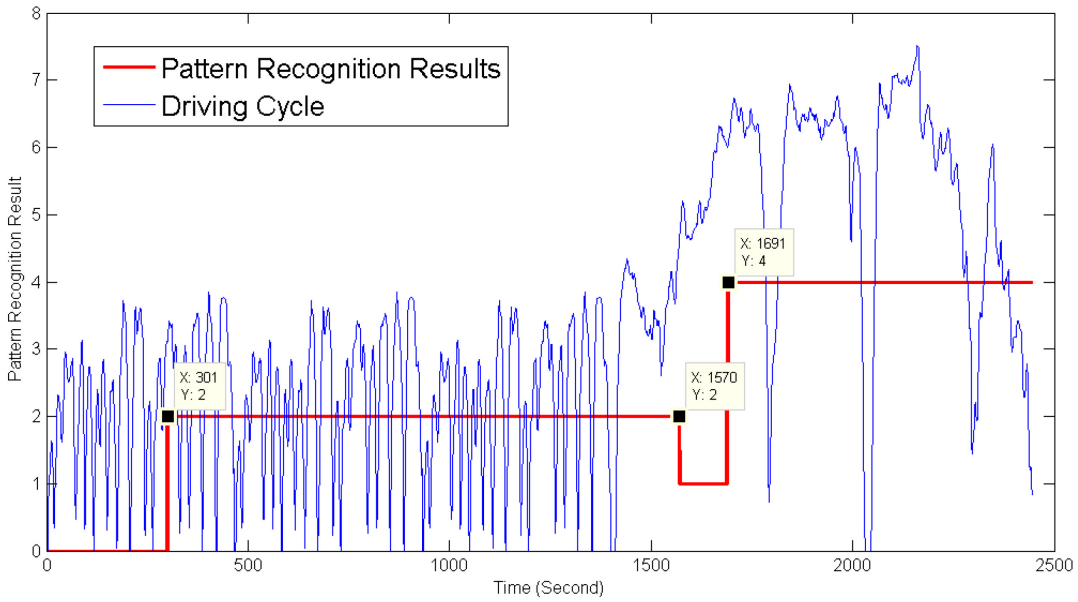

Figure 35. Pattern recognition result for real-world city cycle and real-world highway cycle.

A comparison of fuel economy with default and optimized controller gains is conducted. The input driving cycles are a combination of real-world city cycle and real-world highway cycle and a combination of four federal driving cycles. The simulation result shows that dynamic selection of controller gains based on driving cycles have better fuel economy in both cases. The improvement of fuel economy for two different combinations of driving cycles is shown in Table 5.

Table 5.The improvement of fuel economy with adaptive controller based on driving cycle pattern recognition

Driving cycle	Fuel economy with default controller gains	Fuel economy with optimized controller gains	Fuel economy improvement in percentage

	(Miles per gallon)	(Miles per gallon)	
3 RW-CC + 1 RW-HC	61.87	62.51	1.03%
SC03+UDDS+HW+US06	59.09	60.05	1.60%

.

Chapter 4 Study the Impact of Driver's Behavior on HEV Operation and Energy Efficiency[2] [67]

4.1. Introduction

Under real-world driving scenario, the driver's behavior is one of the major influence factors to the vehicle powertrain performance and fuel economy. Under normal driving circumstances, hybrid electric vehicles have a better fuel economy than conventional vehicles because electric motors are the main traction power sources which have positive impact on the overall efficiency of the vehicle. For power-split HEVs, the engine speed or engine torque becomes independent from the driver's request. Thus, with proper control, the engine operating region could be placed within an optimal region and the better fuel economy will be achieved. However, the scenario described above only happens when the driver's power request is within a reasonable range and the most of the power request could be provided by the electric motors. If the driver's behavior shows aggressive driving style, the power request for the powertrain increases rapidly. Usually, this kind of instantaneous power request peaks will result in frequent starting and stopping of the engine and pushing the engine operation to an inefficient region.

In this chapter, a systematic study of the impact of aggressive driving on the HEV fuel economy, powertrain energy losses, the operating region of powertrain components, and regenerative braking, has been performed. The rest of the chapter is organized as follows. Section 4.2 introduces the modeling of aggressive driving cycles. Section 4.3 discusses the

[2] ASME©. Portions reprinted with permission, from Feng, L. and Chen, B., *Study the Impact of Driver's Behavior on the Energy Efficiency of Hybrid Electric Vehicles*. in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2013. Portland, Oregon, USA: ASME. See Appendix C for documentation of permission to republish this material

impact of aggressive driving on HEV fuel consumption. Section 4.4 presents the impact of aggressive driving on powertrain energy loss. Section 4.5 talks about the impact of aggressive driving on engine operation. Section 4.6 illustrates the impact of aggressive driving on regenerative braking and braking energy loss. Section 4.7 shows the impact of aggressive driving on motor energy loss.

4.2. **Modeling aggressive driving cycles**

To examine the impact of aggressive driving on HEV fuel consumption, aggressive driving cycles are modeled to reflect various level of driving aggressiveness. Since the instantaneous acceleration/deceleration of driving cycles is one of the important factors of the driving aggressiveness, the modeling of aggressive driving cycles can be achieved in several ways, for example, scaling vehicle speed, time, or the rate of speed based on baseline driving cycles. In this study, aggressive driving cycles are modeled by multiplying vehicle speed by a scale factor whereas the driving cycle time is divided by the scale factor as shown in equations (**20**) and (**21**),

$$\lambda = \frac{v_{cycle}^{aggressive}(t)}{v_{cycle}(t)} \quad (\mathbf{20})$$

$$T_{cycle}^{aggressive} = T_{cycle}/\lambda \quad (\mathbf{21})$$

where $v_{cycle}^{aggressive}(t)$ is the speed profile of the aggressive driving cycle; $v_{cycle}(t)$ is the speed profile of the baseline driving cycle; $T_{cycle}^{aggressive}$ is the length of the aggressive driving cycle; T_{cycle} is the length of the baseline driving cycle; and λ is the scale factor. Figure 36 shows the comparison between the baseline UDDS driving cycle and the

aggressive UDDS driving cycle with λ=1.5. The cycle with λ=1.5 has higher overall speed, while the time length of aggressive driving cycle is less than the baseline driving cycle. Applying the same λ value on both time and vehicle speed maintains the distances of the two driving cycles the same.

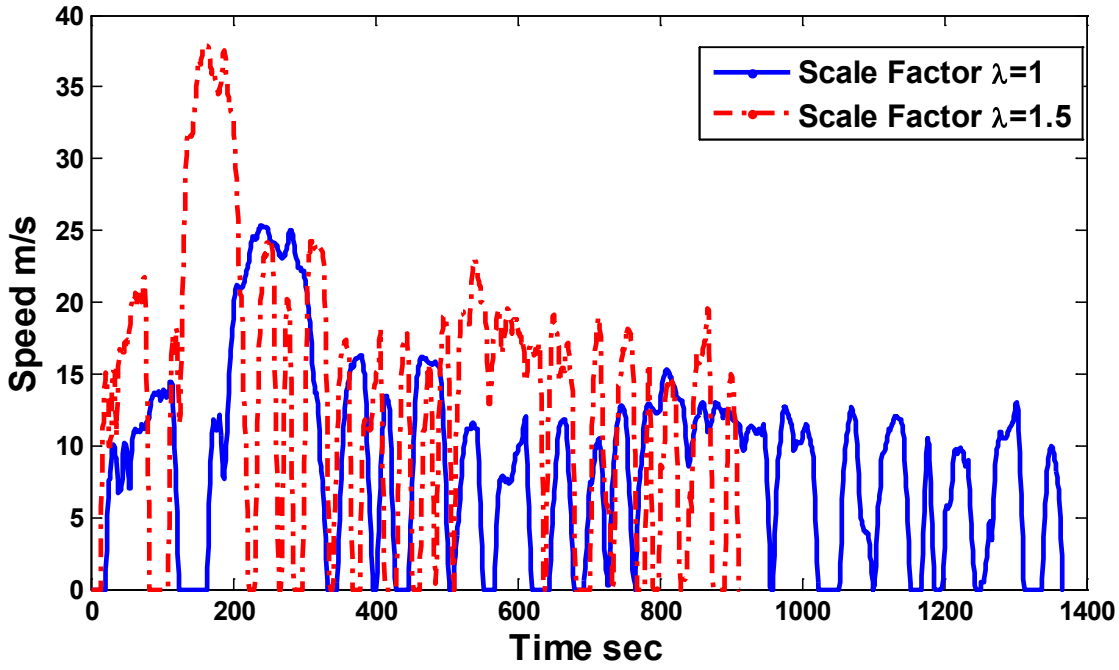

Figure 36. UDDS driving cycle with scale factor λ=1 and λ=1.5.

4.3. The impact of aggressive driving on HEV fuel consumption

Three baseline driving cycles are selected to investigate the impact of aggressive driving on HEV fuel consumption. These three driving cycles are UDDS, HWFET, and US06. The investigation employs powertrain simulation and analysis software – Autonomie[68]. Autonomie is a plug-and-play powertrain and vehicle analysis software developed by the Argonne National Laboratory for the rapid evaluation of new powertrain technologies. The open architecture of this design environment allows users to integrate customized component models, optimization algorithms, and control strategies with

vehicle models and evaluate fuel economy improvement under dynamic testing conditions. The simulation outputs include information for characterizing driver's behavior and the energy efficiency of the powertrain components, including driver (accelerator pedal position, brake pedal position, torque demand, and etc.), engine (fuel consumption, mass flow rate, engine speed and torque, efficiency, energy in/out/loss, power in/out/loss, and etc.), battery (SOC, power in/out/loss, and current/voltage output), and motor (motor speed/torque, load, power in/out/loss, and etc.).

The vehicle dynamic model used in simulation is the "Split Compact Single Mode 2wd Prius MY04," in Autonomie. The vehicle parameters of the major components of Prius MY04 are listed in Table 1.The detailed information for this vehicle model can be found in Autonomie software.

The equivalent fuel economy for three driving cycles with scale factors from 1 to 1.5 is simulated in Autonomie. The equivalent fuel economy is calculated as a combination of electric power and fuel consumption as shown in equation (**22**),

$$MPGe = \frac{Distance}{FC + \frac{E_{EC}}{E_{gas/gal}}} \quad (\mathbf{22})$$

where FC represents the fuel consumption for an entire driving cycle; E_{EC} is the electrical energy consumption for an entire driving cycle; $E_{gas/gal}$ is the conversion factor of equivalent fuel consumption; and $E_{gas/gal}$ =33440 W.h/Gal.

The simulation results shown in Figure 37 illustrate that the driving cycle scale factor significantly impacts the HEV fuel consumption. For the Toyota Prius MY04 model running under UDDS driving cycle, the *equivalent fuel economy* drops from 73.3 mpg to

34.8 mpg when the driving cycle scale factor changes from 1 to 1.5. Similar trends are observed for the HWFET and US06. When the driving cycle scale factor changes from 1 to 1.5, the *equivalent fuel economy* drops from 64.2 mpg to 37.4 mpg for the HWFET, and it drops from 43.4 mpg to 23.4 mpg for the US06. With these simulation results, the average increase of fuel consumption is about 90% for the Toyota Prius MY04 under these three driving cycles when the driving cycle scale factor changes from 1 to 1.5. This is a huge increase of fuel consumption.

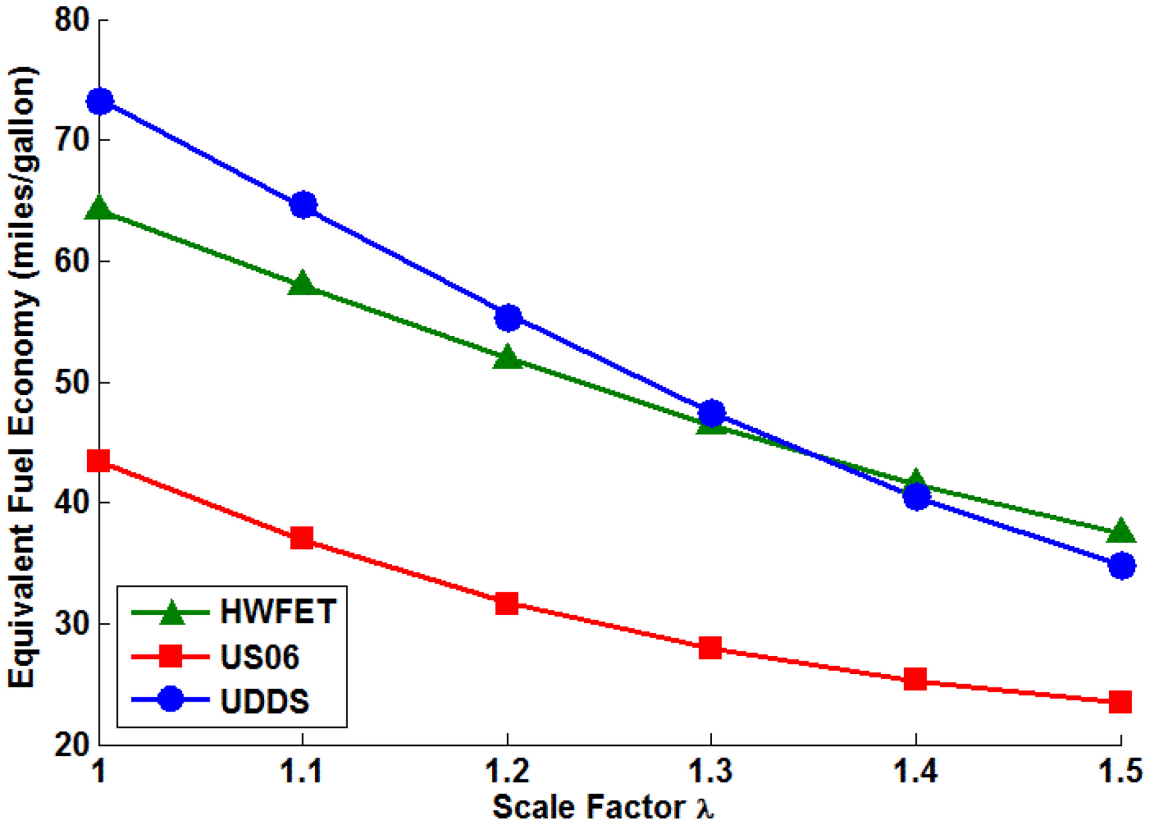

Figure 37. The impact of aggressive driving on HEV fuel consumption

4.4. **The impact of aggressive driving on powertrain energy loss**

To study the major energy losses in powertrain, the overall powertrain energy loss, the engine energy loss, the operating regions of engine, the motor energy loss, the mechanical brake energy loss, and the energy recovered by regenerative braking under the UDDS with various scale factors are examined. For the Toyota Prius MY04, the detailed simulation results can be found in **Appendix A**. Figure 38 and Figure 39 show two engine BSFC maps for λ=1.0 and λ=1.5. From these two engine BSFC maps, it is observed that the engine operates within a higher fuel rate region when λ=1.5 comparing to the fuel rate map at λ=1.0. This will result in the increase of fuel consumption. Figure 40 shows the engine and powertrain energy losses and energy recovered from regenerative braking. Figure 40 illustrates that the majority of the powertrain energy loss (green bars) is due to the engine energy loss (red bars). The engine energy loss increases when the scale factor increases. However, the energy recovered by regenerative braking (blue bars) does not have a significant change.

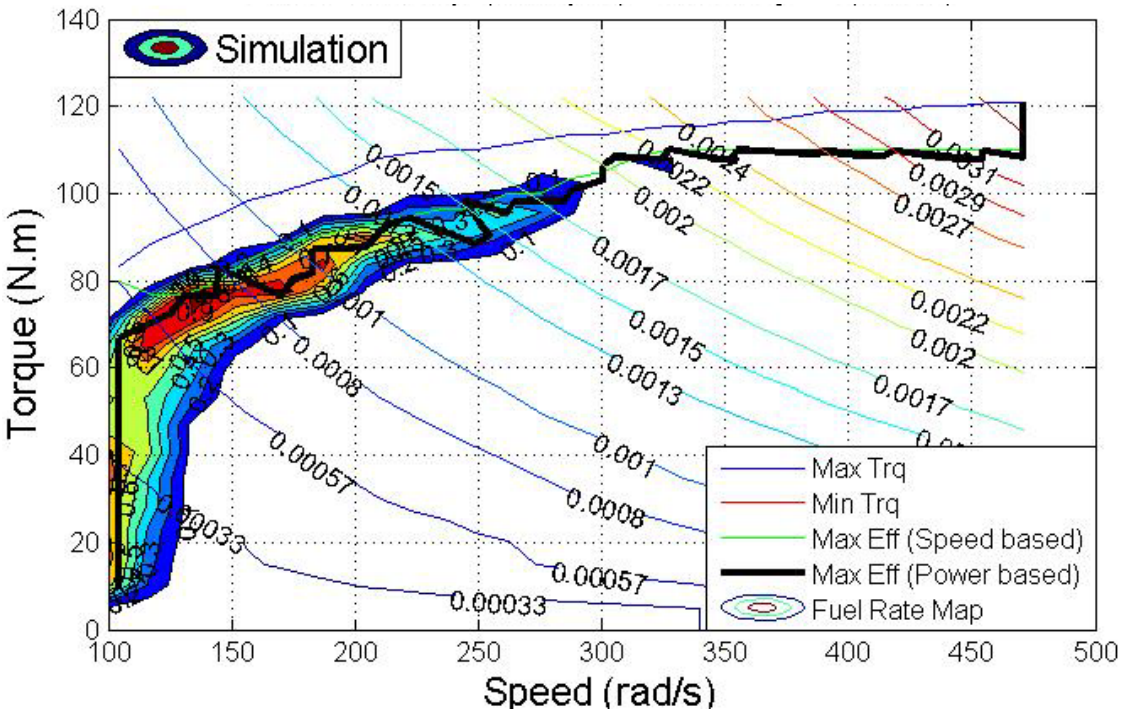

Figure 38. Engine operating region on BSFC map with λ=1

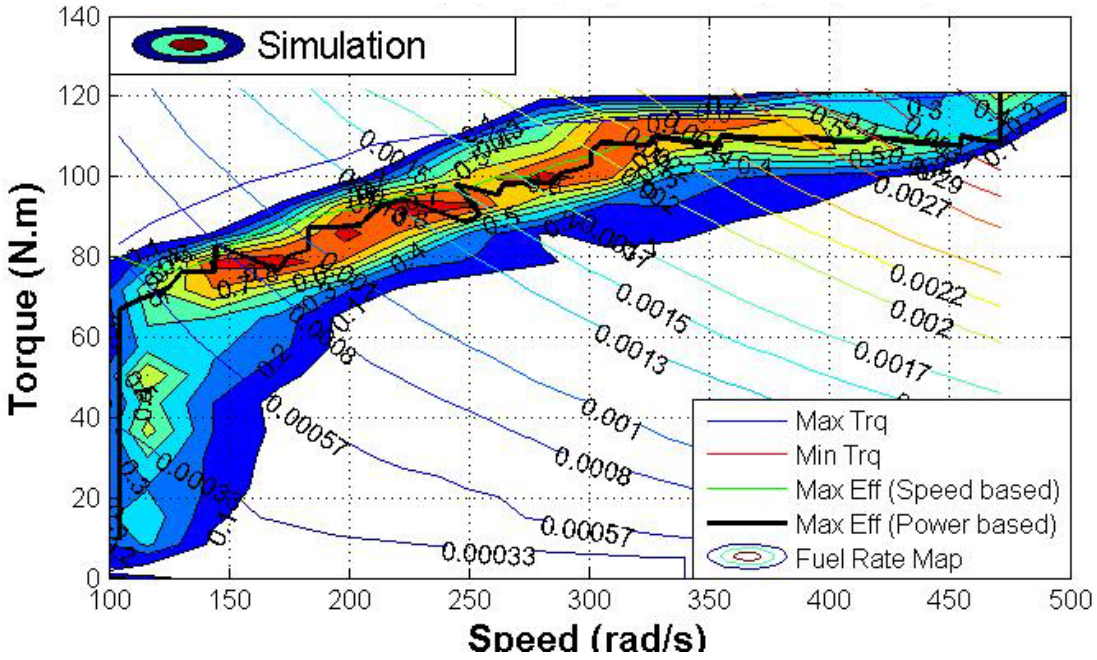

Figure 39. Engine operating region on BSFC map with λ=1.5

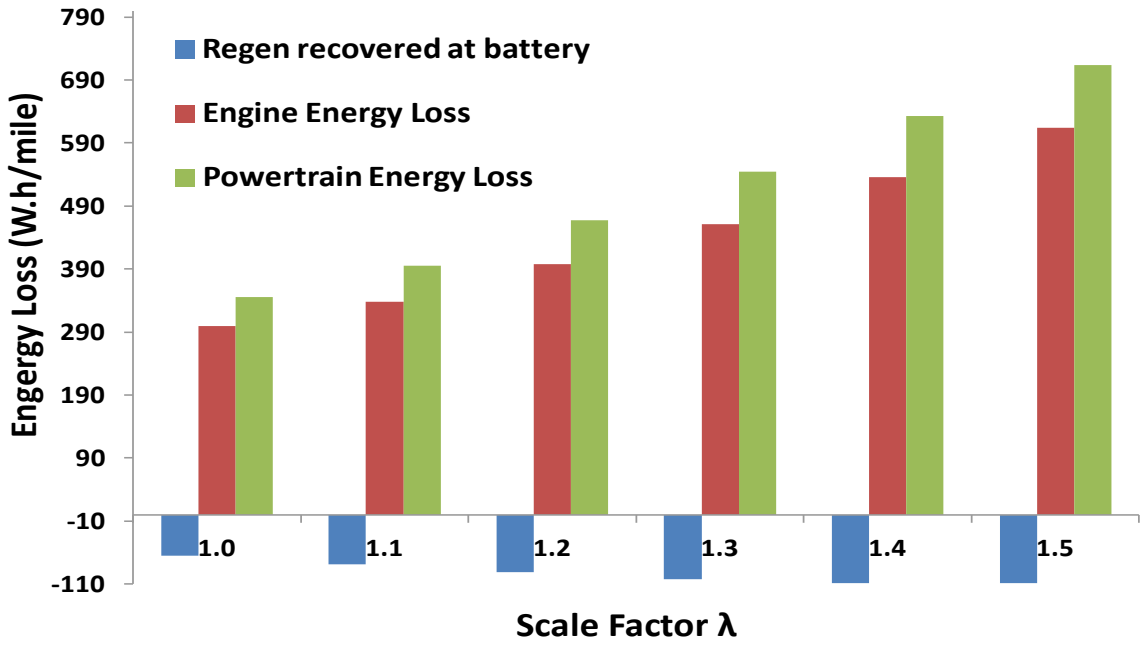

Figure 40. Engine and powertrain energy losses and energy recovered from regenerative braking.

The powertrain energy loss can be calculated by equation (**23**). The powertrain energy loss consists of engine energy loss, battery energy loss, motor energy loss, wind drag, friction energy loss, and brake energy loss.

$$P_{Powertrain\_loss} = P_{eng\_loss} + P_{bat\_loss} + P_{mot\_loss} + P_{Wind\_drag} + P_{Friction\_Loss} + P_{Brake\_loss} \quad (23)$$

Equation (**24**) shows the relationship of engine power input and power out. The power input to the engine is the product of the instantaneous fuel flow rate, $\dot{m}_f$, and the lower heating value of gasoline, Q_{LHV}. P_{eng} is the power output of the engine. Thermal power loss and mechanical power loss are both considered in this equation.

$$P_{eng\_loss} = \dot{m}_f * Q_{LHV} - P_{eng} \quad (24)$$

4.5. The impact of aggressive driving on engine operation

To find the causes for the increase of engine energy loss at a high driving cycle scale factor, further simulation is performed to study the engine power request, the changing rate of the engine power request, and the output of the engine torque and power. Figure 41 shows the changing rate of engine power request for the Toyota Prius MY04 under UDDS driving cycle with scale factor $\lambda = 1$ and $\lambda = 1.5$. The increase of scale factor causes engine to be frequently turned on and off. This is due to the instantaneous power request peaks. For a rule-based control of power-split vehicle, the engine needs to start when the power request for the powertrain is higher than the value that the traction motor is able to provide. When the value of λ is high, the power request is high and transient. Thus, the engine will be started and shut down within a short period of time as shown in the enlarged part in Figure 41. Engine stays at on state only for a very short time of period (around 2-3 seconds sometimes). This type of engine operation will limit the actual engine power output and cause a significant increase of engine energy loss. If the power management system (vehicle level supervisory controller) can avoid this type of operation, the overall HEV fuel efficacy will be improved.

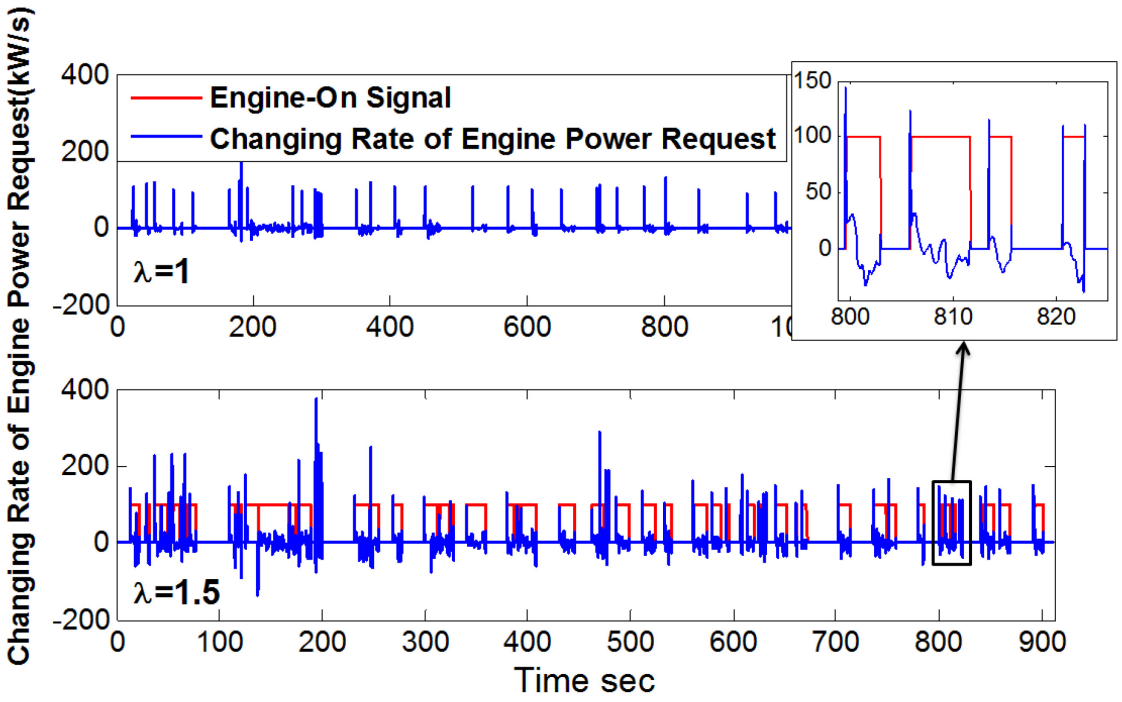

Figure 41. Engine power change rate for Prius MY04 under UDDS driving cycle with $\lambda = 1$ and $\lambda = 1.5$

4.6. **The impact of aggressive driving on regenerative braking and braking energy loss**

Aggressive driving will cause large braking power peaks during quick deceleration. In HEVs, the braking power can be recovered by regenerative braking. Due to physical constraints, however, the amount of braking energy recovered in battery is limited. If the braking power is less than the maximum charging power of a battery, it can be completely recovered by battery. When the braking power exceeds this maximum value, the excess amount of kinetic energy which is requested by a driver will be absorbed by a mechanical brake. Equation (**25**) shows the relationship between the mechanical brake power, the regenerative braking power and the braking power requested by the driver.

$$P_{brake} = P_{mech} + P_{regen} \tag{25}$$

Figure 42 shows the braking power at wheel, the maximum charging power of the battery, and the recovered power in battery for λ=1.0 and λ=1.5. When λ=1.0, the instantaneous braking power is less than the maximum charging power of the battery at most times. As a result, the majority of the braking power can be recovered by regenerative braking. When λ=1.5, large braking power peaks are occurred as shown in the lower plot of Figure 42. When the braking power is greater than the maximum charging power of the battery, the excess amount of power is wasted in mechanical brake as heat.

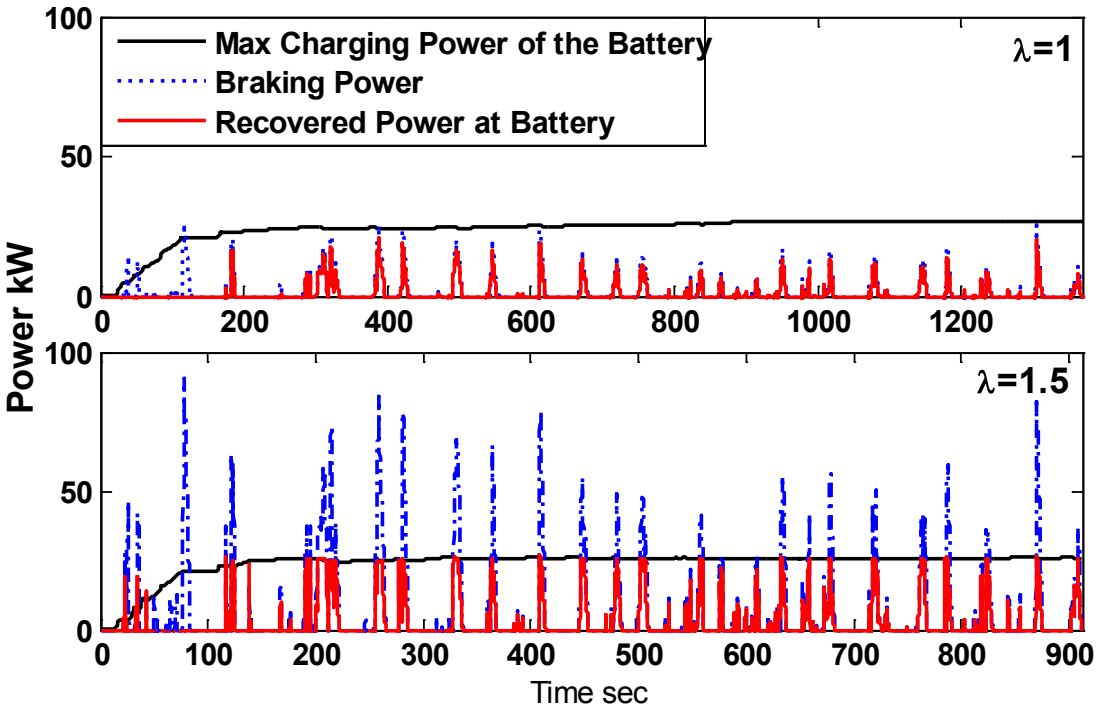

Figure 42. Available braking power at wheel, recovered power, and the maximum charging power of the battery.

The power generated by the traction motor/generator could not be fully recovered in the battery. The recovered power can be calculated by equation below.

$$P_{recover} = \begin{cases} P_{regen}\eta_{pc}\eta_{mot}\eta_{bat} \;\; if \; P_{regen}\eta_{pc}\eta_{mot}\eta_{bat} < P_{maxcharging} \\ P_{maxcharging} \;\; if \; P_{regen}\eta_{pc}\eta_{mot}\eta_{bat} \geq P_{maxcharging} \end{cases} \quad (\mathbf{26})$$

When calculating the amount of energy recovered in the battery, the efficiency of every component in the electrical path should be considered. Thus, the efficiency of the power electronics η_{pc}, the traction motor/generator η_{mot}, and the battery η_{bat} are considered in the equation (**26**). The maximum charging power of the battery is also considered in the equation (**26**). The maximum recovered power cannot exceed the maximum charging power of the battery. This is also showed in Figure 42.

Figure 43 shows the total recovered energy from regenerative braking and mechanical braking energy loss for an entire HWFET driving cycle at various scale factors. From this figure, we can see that the aggressive driving will cause a significant increase of mechanical braking energy loss due to large instantaneous braking power requests. However, the amount of regenerative braking energy recovered in battery does not change too much due to the physical constraints.

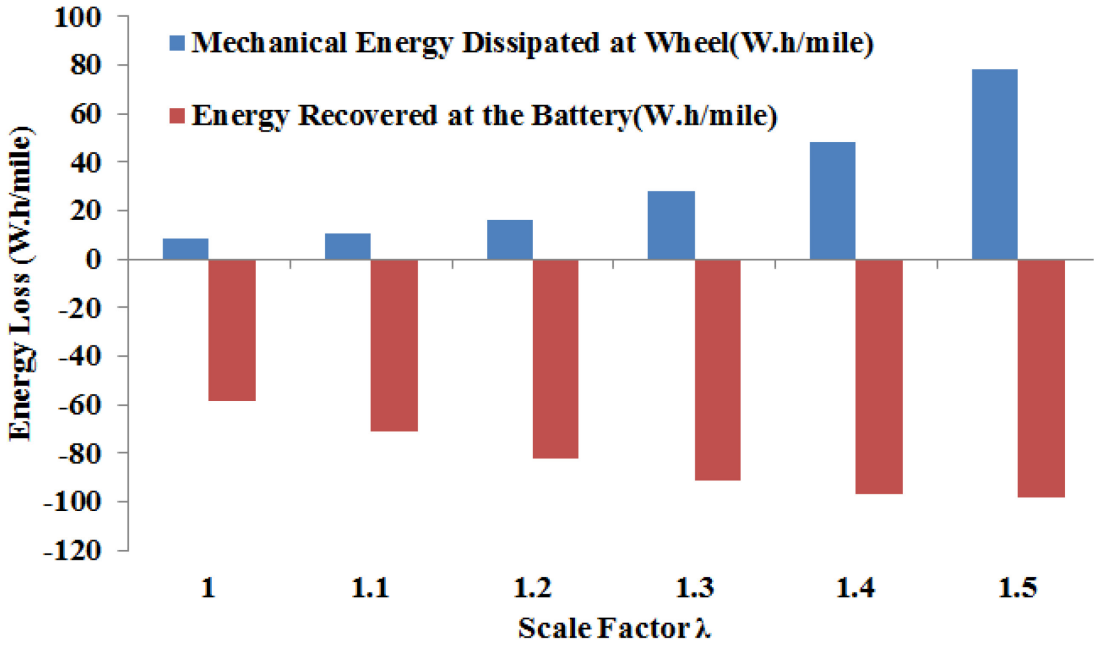

Figure 43. Total recovered energy from regenerative braking and mechanical braking energy loss under UDDS driving cycle.

4.7. The impact of aggressive driving on motor energy loss

The energy loss on the traction motor is an important part of energy consumption of a power-split hybrid vehicle. The power loss of the traction motor is given by equation (**27**). It is the product of motor input current, voltage and the motor efficiency. The power loss of the traction motor is greatly influenced by increasing of scale factor λ as shown in Figure 44.

$$P_{Loss\_motor} = I_{rms\_motor} * V_{rms\_motor} * (1 - \eta_{motor}) \quad (27)$$

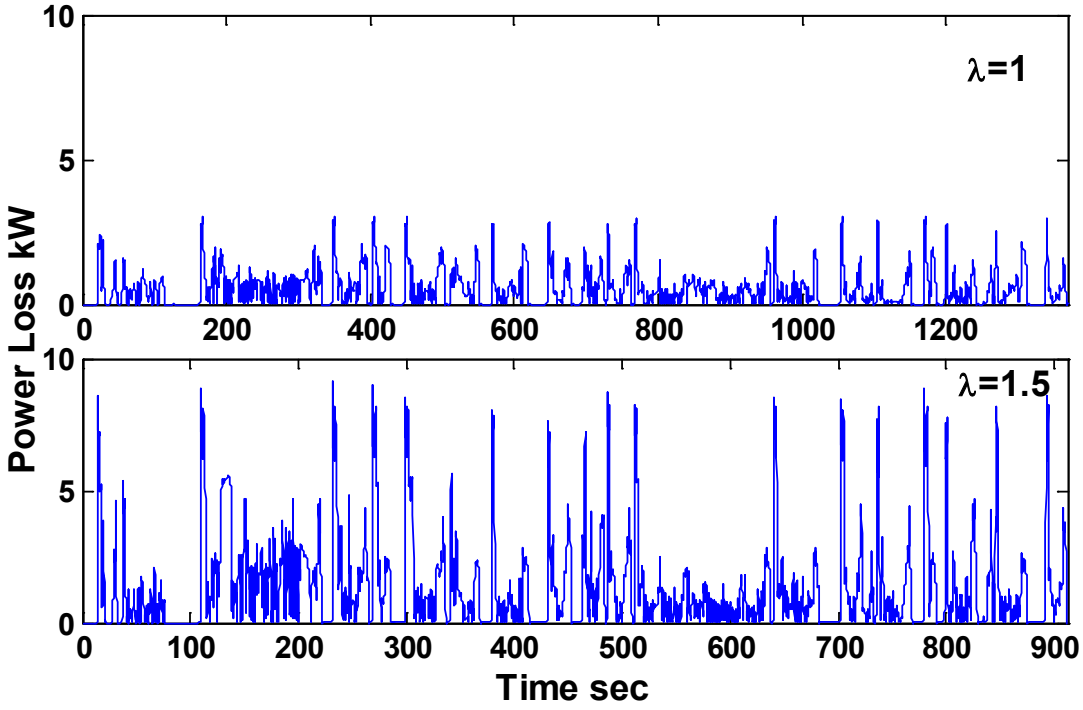

Figure 44. Traction motor power loss under UDDS driving cycle.

In the of Prius MY04 model, the power loss of the traction motor is caused by its internal resistance. High power demand from the driver when λ=1.5 requires high output power and torque from the traction motor. This causes high motor current, as a result, high power loss under aggressive driving scenario. The total energy loss on electric motor for an entire HWFET driving cycle is shown in Figure 45.

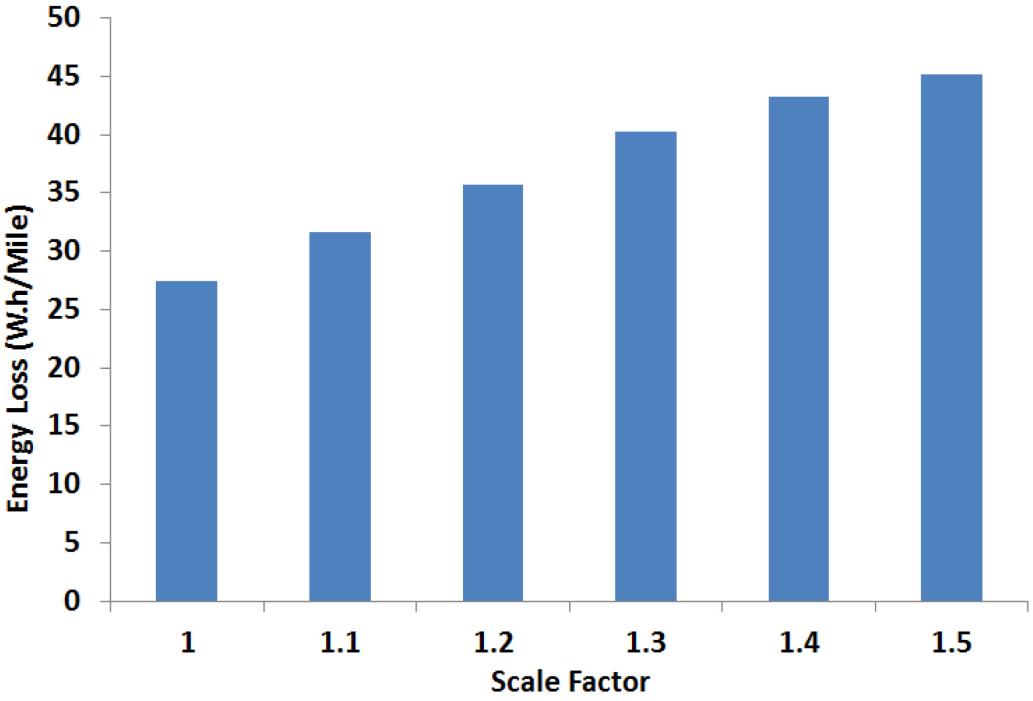

Figure 45. Total motor energy loss under UDDS driving cycle.

Chapter 5 Power-based Model Predictive Control of Power-split HEV Powertrain[3] [69]

A power-split hybrid electric vehicle has better fuel economy and less emission than a conventional internal combustion engine vehicle due to electric powertrain. The introduction of additional powertrain components, however, makes the HEV control more challenging and the performance of HEVs is more sensitive to their control strategies. In this section, a MPC controller has been developed and integrated into a MY04 Prius HEV model in Autonomie.

5.1. Background of model predictive control

This section introduces the basics of MPC. Figure 46 illustrates the receding horizon concept and the working principle of a MPC control scenario.

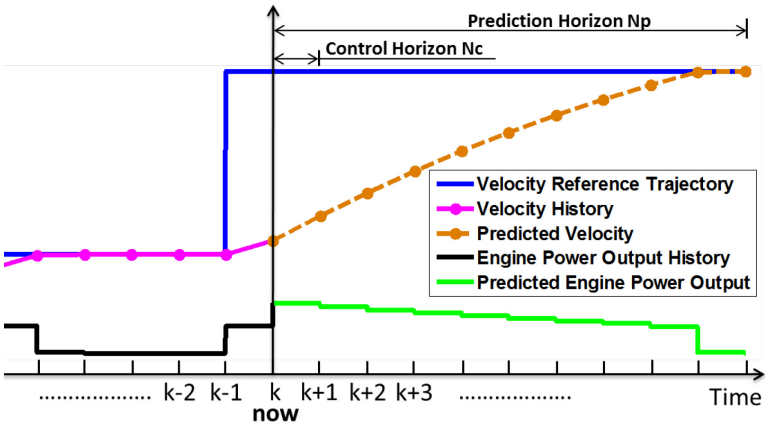

Figure 46. An example of application of MPC in vehicle control.

Figure 46 shows an SISO system in which the vehicle speed y(k) is controlled by engine power output u(k). The controller acts in every time interval Δt and the control commands are fed to a zero-order hold till next time instant. The velocity reference trajectory r(k) in

[3] SAE©. Portions reprinted with permission, from Feng, L., Cheng, M., and Chen, B., "Predictive Control of a Power-Split HEV with Fuel Consumption and SOC Estimation," SAE Technical Paper 2015-01-1161, 2015, doi:10.4271/2015-01-1161. See Appendix B for documentation of permission to republish this material.

blue line is the target of the vehicle speed. Np and Nc are the lengths of prediction horizon and control horizon, respectively. At the current moment when time t=k, the controller will estimate the future vehicle speed y(k) till Np*Δt moment and optimize the engine power output u(k) for next Nc*Δt moment based on current vehicle velocity y(k) and the internal states x(k) of the vehicle so that the vehicle will reach the velocity target r(k) as soon as possible while the change of engine power output Δu(k) is minimized. The control horizon Nc determines how many future engine power outputs u(k) will be optimized. In this case, Nc and Np both equal to 10. The green line is the optimized engine power output u(k) but only the first optimized control input, u($k|k$), will be applied. The process of optimizing u(k) and predicting y(k) will repeat at every time step based on corresponding internal states and vehicle speed. The prediction horizon and control horizon will move forward with the time and this is the reason why MPC is categorized as a receding horizon control. The orange line is the prediction of vehicle velocity y(k), based on the optimized engine power output u(k), up to the prediction horizon Np. The black line is the history of engine power output and the magenta line is the history of vehicle velocity.

Omitting slack variable ε, the cost function of an SISO MPC controller is given below.

$$\min_{\Delta u(k|k),\Delta u(k+1|k)\ldots\Delta u(k+Nc-1|k)} J = \sum_{i=0}^{Np-1}\left(\left|w_{i+1}^{y}\left(y(k+i+1|k)-r(k+i+1)\right)\right|^2 + \sum_{j=1}^{N_c}|w_{i,j}^{\Delta u}\Delta u_j(k+i|k))|^2\right) \quad \textbf{(28)}$$

Subject to

$$\Delta u_j^{min}(i) \leq \Delta u_j(k+i|k) \leq \Delta u_j^{max}(i)$$

$$y^{min}(i) \leq y(k+i|k) \leq y^{max}(i)$$

$$\Delta u(k+h|k) = 0$$

where i=0, …, Np-1 and h=Nc, …, Np-1.

$w_{i+1}^{y}, w_{i,j}^{\Delta u}$ are the weighting factors and $\Delta u_j^{min}, \Delta u_j^{max}, y^{min}, y^{max}$ are the values of the constraints.

The cost function described above consists of two components. Associated with the vehicle example above, the first component is to minimize the difference between the target speed and actual vehicle speed. The second component is to minimize the changing rate u(k) so that the engine will not experience transient power peak. By minimizing the output of the cost function J, the optimal values for $\Delta u(k|k),\ \Delta u(k+1|k), \ldots., \Delta u(k+Nc-1|k)$ are found by Quadratic Programming (QP).

5.2. Vehicle plant model linearization

One of the challenges to apply MPC to vehicle powertrain control is the nonlinear relationship between engine, motor output and SOC. If the linearization is done every time step, it increases computational load significantly. To reduce processing time, a linearized vehicle plant model has been used so that the complicated nature of vehicle powertrain is simplified. In this section, all the parameters, quasi-static maps, and simulations are obtained from MY04 Prius model in Autonomie (version# 1210)[68].

In order to utilize the prediction ability of MPC controller and control battery SOC better, the relation between battery power P_{batt} and the changing rate of SOC, $\dot{SOC}$, is needed. The battery model of Prius'04 in Autonomie is an equivalent circuit model which requires an internal resistance map and battery capacity. These two information are obtained at C/3

charging rate at Argonne National Laboratory. A series of simulation has been performed for the 6.5Ah NiMH battery of MY04 Prius to find out this relation and the simulation results are given in Figure 47.

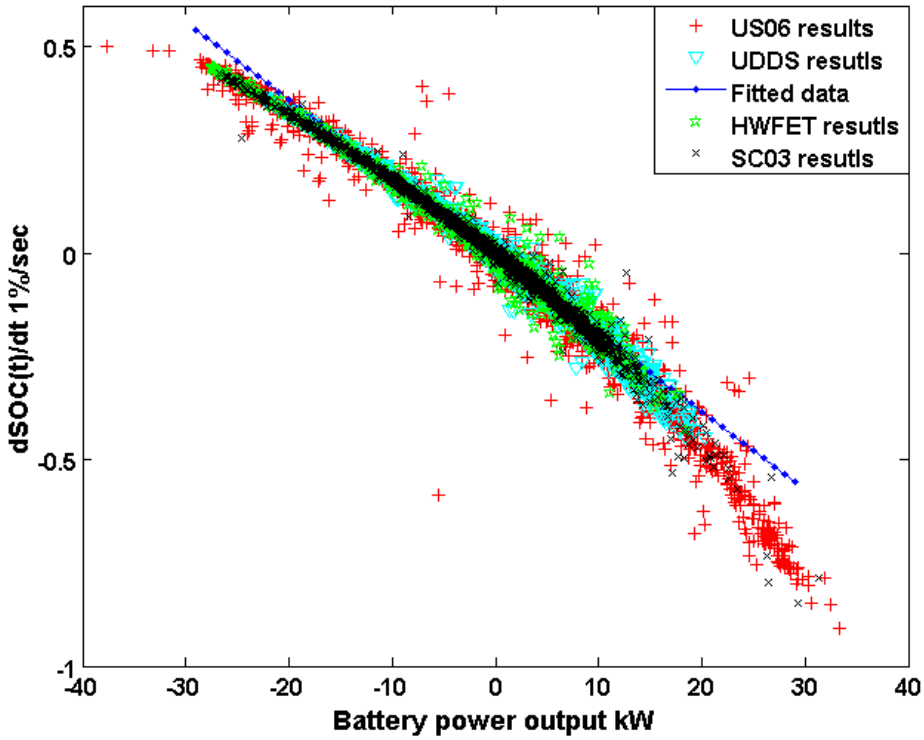

Figure 47. SȮC vs battery power P_{batt}

It can be observed from Figure 47 that $S\dot{O}C$ is almost linear to battery power output P_{batt} from -20kW to 20kW. The overlaid simulation results of 4 driving cycles show that the relation between $S\dot{O}C$ and P_{batt} mostly depends on the characteristic of the battery, not the driving cycles. A linear fitted line is calculated from the simulation results and plotted in Figure 47 and the equation of the fitted line is shown in equation **(29)**.

$$S\dot{O}C = a_1 P_{batt} + b_1 \qquad \textbf{(29)}$$

The values of a_1 and b_1 are found as,

$$a_1 = -1.89 * 10^{-2} \left(\frac{1\%}{kw * s}\right)$$

$$b_1 = -5.64 * 10^{-3} \left(\frac{1\%}{s}\right)$$

To further evaluate the accuracy of the linearized model, the simulated SOC time history and SOC estimated by (29 is compared as shown in Figure 48. Comparison results indicate that the linearized model has reasonable accuracy.

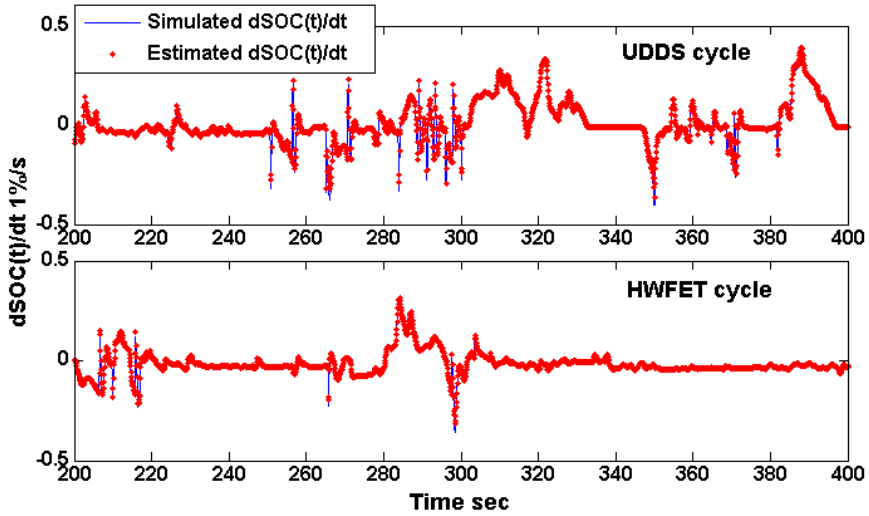

Figure 48. Estimated SOC vs simulated SOC.

The engine model of Prius'04 in Autonomie is a quasi-static engine model with steady-state BSFC map. The linearized relation between engine power output and engine fuel consumption is based on the BSFC map as well as the assumption that the engine will always operate at optimal operating points in Prius MY04 model.

In order to find the linearized relation between engine fuel consumption and engine power output, the following equation is defined.

$$\dot{m}_f = a_2 P_{eng} + b_2 \qquad \textbf{(30)}$$

P_{eng} (kw) is engine power output and $\dot{m}_f$ $(\frac{g}{s})$ is engine instantaneous fuel consumption at the given engine power output. a_2 and b_2 are the constants to be found in the following

process. One advantage of a power-split HEV is that its engine speed is independent from vehicle speed. Thus its engine could operate at the optimal operating points, at which the fuel consumption is minimized. The following discussion will show how the optimal operating points of the engine are found and prove the linear relation between the instantaneous fuel consumption rate and its corresponding engine power output at optimal operating points. The calculation is based on engine BSFC, WOT torque curve and the physical constraints of engine speed range. These data are acquired by dynamometer testing and accessible in Autonomie. The BSFC map and WOT throttle torque curve of the engine in MY04 Prius are given in Figure 49.

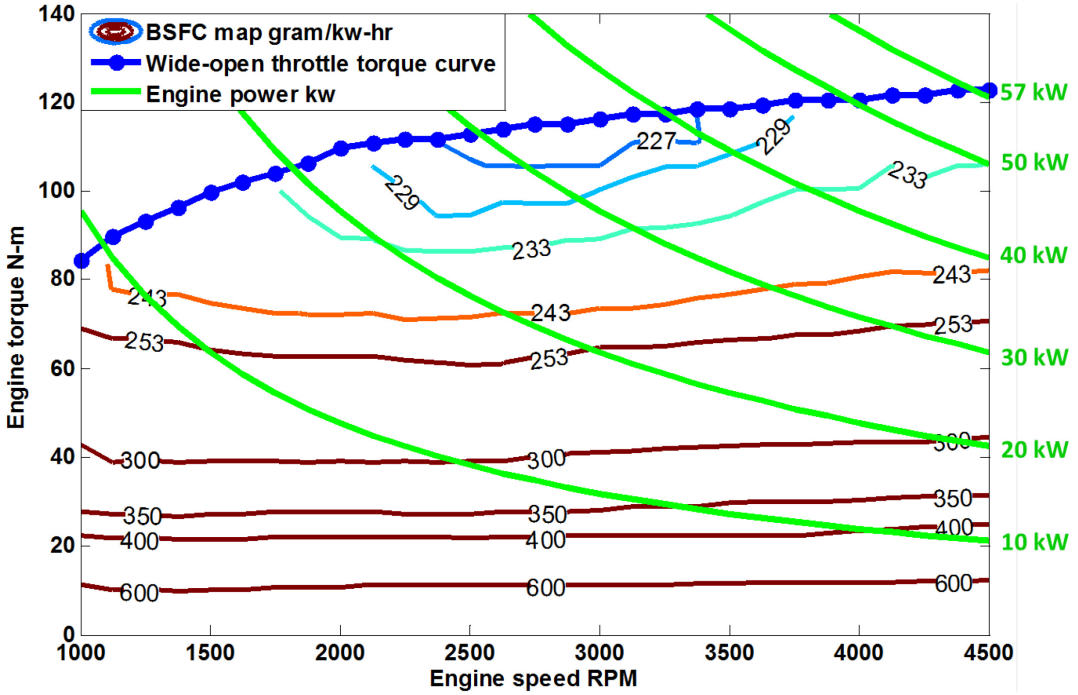

Figure 49. Engine BSFC map and the WOT torque curve.

The region below the WOT torque curve are the possible operating region of the engine. Base on BSFC map, the engine yields best energy conversion efficiency when it operates in the region near WOT curve and from 2200 RPM to 3200 RPM. The green lines are

engine equal-power curves. From the cross point of the WOT torque curve and a specific engine equal-power curve, all the points on the same equal-power curve are possible operating points for the engine at a given power. All the possible operating points and their corresponding fuel consumption could be found from the BSFC map and these data points are presented in Figure 50 in another way.

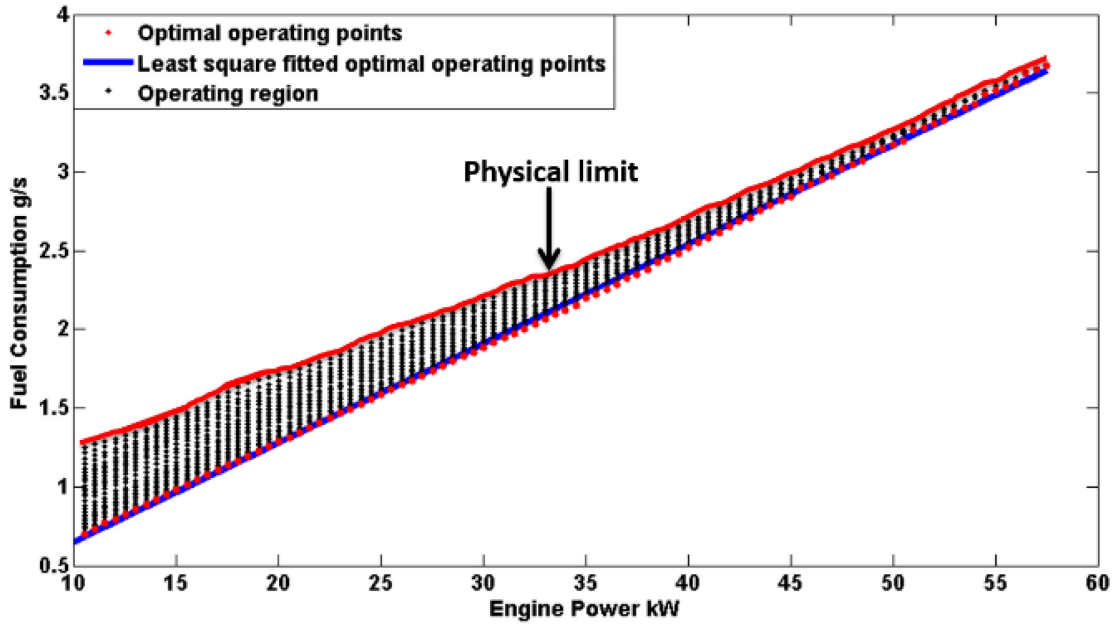

Figure 50. Engine power output vs. optimal fuel consumption at given power.

Each column of black dots are the possible operating points at a given power request. Lower the given power request is, more operating points available for this given power. This relation could be referred back to Figure 49, where smaller the value of an engine equal-power curve is, longer the curve is intercepted by the WOT torque curve. Thus more operating points could be found for a lower power request. That also explains why there are more operating points available for 10kw power request than that of 57 kW.

When the engine power output is held constant, an operating point of least fuel consumption rate is chosen as the optimal operating point. The optimal operating points

for each engine power value are labeled as red dots in Figure 50. A linear relation could be observed from Figure 50 between an engine power output and the instantaneous fuel consumption rate of the optimal operating point. The blue line is the linearly fitted line for all the optimal operating points. Again, this fitted line could be represented in the form of equation **(30)** with the values of a_2 and b_2 given below.

$$a_2 = -1.89 * 10^{-2} \left(\frac{1\%}{kw * s}\right)$$
$$b_2 = -5.64 * 10^{-3} \left(\frac{1\%}{s}\right)$$

To validate equation **(30)**, simulated instantaneous fuel consumption in Autonomie are compared with calculated fuel consumption. The comparison results are given in Figure 51.

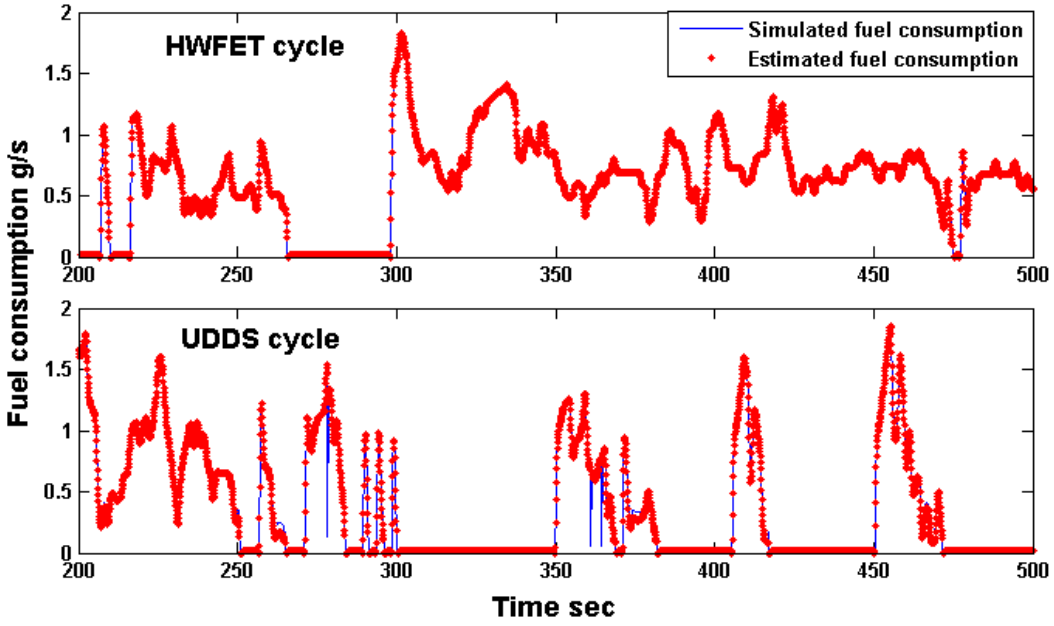

Figure 51. Simulated fuel consumption vs. estimated fuel consumption.

The figure is enlarged to show the simulation results between 200 sec to 500 sec and it shows that the estimated fuel consumption from equation **(30)** is generally accurate except for some instantaneous peaks.

5.3. **Vehicle powertrain model and its state-space representation**

In order to implement linear MPC and explicitly predict the battery SOC and instantaneous engine fuel consumption, a linearized vehicle model is given below.

$$\begin{cases} \dot{Q} = P_{eng} + P_{batt} + P_{mech\_brake} - P_{road} \\ S\dot{O}C = a_1 * P_{batt} + b_1 \\ \dot{m}_f = a_2 * P_{eng} + b_2 \end{cases} \quad \textbf{(31)}$$

$where\ Q = \frac{1}{2}mv^2$.

From the standpoint of energy conservation, the amount of kinetic energy of a vehicle is dynamic and its variation is the combinational effect of the power output of the entire powertrain and the external force which consumes the kinetic energy of the vehicle.

Q is the total kinetic energy of a vehicle and it is a function of vehicle speed. P_{eng} and P_{batt} are the actual power outputs of the engine and the battery to the vehicle powertrain. P_{mech_brake} is power output of the mechanical brake. P_{road} is the estimation of road resistance and its equation will be given later in (34. SOC is the state of charge of vehicle battery. Theoretically, the value of SOC varies from 0 to 100%. However, the battery should not be over charged or over discharged and its value is maintained around 70%. The last term $\dot{m}_f$ is instantaneous fuel consumption of the engine. a1, a2, b1 and b2 are the coefficients of the relation between engine power and instantaneous fuel consumption, and dSOC(t)/dt and battery power output P_{batt}.

Based on above dynamic equations, the discrete-time state-space representation of the linearized vehicle model is given below.

$$\begin{cases} \dot{x} = Ax + B_u u + B_v v \\ y = Cx \end{cases} \quad \textbf{(32)}$$

The states vector x is the same as output vector y,

$$x = y = \begin{pmatrix} Q \\ SOC \\ m_f \end{pmatrix}, and\ \dot{x} = \begin{pmatrix} \dot{Q} \\ \dot{SOC} \\ \dot{m}_f \end{pmatrix}$$

The controlled inputs are

$$u = \begin{pmatrix} P_{eng} \\ P_{batt} \\ P_{mech\_brake} \end{pmatrix}$$

The disturbances are

$$v = \begin{pmatrix} P_{road} \\ b_1 \\ b_2 \end{pmatrix}$$

The values of matrices, A, Bu, Bv, and C are given below.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B_u = \begin{bmatrix} 1 & 1 & 1 \\ 0 & a_1 & 0 \\ a_2 & 0 & 0 \end{bmatrix}$$

$$B_v = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The cost function of the MPC in this study is given below.

$$J(z_k) = \sum_{j=1}^{3} \sum_{i=1}^{N_p} \left\{ w_i^y [y_j(k+i+1|k) - r_j(k+i+1)] \right\}^2 + \sum_{j=1}^{3} \sum_{i=0}^{N_c-1} \left\{ w_i^{\Delta u} [\Delta u(k+i-1|k)] \right\}^2 + \rho_\epsilon \epsilon_k^2 \quad \textbf{(33)}$$

Subject to,

$$P_{eng}^{min} \le P_{eng}(k) \le P_{eng}^{max}; P_{batt}^{min} \le P_{batt}(k) \le P_{batt}^{max}$$

$$P_{mech\_brake}^{min} \le P_{mech\_brake}(k) \le P_{mech\_brake}^{max};\ Q^{min} \le Q \le Q^{max}$$

$$SOC^{min} \le SOC \le SOC^{max}; m_f^{min} \le m_f \le m_f^{max}$$

In which, the $z_k^T = [u(k|k)^T\ u(k+1|k)^T\ \ldots\ u(k+N_c-1|k)^T\ \epsilon_k]$ is the optimal controlled inputs calculated by quadratic programming.

The weighting factors are given by the following equations.

$w^y = [w_Q^y\ w_{SOC}^y\ w_{m_f}^y]$; $w^{\Delta u} = [w_{Peng}^{\Delta u}\ w_{Pbatt}^{\Delta u}\ w_{Pmech\_brake}^{\Delta u}]$

A parametric study will be performed to study the effect of the weighting factors to the performance of the HEV. The simulation results will be given and discussed in the last section.

P_{road} is the power of the road resistance including wind drag and friction force. It's given by the following equation.

$$P_{road} = \hat{v}[\frac{1}{2}\rho A C_d \hat{v}^2 + \mu m g(1 + \frac{\hat{v}}{44.3})] \qquad \textbf{(34)}$$

where ρ ($\frac{kg}{m^3}$) is the density of air. A (m^2) is the front projection area of the vehicle. C_d is the wind drag coefficient. μ is the friction coefficient of the tires. $\hat{v}$ ($\frac{m}{s}$) is the vehicle speed at last time step. P_{road} is estimated with vehicle speed from last time step to maintain the linearity of the equation set, considering the relatively slow dynamics of vehicle velocity comparing to that of the powertrain.

5.4. Integration of MPC controller with vehicle model in Autonomie

The model used in this section is the MY04 Prius power-split HEV model provided by Autonomie. The key parameters of the model are listed in Table 1.

Autonomie operates in Matlab environment. All the models and the initialization files of the model are saved in Matlab format. Users have the access to models of individual powertrain component. Each vehicle model in Autonomie is modulized and can be divided into four major parts, Driver Controller, Vehicle Powertrain Controller, Vehicle Powertrain

Architecture, and Environment as shown in Figure 1. The driver controller simulates the response of the driver by comparing the actual vehicle speed and the target speed. According to the output of the Driver Controller, the Vehicle Powertrain Controller generates component-specific commands, for example, the engine torque and speed command. The control strategy in this controller is rule-based and vehicle-dependent. The Vehicle Powertrain Architecture simulates the response of each powertrain component and the powertrain operating information will be fed back to the two controllers described above. The Environment block provides the environmental information, for example, the ambient temperature and road grade.

In order to test and validate MPC controller and its parameters developed in this study, the controller has been integrated into the Vehicle Powertrain Controller of the MY04 Prius model in Autonomie.

The study presented in this dissertation focuses on improving the VPC while leaving other components the same. The VPA is created and validated by vehicle bench testing or component-wise bench tests. Thus, the validity of VPA can be trusted. In order to take a further look into the details of the model, and how the MPC is integrated with Autonomie, further illustration and block diagrams are provided below.

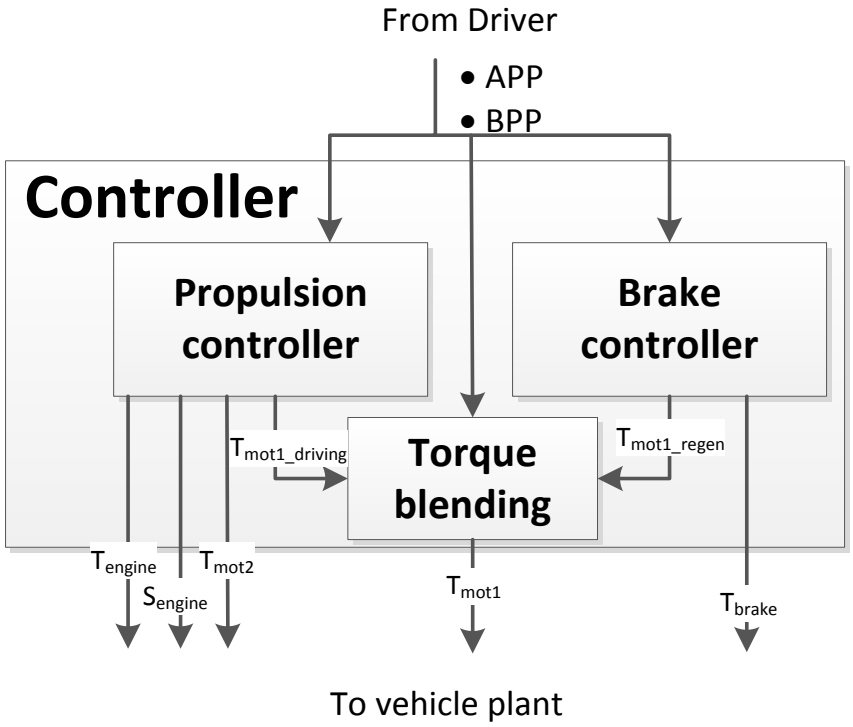

Figure 52. Original Autonomie controller

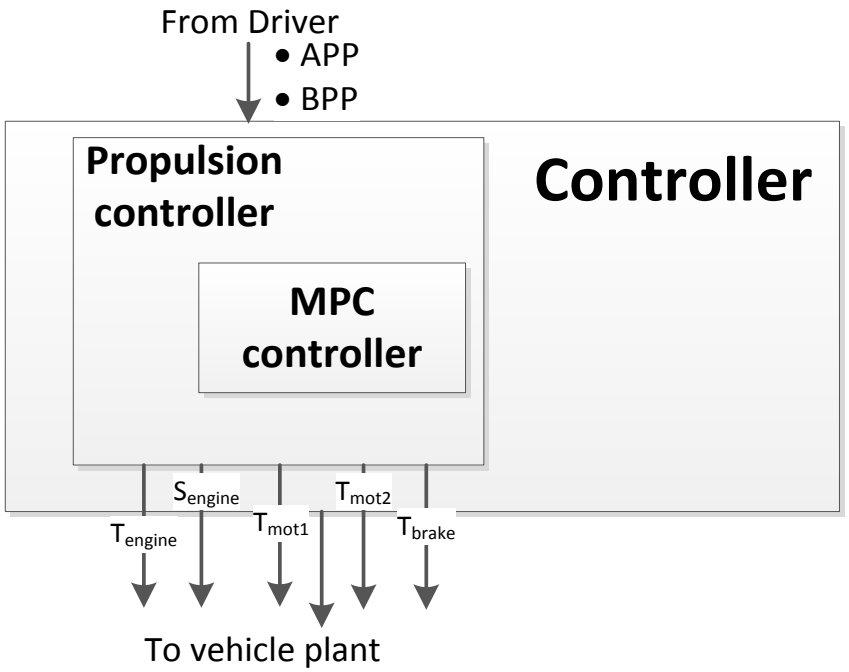

Figure 53. Autonomie controller integrated with MPC

Going one layer deeper into VPC, Figure 52 and Figure 53 shows the comparison between Autonomie original controller and the one integrated with MPC. In Figure 52, Autonomie VPC consists of three components, propulsion controller, brake controller, and the torque blending block. The propulsion controller calculates the torque request of motor 2, T_{mot2}, engine, T_{eng}, and driving torque of motor 1, $T_{mot1\_driving}$. It also makes decisions on when to turn on or off the engine, S_{eng}. The brake controller calculates torque of the mechanical brake T_{brake} and the regenerative braking torque of motor 1, T_{mot1_regen}. In the original rule-based Autonomie controller, the driving torque of motor 1, $T_{mot1\_driving}$, and the regenerative braking torque of motor 1, T_{mot1_regen} is blended together to find the combined torque output of the motor 1 in order to smoothen the torque output and avoid aggressive acceleration and deceleration. On the other hand in Figure 53, the MPC controller takes over the role of the brake controller of original Autonomie model, and calculates the torque request of motor 1 as a whole in order to be more consistent. Going one layer down into the propulsion controller, the following Figure 54 shows the overview of the Autonomie propulsion controller integrated with MPC.

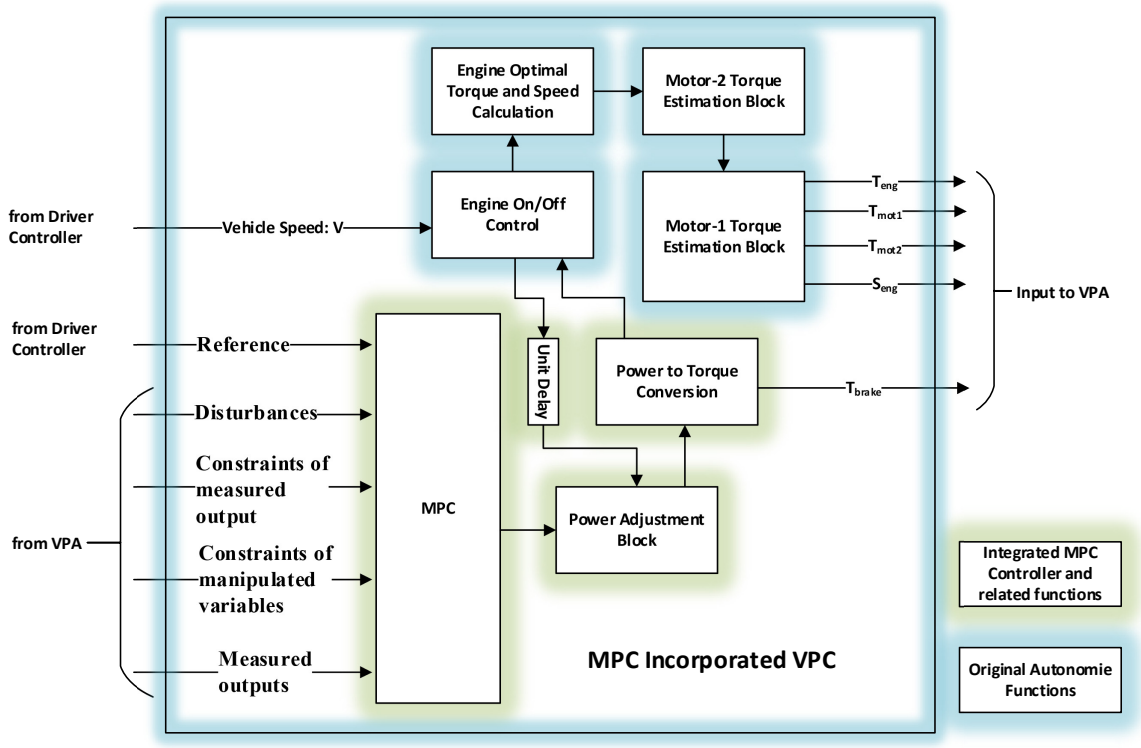

Figure 54. VPC propulsion controller with MPC incorporated

The driver controller's inputs will be considered as the reference of the MPC controller. The constraints, disturbances and measured outputs are feedback signals from the VPA. The manipulated variables will be fed back to MPC as the required information for the optimization in next time step. The MPC controller integrated into Autonomie vehicle model is shown in Figure 55.

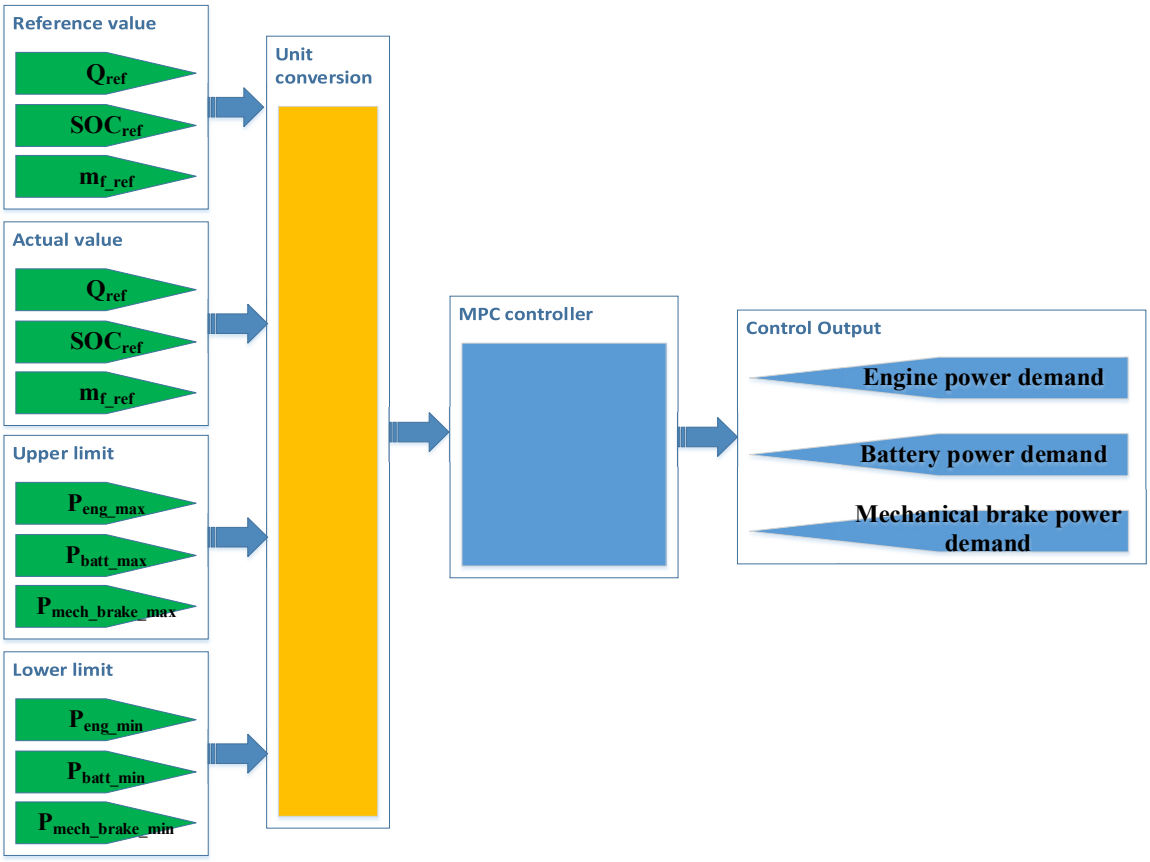

Figure 55. The MPC controller integrated into Autonomie

As shown in *Figure 55*, the input to the MPC controller are the references for vehicle kinetic energy Q, battery SOC, and fuel consumption m_f. The actual vehicle kinetic energy Q, battery SOC, and fuel consumption m_f are fed back to MPC controller from the Vehicle powertrain architecture. The constraints for P_{eng}, P_{batt}, and P_{brake} are inputs to the MPC controller as well. It is worth mentioning that the constraints for P_{batt} are updated each time instant based on the SOC. In order to maintain the differences of the magnitude of each measured output within a threshold, the unit of each input to MPC controller is carefully chosen, and unit conversion blocks are created to fulfill this purpose. The last part of the model are the controller outputs and they will be routed to the low-level controller

for torque value calculation. At last, the inputs and outputs of the MPC controller integrated into Autonomie are listed in *Table 6*.

Table 6. Inputs to the MPC controller

	Signal Name	Description	Signal type	Value	Unit
Reference	Q_{ref}	Kinetic energy target	Updated in real-time	$\frac{1}{2}mv^2$	Joule
	SOC_{ref}	SOC target	Constant	70	%
	m_{f_ref}	fuel consumption rate target	Constant	0	gram/sec
Output constraints	Q_{max}	Maximum kinetic energy allowed	Constant	2E+06	Joule
	Q_{min}	Minimum kinetic energy allowed	Constant	0	Joule
	SOC_{max}	Maximum SOC allowed	Constant	90	%
	SOC_{min}	Minimum SOC allowed	Constant	50	%
	m_{f_max}	Maximum mf allowed	Constant	10	gram/sec
	m_{f_min}	Minimum mf allowed	Constant	0	gram/sec
Input constraints	P_{eng_max}	Maximum engine power	Updated in real-time	-	watt
	P_{eng_min}	Minimum engine power	Updated in real-time	-	watt
	P_{batt_max}	Maximum battery power	Updated in real-time	-	watt
	P_{batt_min}	Minimum battery power	Updated in real-time	-	watt
	$P_{mech\_brake\_max}$	Maximum mechanical brake power	Constant	0	watt
	$P_{mech\_brake\_min}$	Minimum mechanical brake power	Updated in real-time	-	watt
Disturbance	P_{road}	Road resistance	Updated in real-time	(34	watt

	b_1	Battery SOC changing rate estimation coefficient	Constant	-0.33	1%/sec
	b_2	Engine fuel consumption estimation coefficient	Constant	0.05	gram/sec

5.5. Simulation results and analysis

5.5.1. Simulation results from UDDS cycle.

In this section, the simulation results of the MY04 Prius incorporated with model predictive control is analyzed. The following figure shows the vehicle speed and battery SOC vs. time.

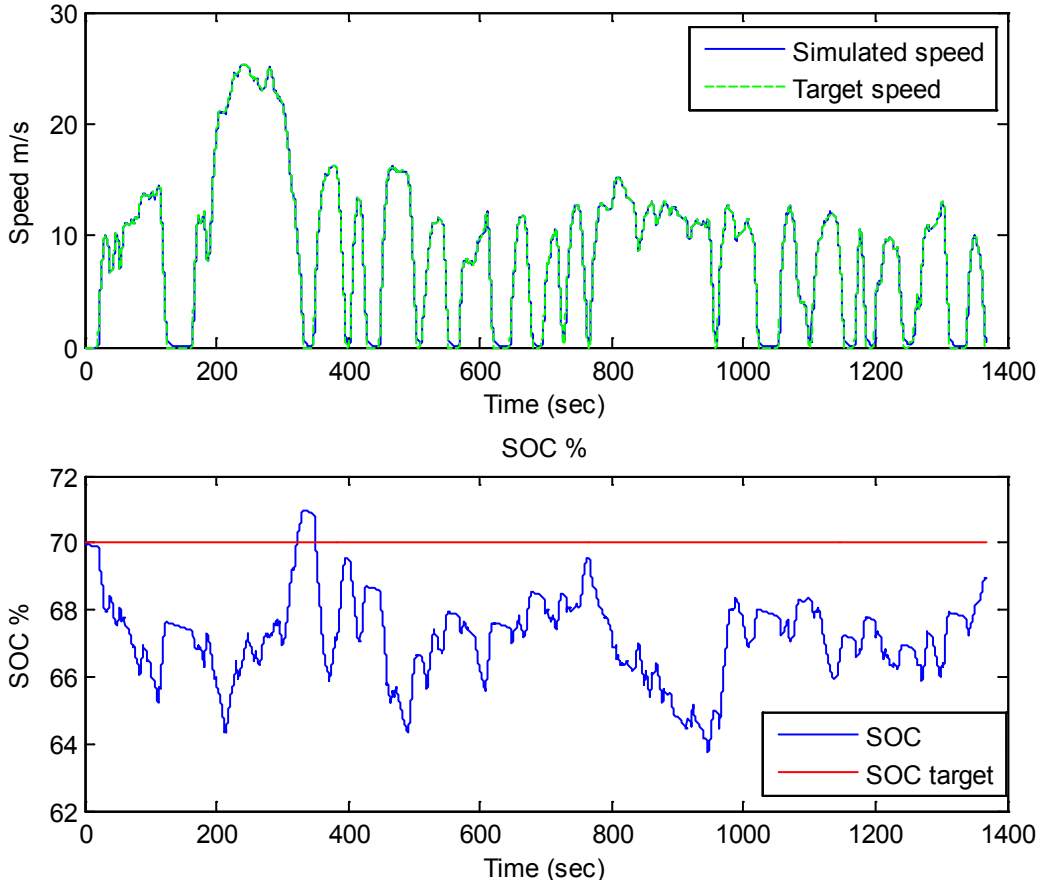

Figure 56. SOC vs. time and vehicle speed in UDDS.

Figure 56 shows that the MPC performs precisely in tracking the vehicle speed to the target vehicle speed. The second observation is that MPC has good control over the regenerative braking. Whenever there is decceleration, regenerative braking will take place and cause a rise in the SOC. The third observation is that the SOC balancing is achieved, since the SOC is maintained well around 70%, which is the mid points of its operating range.

The following figure shows the manipulated variable output from the MPC controller. They are power of the engine P_{eng}, power of the battery P_{batt}, and the power of the mechanical brake P_{mech_brake}.

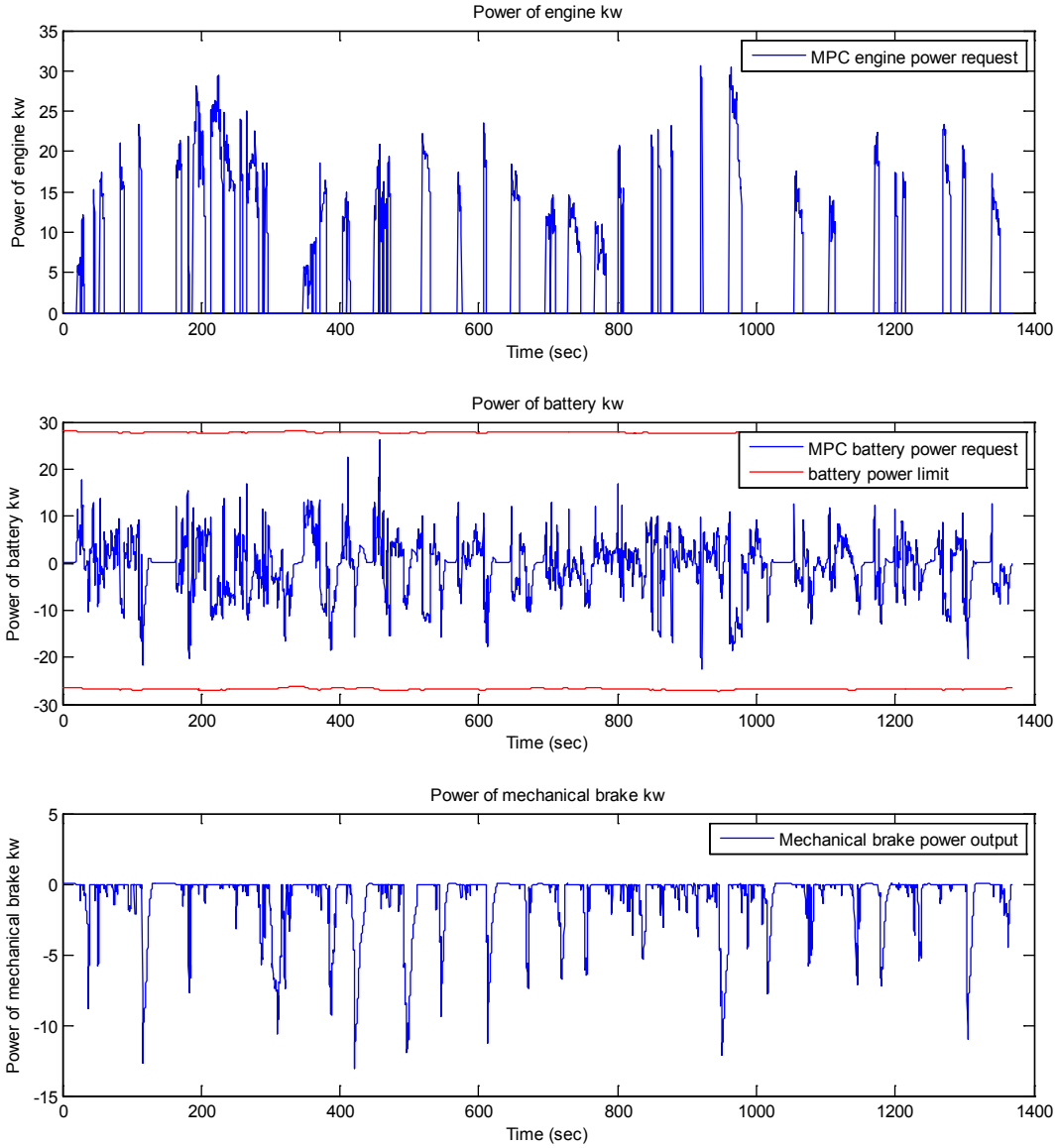

Figure 57. Manipulated variables of the MPC in UDDS cycle.

The top subplot in Figure 57 is the engine power request from MPC. Every peak in this subplot is an engine ON/OFF event. In the UDDS driving cycle, the enigne power output is maintained around 20 kw to 30kw, which is within the optimal region of this engine according to the BSFC map of this engine. The middel subplot shows the battery power request from the MPC as well as the upper and lower limits of the battery. The upper and lower limits of the battery are determined by battery state of charge. As shown in the this

subplot, the MPC controller is able to maintain battery power request within the battery limit. The constraints of the battery are updated at each simulation time-step, and MPC is able to take the real-time constraints into consideration. The bottom subplot in Figure 57 is the mechanical brake power request. Apparently, the power request for the mechanical brake is too often, which might lead to waste of vehicle kinetic energy during braking events. In order to address this issue, the power request signals from MPC will be processed later in the power adjustment block in Figure 54 to adjust the mechanical brake power P_{brake} to an appropriate level. Thus, regenerative braking will be maximized. The integration of MPC controller with low level control logic is helpful to adjust the manipulated variables. For example, it is reasonable and necessary to maximize regnerative braking torque during braking events. Only when regenerative braking torque is satuarated, the excess amount of braking torque will be assigned to mechanical brake. However, expressing this in a cost function means the weighting factor of regenerative braking shall be infinite or the weighting factor for mechanical brake shall be zero. Either of these two methods will lead to illed solution from QP solver.

After MPC calculatets the power of the engine $P_{eng,}$ the power of the battery P_{batt}, and the power of the mechancal brake P_{brake}. A low-level controller will calculate the torque request for the engine, the motor, and the motor 2, which are given in Figure 58.

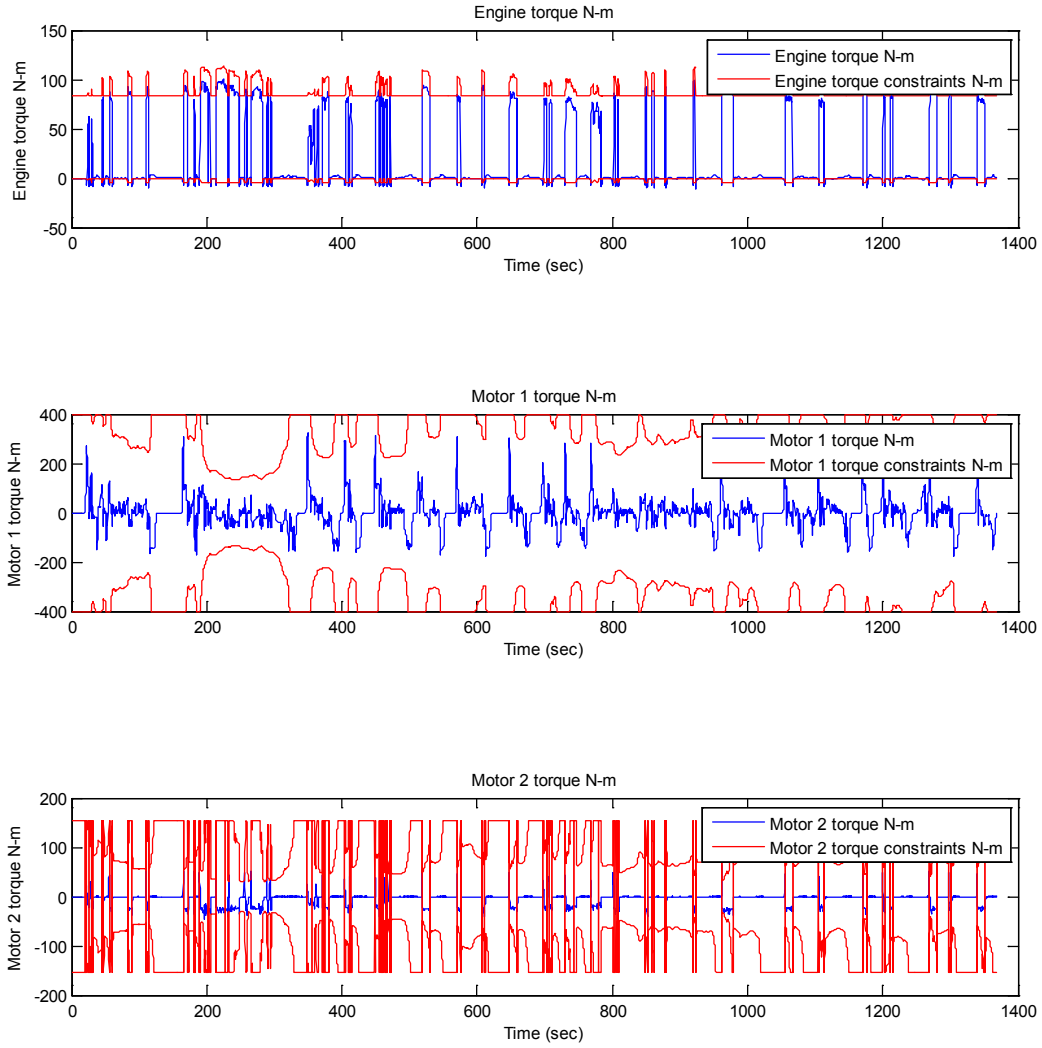

Figure 58. Torque request of the engine, motor 1 and motor 2 with their constraints.

It can be observed from the simulation results that the MPC controller is able to keep the torque values within the constraints. The torque constraints of the engine is related to the engine speed. The torque constraints of the motor 1 is influenced by the vehicle speed. The available torque of motor 1 and motor 2 depend on the battery SOC.

5.5.2. Parametric study on the weighting factors for MPC.

A parametric study was performed to compare the effect of three parameters, w_Q^y, w_{SOC}^y, and $w_{m_f}^y$. They are weighting factors that punish the difference between the reference

values and the corresponding controlled outputs including vehicle kinetic energy Q, battery SOC, and accumulated engine fuel consumption m_f. The relation between the three weighting factors and the MPC cost function is discussed in "Vehicle powertrain model and its state-space representation" section. The matrix of the parametric study is given in the following table.

Table 7. Parametric study of the weighting factors.

Parameter Sets	w_Q^y	w_{SOC}^y	$w_{m_f}^y$
A	1	1	1
B	1	0.1	1
C	0.1	1	1
D	1	1	0.1
E	0.1	0.1	1
F	0.1	1	0.1
G	1	0.1	0.1

Generally speaking, greater weighting factor means faster response speed of its corresponding controlled output. The reference of kinetic energy Q is calculated from target vehicle speed at the moment. The SOC reference is at 70%, the median point of battery operating range. The target accumulated fuel consumption is set to 0 for the purpose of minimizing fuel consumption. An important constraint is the lower limit of battery SOC and it is set to 50% which means the MPC controllers will try to follow the 70% SOC reference while keep SOC above 50%. The following two tables showed the simulation results of each parameter set in UDDS and HWFET cycle.

Table 8. Simulation Results for UDDS cycle

Parameter Sets	Speed Error (%)	Delta SOC (%)	Fuel Consumption (gram)
A	4.53%	-6.50%	283.34
B	3.90%	-16.44%	295.65
C	10.07%	-6.92%	285.49
D	4.00%	0.47%	300.14
E	8.28%	-17.65%	321.69
F	7.60%	0.19%	299.31
G	4.09%	-5.25%	282.01

Table 9. Simulation Results for HWFET cycle

Parameter Sets	Speed Error (%)	Terminal SOC (%)	Fuel Consumption (gram)
A	0.56%	-11.16%	436.04
B	0.56%	-14.95%	446.89
C	1.48%	-12.06%	436.37
D	0.43%	7.04%	495.69
E	6.71%	-14.52%	441.68
F	1.23%	6.25%	495.29
G	0.43%	3.21%	491.17

Speed error (%) is defined as the percentage of time when the simulated vehicle speed is more than 2 miles per hour (mph) above or below the target speed. This parameter reflects

a controller's ability of speed tracking and it is controlled by weighting factor w_Q^y. Terminal SOC (%) is simulated battery SOC at the end of a driving cycle and it is dependent on the driving cycle as well as the weighting factor w_{SOC}^y. Fuel consumption (gram) is the accumulated fuel consumption at the end of a driving cycle and its value is influenced by weighing factor $w_{m_f}^y$.

The values in the speed error column showed a good correlation with variation of weighting factor w_Q^y. In parameter sets C, E and F in which w_Q^y is 0.1, the speed error is greater than 5% in UDDS and 1 % in HWFET. It is because MPC controller is putting more effort in charge balancing and fuel consumption reduction when w_Q^y is relatively small, thus less effort is made for tracking vehicle speed. Another observation is that certain combination of parameter set could yield better performance than other parameter sets. For example, in UDDS driver cycle, parameter set G is better than parameter sets E in both fuel economy and terminal SOC. In conclusion, speed error is sensitive to w_Q^y and greater fuel consumption could be found in parameter sets D, F, and G because $w_{m_f}^y$ equals to 0.1 in these 3 parameter sets. The relation between fuel consumption, terminal SOC and the corresponding weighting factors, $w_{SOC}^y, and\ w_{m_f}^y$ are further studied below. Figure 59 shows the SOC trends in UDDS simulation.

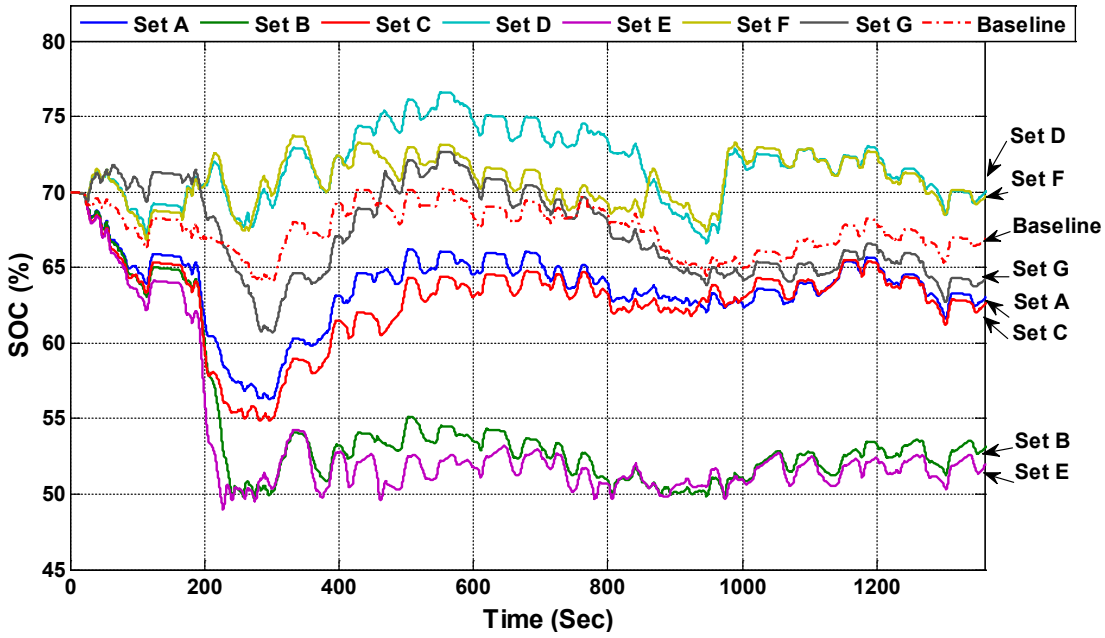

Figure 59. Battery SOC comparison in UDDS Driving Cycle

The SOC figure showed trends associated with different combination of weighting factors. The baseline trend is the simulation result of the PID controller in original Prius MY04 model. The rest trends are the SOC curves corresponding to each of the 7 parameter sets. With w_{SOC}^{y} set to 0.1 and $w_{m_f}^{y}$ set to 1, parameter sets B and E operate the battery near 50% SOC, which is its lower limit, with little charge-balancing observed. The controller tends to use electric energy with priority, while maintains SOC above its lower limit. Higher fuel consumption is observed especially parameter set E. With w_{SOC}^{y} and $w_{m_f}^{y}$ both set to 1, parameter sets A and C is able to maintain SOC slightly below 70% as well as to perform charging balancing. The controller is balancing between using electrical motor and engine. Parameter sets D and F has a w_{SOC}^{y} that is 10 times larger than their $w_{m_f}^{y}$, thus the SOC of these two parameters sets are maintained close to 70% SOC reference at the price of greater fuel consumption. The following figure is given to support the explanation above.

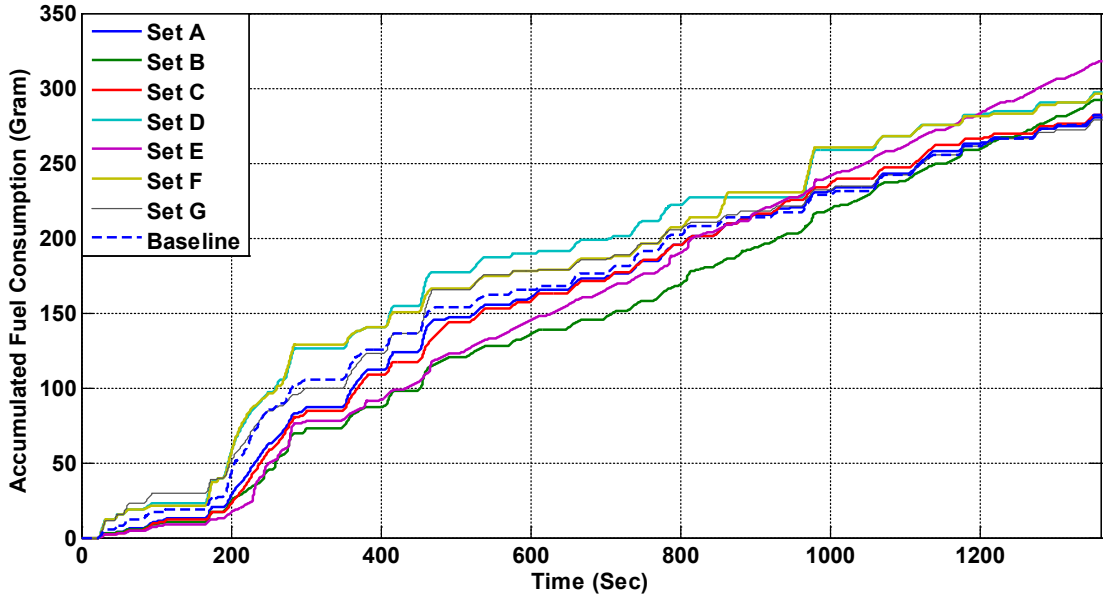

Figure 60. Engine Fuel Consumption Comparison in UDDS Driving Cycle

Figure 60 shows the accumulated fuel consumption of each of the 7 parameter sets as well as the PID controller in the original Prius MY04 model in Autonomie. The trend of accumulated fuel consumption could be explained by the combination of w_{SOC}^{y} and $w_{m_f}^{y}$. The fuel consumption trends of parameter sets E and B increases linearly with time. After the SOC drops to 50% at around 200 seconds, the engine is steadily consuming extra fuel to keep the battery SOC above the lower limit line, thus engine fuel consumption trend becomes irrelevant to the driving cycle. Parameter set D and set F also consume relatively more fuel because it takes more energy to maintain SOC at a higher level than the rest of the parameter sets. UDDS baseline, set A, set C, and set G consumes almost same amount of fuel at the end of the driving cycle.

Figure 61 and Figure 62 show the simulation results of HWFET cycle and the figures for SOC and fuel consumption are given below.

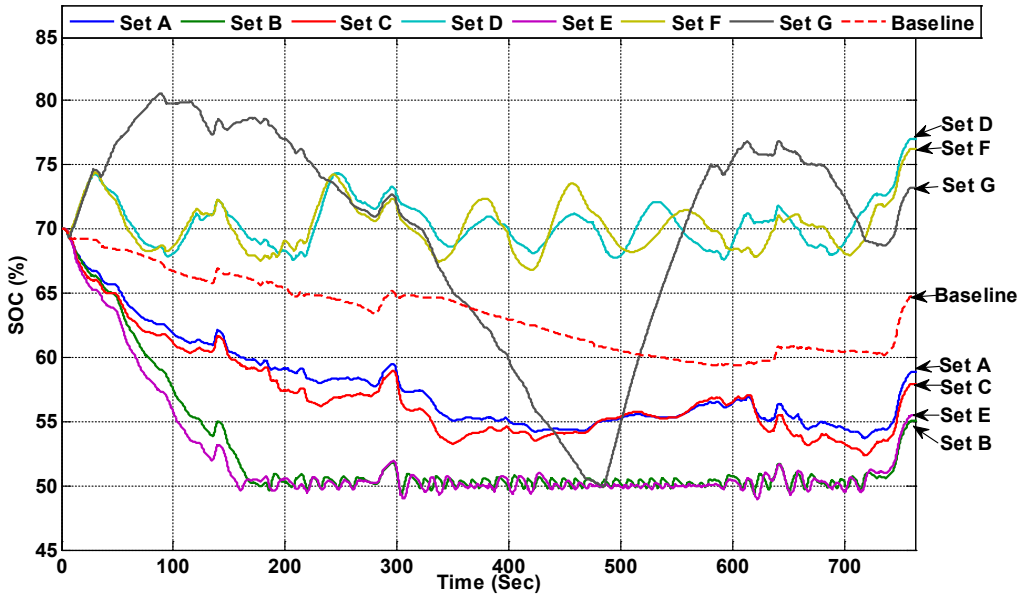

Figure 61. Battery SOC Comparison in HWFET Driving Cycle

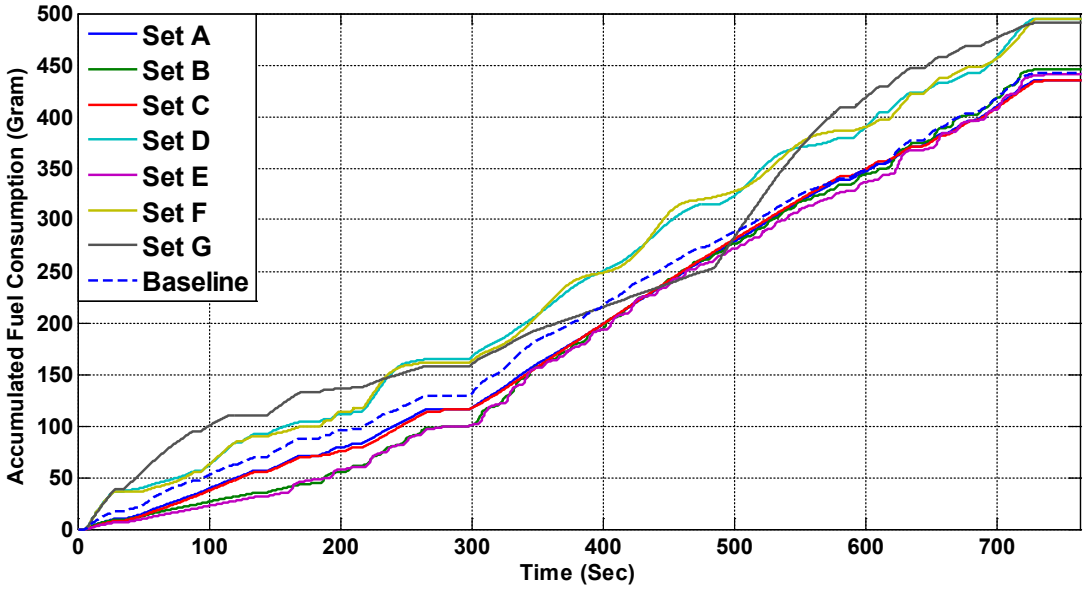

Figure 62. Engine Fuel Consumption comparison in HWFET Driving Cycle

Figure 61 shows similar trends as observed in Figure 59. The SOC trends are generally divided into 4 groups based on parameter sets. Group 1 is set D and F which are able to maintain SOC near 70% reference line. Group 2 is set A and set C which could maintain SOC slightly below 70% reference line. Group 3 is set B and Set E which fail to maintain

SOC above its lower limit and it results in high fuel consumption and low terminal SOC. The irregular behavior of set G is because the QP solver encountered severe numerical problem. In Figure 62, it could be inferred that Group 1 resulted in highest fuel consumption to maintain the SOC around 70%, while the irregular fuel consumption of set G is caused by the same issue discussed in Figure 61.

In summary, the performance of a HEV integrated with MPC controller is highly dependent on the weighting factors of the controlled outputs. There are two conclusion from the analysis in this section. One is that the speed error is solely dependent on the value of w_Q^y. The other conclusion is that the terminal SOC is determined by the combination of w_{SOC}^y and $w_{m_f}^y$.

.

Chapter 6 Conclusion and Future Work.

In conclusion, the dissertation is motivated by the need of improving the fuel economy of hybrid electric vehicles. The contribution of this dissertation is given as following.

First of all, driving cycle recognition is realized by pattern recognition method. By taking this approach, a driving cycle is represented by feature vectors. These feature vectors are formed by a number of features and corresponding weighting factors. In order to classify a real-world driving cycle to one of the driving cycles in a reference database, four federal driving cycles: UDDS, HWFET, US06, and SC03 are used to form this reference driving cycle database. Fifteen feature parameters to which four federal driving cycles are sensitive and corresponding weighting factors are identified to form the representative feature vectors for the selected four driving cycles. The performance of the presented driving cycle pattern recognition method and the impact of the dissimilarity measures and the feature extraction methods on the success rate of the driving cycle pattern recognition and the quality of representative feature vectors are investigated using two real world driving cycles: real-world highway cycle and real-world city cycle. The evaluation result shows that the size of sample window, the type of dissimilarity measures, and the feature extraction method have a great impact on the performance of driving cycle pattern recognition. The presented pattern recognition algorithm is integrated with the Prius MY04 vehicle model in Autonomie. The effectiveness of the adaptive control is studied by comparing the fuel economy of adaptive control with the fuel economy of fixed control parameters when a combination of different driving cycles is inputted into the model. The simulation results show that the adaptive control can improve the fuel economy up to

2.63%. In the future study, additional factors such as driver behavior, traffic condition, and the weather condition, could be taken into account in the driving pattern recognition. In addition, dynamic selection of control strategies for different driving patterns also is recommended to be investigated.

As the second part of the dissertation, the simulation result presented in this chapter shows that the aggressive driving exhibit great impact on the overall efficiency of the HEV as well as the fuel economy. In particular, the engine energy loss is the major contributor to the increase of fuel consumption. Through detailed analysis, it is found that one of the most important factors that cause the increase of engine energy loss is due to the frequent turn on/off engine under aggressive driving. To reduce the impact of aggressive driving on the HEV fuel consumption, vehicle controllers should limit the engine power change rate. Another valuable finding from this study is that aggressive driving will cause large instantaneous braking power peaks. Due to physical constraints of regenerative braking, the amount of energy recovered by regenerative braking does not change too much. However, the mechanical braking energy loss is increased a lot.

To design vehicle control strategies that could reduce the impact of aggressive driving, further simulation study could be conducted with other two types of HEV configurations (parallel and series), various driving cycles such as UDDS, US06, HWFET and SC03, and different driving cycle scale factors. For each combination of vehicle configuration, driving cycle, and scale factor, the powertrain energy flows, the energy consumption of individual powertrain components, their operating regions, and the energy losses of these components could be investigated to identify the major factors that needed to be considered in the fuel efficient powertrain control.

The simulation results obtained from a power-split vehicle is valuable to other two types of vehicle configurations. However, different powertrain configuration introduces new problems. For example, in parallel vehicle, the engine is directly connected to the powertrain shaft which means that the engine speed is not independent to the vehicle speed. Thus, the control strategy dealing with the aggressive driving will be more challenging because it is harder to keep the engine within an optimal operating range.

In the last part of the study, a linearized vehicle model has been developed in order to apply linear MPC for powertrain control. Linearization has been performed to the relations between engine power output and its engine fuel consumption, and the changing rate of battery SOC and battery power output. An MPC controller is developed and integrated with the MY04 Prius model in Autonomie.

A parametric study has been conducted to understand the influence of 3 weighting factors, w_Q^y, w_{SOC}^y, and $w_{m_f}^y$, in MPC cost function on the performance of the vehicles. These three weighting factors determine the response speed of vehicle kinetic energy, SOC, and fuel consumption to the reference. There are two conclusions. One is that the speed error is solely dependent on the value of w_Q^y. The other conclusion is that the terminal SOC is determined by the combination of w_{SOC}^y and $w_{m_f}^y$. Even though MPC has not been proven to exhibit higher efficiency than original PID controller at the current stage of research, it is reasonable to expect better results with parametric study of higher resolution. This study is also a good starting point on implementing advanced predictive control algorithm in power-split HEVs. As the following research of this study, Markov chain will be integrated with the current HEV model to provide a prediction of driver's behavior based on past vehicles states information.

APPENDICES

Appendix A. The Simulation Results and the Calculation of Equivalent Fuel Economy of HEV

A.1 The Simulation Results

These following tables contain the simulation results for power-split vehicles in 3 driving cycles.

Table 10. The Simulation Results for Power-Split Vehicle in HWFET

Driving Cycle	HWFET					
Scale Factor λ	1	1.1	1.2	1.3	1.4	1.5
Fuel Consumption(Gallon)	0.03	0.06	0.13	0.20	0.25	0.29
Electrical Consumption(kW.h)	2125.4	2040.3	1476.5	1045.2	760.6	769.8
MPGe (Gallon/Mile)	110.89	83.49	58.41	45.23	37.42	33.14

Table 11. The Simulation Results for Power-Split Vehicle in US06

Driving Cycle	US06					
Scale Factor λ	1	1.1	1.2	1.3	1.4	1.5
Fuel Consumption(Gallon)	0.15	0.20	0.23	0.25	0.28	0.30
Electrical Consumption(kW.h)	1088.7	1039.0	1310.4	1739.9	1698.8	1775.7
MPGe (Gallon/Mile)	43.26	34.32	29.37	25.86	23.18	21.43

Table 12. The Simulation Results for Power-Split Vehicle in UDDS

Driving Cycle	UDDS					
Scale Factor λ	1	1.1	1.2	1.3	1.4	1.5
Fuel Consumption(Gallon)	0.02	0.03	0.07	0.11	0.15	0.19
Electrical Consumption(kW.h)	1393.8	1990.2	1777.0	1587.3	1569.2	1558.8
MPGe (Gallon/Mile)	130.37	84.30	60.96	46.25	38.16	31.82

A.2 The Calculation of Equivalent Fuel Consumption

The following equation is used to calculate the equivalent fuel economy,

$$\text{Fuel Economy}_e = \frac{\text{Distance}}{\text{FC} + \frac{E_{EC}}{E_{gas/gal}}}$$

The FC represents the fuel consumption of the whole driving cycle.

EEC is the electrical consumption of the whole driving cycle.

Egas/gal is the conversion factor of equivalent fuel consumption.

Egas/gal=33440 W.h/Gal

Appendix B. SAE Copyright documentation for Chapter 3 and Chapter 5.

Permission is hereby granted and subject to the following terms and conditions:

--Permission is granted for non-exclusive world English language rights, for this one-time single use.

--Permission is required for new requests, subsequent editions, or further use of the material.

--The SAE material must be clearly identified and include the following statement "Reprinted with permission from SAE International."

--We also request that you include a complete reference to the SAE document in the reference section of your report.

--This permission does not cover any third party copyrighted work which may appear in the material requested.

--Licensor's use of this material, in whole or in part, is entirely its responsibility, and SAE International does not warrant or is not responsible for any use of the material.

Kind regards,

_____

Monica Nogueira
Content Acquisition Manager
Multimedia

SAE INTERNATIONAL
400 Commonwealth Drive
Warrendale, PA 15096

Office: +1.724.772.7525
Mobile: +1.412.996.5809
E-mail: Nogueira@sae.org
www.sae.org

Appendix C. ASME Copyright documentation for Chapter 4.

It is our pleasure to grant you permission to use all or any part of the following ASME paper as print only "Study the Impact of Driver's Behavior on the Energy Efficiency of Hybrid Electric Vehicles," by Lei Feng and Bo Chen, Paper No. DETC2013-12785, cited in your letter for inclusion in the PhD dissertation entitled INCORPORATING DRIVER'S BEHAVIOR INTO PREDICTIVE POWERTRAIN ENERGY MANAGEMENT FOR A POWER-SPLIT HYBRID ELECTRIC VEHICLE to be published by Michigan Technological University.

Permission is granted for the specific use as stated herein and does not permit further use of the materials without proper authorization. Proper attribution must be made to the author(s) of the materials. Please note: if any or all of the figures and/or Tables are of another source, permission should be granted from that outside source or include the reference of the original source. ASME does not grant permission for outside source material that may be referenced in theASME works.

As is customary, we request that you ensure full acknowledgment of this material, the author(s), source and ASME as original publisher. Acknowledgment must be retained on all pages printed and distributed.

Many thanks for your interest in ASME publications.

Sincerely,

Beth Darchi
Publishing Administrator
ASME
2 Park Avenue, 6th Floor
New York, NY 10016-5990
Tel 1.212.591.7700
darchib@asme.org

REFERENCES

1. EPA/NHTSA, *Light-Duty Vehicle Greenhouse Gas Emission Standards and Corporate Average Fuel Economy Standards; Final Rule.*, F. Register, Editor. 2010.
2. EPA/NHTSA, *2017 and Later Model Year Light-Duty Vehicle Greenhouse Gas Emissions and Corporate Average Fuel Economy Standards; Final Rule.*, F. Register, Editor. 2012.
3. Moura, S.J., et al., *A Stochastic Optimal Control Approach for Power Management in Plug-In Hybrid Electric Vehicles.* Control Systems Technology, IEEE Transactions on, 2011. **19**(3): p. 545-555.
4. Liu, L. and F.G. Yuan, *Wireless sensors with dual-controller architecture for active diagnosis in structural health monitoring.* Smart Materials & Structures, 2008. **17**(2).
5. Borhan, H.A., et al. *Predictive energy management of a power-split hybrid electric vehicle*. in *American Control Conference, 2009. ACC '09.* 2009.
6. Serrao, L., S. Onori, and G. Rizzoni, *A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles.* Journal of Dynamic Systems, Measurement, and Control, 2011. **133**(3): p. 031012.
7. Sciarretta, A. and L. Guzzella, *Control of hybrid electric vehicles.* Control Systems, IEEE, 2007. **27**(2): p. 60-70.
8. Ericsson, E., *Independent Driving Pattern Factors and their Influence on Fuel-use and Exhaust Emission Factors.* Transportation Research Part D-Transport and Environment, 2001. **6**(5): p. 325-345.
9. Ericsson, E., *Variability in Urban Driving Patterns.* Transportation Research Part D-Transport and Environment, 2000. **5**(5): p. 337-354.
10. Jeon, S.-i., et al., *Multi-Mode Driving Control of a Parallel Hybrid Electric Vehicle Using Driving Pattern Recognition.* Journal of Dynamic Systems, Measurement, and Control, 2002. **124**(1): p. 141-149.
11. Lin, C.-C., et al., *Driving Pattern Recognition for Control of Hybrid Electric Trucks.* Vehicle System Dynamics, 2004. **42**(1-2): p. 41-58.
12. Lipeng, Z., L. Cheng, and N. Xiang. *Optimization of Control Strategy for Plug-in Hybrid Electric Vehicle Based on Differential Evolution Algorithm*. in *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific*. 2009.
13. Langari, R. and J.S. Won, *Intelligent energy management agent for a parallel hybrid vehicle - Part 1: System architecture and design of the driving situation identification process.* Ieee Transactions on Vehicular Technology, 2005. **54**(3): p. 925-934.
14. Berry, I.M., *The Effects of Driving Style and Vehicle Performance on the Real-World Fuel Consumption of U.S. Light-Duty Vehicles*, in *Department of Mechanical Engineering*. 2007, Massachusetts Institute of Technology: Boston.
15. Gonder, J., et al., *Using GPS Travel Data to Assess the Real World Driving Energy Use of Plug-In Hybrid Electric Vehicles (PHEVs)* in *the Transportation Research Board (TRB) 86th Annual Meeting* 2007: Washington, D.C. .

16. Saboohi, Y. and H. Farzaneh, *Model for developing an eco-driving strategy of a passenger vehicle based on the least fuel consumption.* Applied Energy, 2009. **86**(10): p. 1925–1932.
17. Sharer, P., R. Leydier, and A. Rousseau, *Impact of Drive Cycle Aggressiveness and Speed on HEVs Fuel Consumption Sensitivity*, in *2007 SAE International World Congress*. 2007: Detroit, Michigan, USA.
18. Carlson, R.B., et al., *Drive Cycle Fuel Consumption Variability of Plug-In Hybrid Electric Vehicles Due to Aggressive Driving*, in *SAE World Congress & Exhibition*. 2009: Detroit, MI, USA.
19. Lee, T. and J. Son, *Relationships between Driving Style and Fuel Consumption in Highway Driving*. 2011, The Automotie Research Association of India, Pune, India.
20. Sundstroem, O., D. Ambuehl, and L. Guzzella, *On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints.* Oil & Gas Science and Technology-Revue D Ifp Energies Nouvelles, 2010. **65**(1): p. 91-102.
21. Lin, C.-C., et al., *Power management strategy for a parallel hybrid electric truck.* Control Systems Technology, IEEE Transactions on, 2003. **11**(6): p. 839-849.
22. Lars, J., A. Mattias, and E. Bo, *Assessing the Potential of Predictive Control for Hybrid Vehicle Powertrains Using Stochastic Dynamic Programming.* Intelligent Transportation Systems, IEEE Transactions on, 2007. **8**(1): p. 71-83.
23. Koot, M., et al., *Energy management strategies for vehicular electric power systems.* IEEE Transactions on Vehicular Technology. **54**(3).
24. Perez, L.V., et al., *Optimization of power management in an hybrid electric vehicle using dynamic programming.* Mathematics and Computers in Simulation, 2006. **2006**(73): p. 244-254.
25. Chan-Chiao, L., P. Huei, and J.W. Grizzle. *A stochastic control strategy for hybrid electric vehicles*. in *American Control Conference, 2004. Proceedings of the 2004*. 2004.
26. Tate, E.D., J.W. Grizzle, and H. Peng, *Shortest path stochastic control for hybrid electric vehicles.* International Journal of Robust and Nonlinear Control, 2008. **18**(14): p. 1409-1429.
27. Borhan, H., et al., *MPC-Based Energy Management of a Power-Split Hybrid Electric Vehicle.* Control Systems Technology, IEEE Transactions on, 2012. **20**(3): p. 593-603.
28. Bichi, M., et al. *Stochastic model predictive control with driver behavior learning for improved powertrain control*. in *Decision and Control (CDC), 2010 49th IEEE Conference on*. 2010.
29. Ripaccioli, G., et al. *A stochastic model predictive control approach for series hybrid electric vehicle power management*. in *American Control Conference (ACC), 2010*. 2010.
30. Kim, N., S. Cha, and H. Peng, *Optimal Control of Hybrid Electric Vehicles Based on Pontryagin's Minimum Principle.* Control Systems Technology, IEEE Transactions on, 2011. **PP**(99): p. 1-9.

31. Muller, E.A., A.G. Stefanopoulou, and L. Guzzella, *Optimal Power Control of Hybrid Fuel Cell Systems for an Accelerated System Warm-up.* IEEE Transactions on Control Systems Technology, 2007. **15**(2).
32. Ambühl, D., et al., *Explicit optimal control policy and its practical application for hybrid electric powertrains.* Control Engineering Practice, 2010. **18**(12): p. 1429-1439.
33. Sciarretta, A., M. Back, and L. Guzzella, *Optimal control of parallel hybrid electric vehicles.* Control Systems Technology, IEEE Transactions on, 2004. **12**(3): p. 352-363.
34. Musardo, C., G. Rizzoni, and B. Staccia. *A-ECMS: An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management*. in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*. 2005.
35. Paganelli, G., et al., *General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles.* JSAE Review, 2001. **22**(4): p. 511-518.
36. Karbowski, D., et al., *Instantaneously Optimized Controller for a Multimode Hybrid Electric Vehicle.* 2010.
37. Musardo, C., et al., *Supervisory Control for NOx Reduction of an HEV with a Mixed-Mode HCCI/CIDI Engine*, in *2005 American Control Conference*. Portland, OR, USA.
38. Pisu, P. and G. Rizzoni, *A Supervisory Control Strategy for Series Hybrid Electric Vehicles with Two Energy Storage Systems*, in *Vehicle Power and Propulsion, 2005 IEEE Conference*.
39. Park, J., Y. Park, and J.-H. Park, *Model-Based Optimization of Supervisory Control Parameters for Hybrid Electric Vehicles.* SAE Technical Papers, 2008. **2008-01-1453**.
40. Gonder, J.D., *Route-Based Control of Hybrid Electric Vehicles.* 2008.
41. Kim, N. and A.P. Rousseau, *Comparison Between Rule-Based and Instantaneous Optimization for a Single'Mode, Power-Split HEV.* SAE Technical Papers, 2011. **2011-01-0873**.
42. Banvait, H., S. Anwar, and Y. Chen, *A Rule-Based Energy Management Stategy for Plug-in Hybrid Electric Vehicle (PHEV)*, in *2009 American Control Conference*. Hyatt Regency Riverfront, St. Louis, MO, USA.
43. Gong, Q., Y. Li, and Z. Peng, *Trip-Based Optimal Power Management of Plug-in Hybrid Electric Vehicles.* IEEE Transactions on Vehicular Technology, 2008. **57**(6).
44. Zhu, Y., et al. *A Four-Step Method to Design an Energy Management Strategy for Hybrid Vehicles*. in *Proceeding of the 2004 American Control Conference*. 20044. Boston. Massachusetts
45. Pisu, P. and G. Rizzoni. *H_∞ control for hybrid electric vehicles*. in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. 2004.
46. Salmasi, F.R., *Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends.* Vehicular Technology, IEEE Transactions on, 2007. **56**(5): p. 2393-2404.

47. Lee, H.-D. and S.-K. Sul, *Fuzzy-logic-based Torque Control Strategy for Parallel-Type Hybrid Electric Vehicle.* IEEE Transactions on Industrial Electronics. **45**(4).
48. Rajagopalan, A., et al., *Development of Fuzzy Logic and Neural Network Control and Advanced Emissions Modeling for Parallel Hybrid Vehicles*. National Renewable Energy Laboratory: December 2003.
49. Poursamad, A. and M. Montazeri, *Design of Generic-fuzzy Control Strategy for Parralle Hybrid Electric Vehicles.* Control Engineering Practice, 2008. **2008**(16): p. 861-873.
50. Shi, G., et al. *Study and Simulation of Based-fuzzy-logic Parallel Hybrid Electric Vehicles Control Strategy*. in *the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*. 2006.
51. Salmasi, F.R., *Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison and Future Trends.* IEEE Transactions on Vehicular Technology. **56**(5).
52. Jinming, L. and P. Huei, *Modeling and Control of a Power-Split Hybrid Vehicle.* Control Systems Technology, IEEE Transactions on, 2008. **16**(6).
53. Jinming, L., P. Huei, and Z. Filipi. *Modeling and Control Analysis of Toyota Hybrid System*. in *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*. 2005.
54. Feng, L., W. Liu, and B. Chen, *Driving Pattern Recognition for Adaptive Hybrid Vehicle Control.* SAE International Journal of Alternative Powertrains, 2012. **1**(1): p. 169-179.
55. Theodoridis, S. and K. Koutroumbas, *Pattern Recognition*. 2008, maryland heights: Academic Press.
56. U.S. Environmental Protection Agency, *EPA and NHTSA Finalize Historic National Program to Reduce Greenhouse Gases and Improve Fuel Economy for Cars and Trucks*. 2010, Office of Transportation and Air Quality,.
57. Google_Maps, *Map of Houghton and Hancock, MI*. 2010.
58. Korn, F., H.V. Jagadish, and C. Faloutsos, *Efficiently supporting ad hoc queries in large datasets of time sequences.* Sigmod Record, 1997. **26**(2): p. 289-300.
59. Agrawal, R., T. Imielinski, and A. Swami, *DATABASE MINING - A PERFORMANCE PERSPECTIVE.* Ieee Transactions on Knowledge and Data Engineering, 1993. **5**(6): p. 914-925.
60. Faloutsos, C., M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases.* Sigmod Record, 1994. **23**(2): p. 419-429.
61. Kin-Pong, C. and F. Ada Wai-Chee. *Efficient time series matching by wavelets*. in *Proceedings of IEEE Computer Society 15th International Conference on Data Engineering*. 1999. Sydney, NSW, Australia: IEEE Comput. Soc.
62. Keogh, E., et al., *Locally adaptive dimensionality reduction for indexing large time series databases.* Sigmod Record, 2001. **30**(2): p. 151-162.
63. Cai, Y. and R. Ng. *Indexing spatio-temporal trajectories with Chebyshev polynomials*. in *International Conference on Management of Data* 2004. Paris, France.
64. Lin, J., et al., *Experiencing SAX: a novel symbolic representation of time series.* Data Mining and Knowledge Discovery, 2007. **15**(2): p. 107-144.

65. Pittner, S. and S.V. Kamarthi, *Feature extraction from wavelet coefficients for pattern recognition tasks.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999. **21**(1): p. 83-88.
66. Sugimura, H., et al., *Development of HEV Engine Start-Shock Prediction Technique Combining Motor Generator System Control and Multi-Body Dynamics (MBD) Models.* SAE Int. J. Passeng. Cars - Mech. Syst., 2013. **6**(2): p. 1363-1370.
67. Feng, L. and B. Chen. *Study the Impact of Driver's Behavior on the Energy Efficiency of Hybrid Electric Vehicles*. in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2013. Portland, Oregon, USA: ASME.
68. *Autonomie - Powertrain and Vehicle Model Architecture and Development Environment*. Version: 1210]. Available from: http://www.autonomie.net/.
69. Feng, L., M. Cheng, and B. Chen, *Predictive Control of a Power-Split HEV with Fuel Consumption and SOC Estimation*, in *SAE Technical Paper 2015-01-1161*. 2015.