Connected Hybrid Electrical Vehicle: Powertrain Optimization Strategy and Experiment

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Mohd Azrin Mohd Zulkefli

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Zongxuan Sun, Advisor

August 2017

Acknowledgements

It is with great honor that I would like to thank my advisor, Prof. Zongxuan Sun, for allowing me the opportunity to do research under his guidance in the PhD program at the University of Minnesota. His broad and deep technical knowledge, strategic and logical thoughts have always been presented clearly and served as guidance and an example for me to overcome countless research hurdles. Without his continuous support and encouragements, I would not be here writing this thesis today.

I would also like to extend my many thanks to the whole research team, especially to my research and lab partners, Yunli Shao, Pratik Mukherjee and Ye Lwin Oo for their diligence in all the works that we had done together. My deepest appreciation also goes to the Federal Highway Administration (FHWA) who provided us with the necessary research funding and research collaboration in the highly competitive, multi-disciplinary nature of this research topic. Also not to forget go to Dr Yu Wang who developed the powertrain research platform as part of his PhD research, which is now a component in the hardware-in-the-loop testbed for evaluating connected vehicle applications.

The last five years had been very challenging. Encouragements from individuals fuel the needed morale for me to face the countless challenges that come hand in hand in this pioneering research in engineering. It is therefore my utmost concern to express my heartfelt gratitude to my life partner Mrs Nazatul Irma Yahaya and daughter Rinnaz Adeel, my parents Mr Mohd Zulkefli and Mrs Nor Akmal, my siblings, Mohd Izar, Fatin Athirah and Farah Najiah.

Mohd Azrin Mohd Zulkefli

Dedication

This thesis is dedicated to my lovely wife and daughter.

Abstract

Power-split Hybrid Electric Vehicle (HEV), which accounts for almost 40% of US hybrid-car total sales in 2013, has the ability to store excess energy during driving and braking, and to split the demanded power between the engine and battery. With the advent of connected vehicles, traffic information can be shared and utilized to further optimize HEV's energy use, by predicting the demanded power and optimizing the power-split. However, traffic conditions, and therefore the demanded power, are constantly changing. As a result, the optimization method not only has to account for optimality and charge-sustaining conditions, but also driving-cycle sensitivity and speed of calculation for real-time implementation. This research therefore proposes fast HEV powertrain optimization to improve fuel economy for connected vehicle applications. Additionally, in order to measure the performance of connected vehicle applications, a hardware-in-the-loop system (HiLS), that combines an existing laboratory powertrain research platform with a microscopic traffic simulator, is developed. A computationally-efficient analytical solution to the HEV powertrain optimization problem utilizing vehicle speed prediction based on Inter-Vehicle Communications and Vehicle-Infrastructure Integration is proposed for real-time implementation. First, Gipps' car following model for traffic prediction is used to predict the interactions between vehicles, combined with the cell-transmission-model for the leading vehicle trajectory prediction. Secondly, a computationally efficient charge-sustaining (CS) HEV powertrain optimization strategy is analytically derived and simulated, based on the Pontryagin's Minimum Principle (PMP) and a CS-condition constraint. A 3D lookup-map, generated offline to interpolate the optimizing parameters based on the predicted speed, is also utilized to speed up the calculations. Simulations are conducted for 6-mile and 15-mile cases with different prediction update timings to test the performance of the proposed strategy against a Rule-Based (RB) control strategy on a Toyota Prius engine. Results for accurate-prediction cases show 9.6% average fuel economy improvements in miles-per-gallon (MPG) over RB for the 6-mile case and 7% improvements for the 15-mile case. Prediction-with-error cases show smaller average MPG's improvements, with 1.6% to 4.3% improvements for the 6-mile case and 2.6% to 3.4% improvements for the 15-mile case. For practical purposes, the HEV engine operating range and transient response have to be considered, which introduces additional optimization constraints. Solving a nonlinear optimization problem with constraints analytically is difficult, while numerically

is computational heavy and time consuming. Therefore, the nonlinear HEV optimization problem with constraints is expressed and solved as a Separable Programming (SP) problem. First, given the flexibility of the power-split HEV powertrain, the relationship between the minimum fuel consumption and the power-split levels between the engine and battery is calculated and stored offline for all possible vehicle power demands. Therefore, the relationship between HEV power-split levels and engine operating points with minimum fuel consumption for a given vehicle power demand is obtained. Secondly, the problem is formulated with fuel consumption as the cost and power-split level as the optimizing input and solved using SP. In SP, the nonlinear fuel cost and battery charging rate relationships with the power-split levels are approximated as linear-piecewise functions which introduce dimensionless variables that are linear to the input and outputs of the nonlinear functions. The input range constraint and the engine transient dynamics are also formulated. The optimization problem is then solved as a large-dimension linear problem with linear constraints using efficient Linear Programming solvers. The proposed optimization method is then simulated in a receding horizon fashion with various vehicle speed profiles and a case study was tested on a real John Deere diesel engine. Comparable fuel economy with Dynamic Programming is shown with significantly less calculation time and fuel savings of 4.0% and 10.4% over PMP and RB optimizations are observed. A HiLS testbed to evaluate the performance of connected vehicle applications is proposed. A laboratory powertrain research platform, which consists of a real engine, an engine-loading device (hydrostatic dynamometer) and a virtual powertrain model to represent a vehicle, is connected remotely to a microscopic traffic simulator (VISSIM). Vehicle dynamics and road conditions of a target vehicle in the VISSIM simulation are transmitted to the powertrain research platform through the internet, where the power demand can then be calculated. The engine then operates through an engine optimization procedure to minimize fuel consumption, while the dynamometer tracks the desired engine load based on the target vehicle information. Test results show fast data transfer at every 200 milliseconds and good tracking of the optimized engine operating points and the desired vehicle speed. Actual fuel and emissions measurements, which otherwise could not be calculated precisely by fuel and emission maps in simulations, are achieved by the testbed. In addition, VISSIM simulation can be implemented remotely while connected to the powertrain research platform through the internet, allowing easy access to the laboratory setup.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Background

1.1 Introduction of HEV Powertrain and Connected Vehicles

Sustainable energy supply and the environmental impact are the main concerns in the transportation sector. The Energy Information Administration reported that in the U.S., transportation encompassed a 28% share of energy consumption and a 72 % share of total petroleum consumption in 2010 [1]. The concerns are intensified especially with escalating traffic congestion problems. According to the 2011 Urban Mobility Report [2], in 2010, a driver spent an average of 34 extra hours on the road due to traffic congestion, corresponding to $713 of wasted fuel. In response to these problems, the government through the Energy Policy and Conservative Act of 1975 (EPCA) introduced the Corporate Average Fuel Economy (CAFE) which targets the average fuel economy standards for passenger vehicles at 37.8 MPG in 2016 and 54.5 MPG in 2025 [3].

In response to increasing fuel economy regulation, innovative vehicles with higher fuel economy were delivered to the market by automotive manufactures, where the hybrid electric vehicles (HEVs) are the first to be introduced and most pervasively available in the market today. Different from conventional vehicles with single power source, namely gasoline or diesel engine, a HEV powertrain is equipped with an additional power storage device, for example, a battery [4,5]. The secondary power source could supplement or replace the engine as the source of power to avoid inefficient engine operation regions, particularly at low power demands. This is particularly important for urban driving conditions with multiple stop-and-go scenarios. Consequently, the overall fuel consumption can be potentially improved up to 30% to 40% over conventional vehicles [6].

Connected Vehicles (CV) technologies utilizing Inter Vehicle Communication (IVC) and Vehicle Infrastructure Integration (VII) [7,8] have gained attentions to improve traffic safety and mobility. The Dedicated-Short-Range-Communication (DSRC) used for traffic communication has been proven to be reliable [9-11] and field works have been done to evaluate the scalability, security and interoperability of DSRC communications in a real world setting [12]. Rigorous tests were also done to investigate DSRC communication reliability under different cooperative active safety applications [13]. A CV environment where IVC and VII work seamlessly through DSRC is shown in Fig. 1.1. Traffic data including Signal Phase and Timing (SPaT) from traffic light, vehicle detector from the road and vehicles information can be transferred to and from the vehicles and the traffic center. Reliable IVC and VII therefore make it possible to utilize the abundance of traffic information for numerous connected vehicle applications, including for vehicle fuel consumption and emissions improvements by merging traffic information with powertrain optimization.

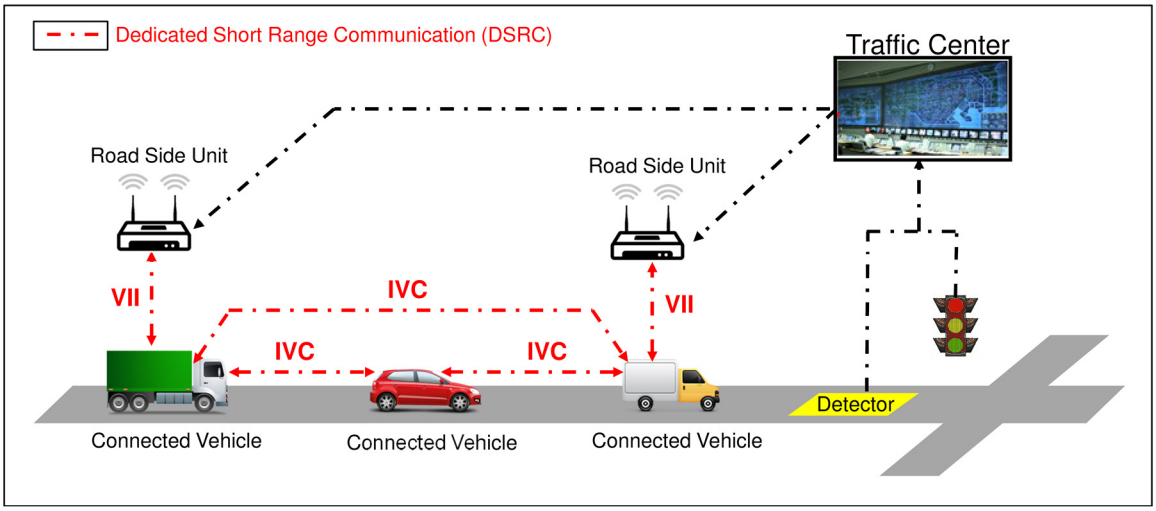


Figure 1.1. Connected Vehicle Environment

HEV offers a solution to meeting the increasing fuel efficiency requirements by combining different power sources and operating them in the most efficient manners for any given vehicle load. With the introduction of IVC and VII under the connected vehicle (CV) framework [12], vehicle loads across a time horizon can be estimated using shared traffic information. Therefore, incorporating traffic information in HEV powertrain optimization within a CV environment will increase fuel saving potentials.

1.2 HEV Powertrain Optimization Methods

Numerous optimization methods have been developed for HEVs to optimize fuel use. Dynamic Programming (DP) is widely used to optimize the HEV [14-16] but the heavy computations and non-causality of the method prevents its application in a transient traffic environment. Stochastic-DP is proposed [17] to alleviate the restrictive nature of DP, but still does not address the heavy computation issue. High computational burden due to the nonlinearities leads to linearization efforts in Model Predictive Control (MPC) [18]. In [19] feedback linearization is used in combination with MPC to linearize the plant and optimize a quadratic approximated nonlinear cost through Quadratic Programming (QP). This however resulted in nonlinear constraints and the problem is solved iteratively or using inaccurate approximations. Online implementation of MPC with QP is used in [20] by linearizing the plant and constraints at every time step, but the optimality is affected by the short prediction to reduce QP computations. Realizing the need for a fast optimization strategy, the Equivalent Consumption Minimization Strategy (ECMS) is devised where a combined actual-fuel and equivalent-fuel from the motor/generator is minimized [14,15,21]. ECMS can be implemented online, but the equivalent-ratio (ER), which is a gain multiplied to the electrical energy consumed by the motor/generator to find the equivalent-fuel use, has to be iterated offline to ensure Charge-Sustaining (CS) HEV operation. In [22], it is pointed out that the Pontryagin's Minimum Principles (PMP) framework can be shown as the underlying optimization principles for ECMS, but online implementation is infeasible due to the iterations in finding the initial value of the dynamic-ER for CS-operation. Similar PMP based method is also used in [23] with a constant ER, which is found numerically for CS-operation using multiple-shooting method. Considering the transient traffic system and the need for a fast optimization procedure that ensures CS-operation, the Adaptive-ECMS (A-ECMS) is developed [15,16] and is further implemented [24,25]. In A-ECMS, the ER is repeatedly adapted for the same driving cycle based on the final battery State-Of-Charge (*SOC*) level. The adapted-ER, however, cannot ensure a CS-operation if the subsequent driving cycles are different. Numerical methods are also formulated with linearized cost and constraints to minimize the computation efforts. In [26] a linear battery model and a two-region piecewise linear cost function for a Series-HEV are used to optimize the powers for engine, braking and battery using Linear Programming (LP). However, approximating a nonlinear battery with a simplified linear model introduces approximation errors which

affect the optimization results. Similarly, [27] utilizes the battery current to linearize the fuel cost within a range of input values and solve the linear problem with LP. This introduces approximation errors in the cost and limits the operating range of the input. Results in [28] extend the application of [27] using Mixed Integer Linear Programming (MILP) for Parallel-HEV (PHEV) to allow engine declutching for engine shutoff. Similarly, in [29] the engine power levels are discretized and MILP is used to optimize the PHEV operations.

1.3 HEV Powertrain Optimization with Traffic Information

Studies have been conducted to integrate traffic information into HEV powertrain optimization in CV settings. In [24], researchers investigated fuel efficiency and emission improvements by implementing a VII on a series HEV and plug-in hybrid electrical vehicle (PHEV) which is then interpolated to a network-level cost-benefit analysis with a 15-years projection to determine the minimum penetration rate. In [25], the authors analyze improvements of a power-split PHEV across different prediction lengths, total mileages, driving cycles and average prediction errors. In [30,31], the authors utilize the preceding vehicle speed and simple kinematic equations to recalculate a less aggressive future vehicle speed for fuel economy. Authors in [14] rely on road-grade information with HEV energy management strategy (EMS) to save fuel. In [32], the authors estimate the total change of battery State-of-Charge (*SOC*) in a pure electric vehicle (EV) by calculating the vehicle loads using a constant-acceleration probabilistic model in road segments from historical traffic data. Integrating traffic information with powertrain optimization can offer fuel benefits, but the transient nature of traffic information necessitates the development of a fast HEV optimization for real-time implementation.

1.4 Powertrain Optimization Evaluation and Hardware-in-the-Loop Systems for Traffic Environment

Current methods to measure the performance of a vehicle's fuel economy and emissions in traffic are done by either simulation, utilizing fuel consumption and emissions maps or by instrumenting the vehicle, but there are drawbacks of both approaches. A simulation-based approach usually employs steady-state fuel-use and emission maps as a function of the engine torque and speed, which are inaccurate compared to actual measurements especially during engine transients [33,34]. Furthermore, sophisticated CFD model and

chemical kinetics are required to model emissions such as NO_X and soot accurately [35,36]. Averaged emissions rates corresponding to vehicle power demand levels are used in the U.S. Environmental Protection Agency's software package, MOVES, to estimate vehicles emissions, but comparisons with actual measurements shows high variability [37]. In [38], a driving simulator is utilized with human drivers to investigate the benefits of eco-driving, where the fuel use and emissions are estimated from MOVES. Similarly, the energy consumption and emissions are estimated from MOVES in an integrated traffic and driving simulators in [39]. In [40], high-resolution event data such as traffic light status and vehicle queues collected from real roads is used to optimize the speed trajectory of a virtual vehicle for fuel benefit, which is estimated from the vehicle power demand. The difficulty of conducting real field tests for connected vehicle applications prompted the use of simulation approach. Moreover, different components of the complex traffic environment need to be simulated accurately to provide realistic and meaningful results.

In order to integrate different components of a traffic environment, integrated simulators and HiLS approaches are used. In [39], PARAMICS traffic simulator is integrated with a driving simulator in real-time for connected vehicle and Vehicle Ad-hoc Network (VANET) studies. Traffic routing strategy based on energy and travel time in MATLAB-Simulink is implemented in [41], which is coupled with VISSIM traffic simulator for simulating traffic dynamics. In order to simulate a realistic car-to-car communication, the NS-2 network simulator is integrated and synchronized with CARISMA traffic simulator through TCP connection in [42]. Similar platform was developed that synchronized PARAMICS traffic simulator with NS-2 network simulator to implement travel-time prediction methods using artificial intelligence in [43]. In some cases, a component of traffic environment cannot be modeled accurately by simulation, prompting the use of a HiLS that integrate actual hardware with simulation packages. For example, in order to evaluate the actual performance of implementing complex traffic signal control algorithm on actual signal controller, [44] utilizes CORSIM traffic simulator and sends simulated detector information to a physical signal controller. Similar HiLS structure is developed that integrate VISSIM traffic simulation with Econolite ASC/2-2000 signal controller in [45]. The main objective of the proposed HiLS is to accurately evaluate fuel and emissions performance of connected vehicle applications. Powertrain dynamics can be modeled accurately through simulation such as the ones developed by the NREL

[46,47], while calibration of traffic simulation software can provide reasonable estimate of actual traffic dynamics [48]. However, as discussed, fuel consumption and emissions models are very complex and cannot be accurately simulated. Therefore, a real engine has to be utilized to provide accurate fuel and emissions measurements in real-time.

For accurate measurements, [49] utilized a real engine with an engine loading device to measure the fuel and emissions performances of an eco-driving approach with predefined standard driving cycles which however lack realistic background traffic dynamics. In order to incorporate real traffic dynamics while measuring actual fuel consumption and emissions, road vehicles are instrumented. However, instrumenting a vehicle is difficult and time consuming [37,50]. With a HiLS, different vehicles can be tested quickly and flexibly by changing the engine and the load settings on the hydrostatic dynamometer. Equipping large precision measurement devices on large vehicles such as buses [51] or trucks [52,53] may be feasible, but would be challenging for smaller passenger vehicles. Smaller portable measurement devices can be used, but are less accurate, especially during engine transients and require calibrations for different driving cycles [54,55]. Finally, testing connected vehicle technologies in a simulated but realistic traffic is more economical without having to instrument multiple vehicles and safer than in real traffic where uncertainties are present [56,57].

1.5 Thesis Outline and Contributions

In Chapter 2, a combined approach of a time-efficient powertrain optimization strategy, utilizing vehicle speed trajectory prediction based on IVC and VII is proposed. A computationally efficient charge-sustaining (CS) HEV powertrain optimization strategy is analytically derived and simulated, based on the Pontryagin's Minimum Principle and a CS-condition constraint. A 3D lookup-map, generated offline to interpolate the optimizing parameters based on the predicted speed, is also utilized to speed up the calculations. Simulations are conducted for 6-mile and 15-mile cases with different prediction update timings to test the performance of the proposed strategy against a Rule-Based (RB) control strategy. This work is also documented in [58,59].

In Chapter 3, a fast HEV powertrain optimization that considers the associated constraints and engine dynamics is proposed. The HEV powertrain optimization problem is solved in two steps. First, given the flexibility of the power-split HEV powertrain, the

relationship between the minimum fuel consumption and the power-split levels between the engine and battery is calculated and stored offline for all possible vehicle power demands. This relationship ensures the engine is operating at the most optimal point for every possible HEV power-split level for a given vehicle power demand. Then, the nonlinear fuel cost and battery dynamics are approximated by linear piecewise functions and formulated as a Separable Programming problem. The piecewise-linear functions introduce new dimensionless variables which are solved as a large-dimension constrained linear problem with efficient LP solvers. The engine operating range and engine transient dynamics are represented as linear constraints in the LP problem to ensure the engine operating points are feasible. Traffic information from CV is integrated in the optimization by integrating the driving cycle prediction into the powertrain optimization. Fast calculation time allows the optimization method to be implemented with repeated speed prediction updates due to the transient nature of traffic. Simulations results with accurate speed prediction are compared with DP, PMP and RB methods and experimental results show the feasibility of the optimized engine operating points. Simulation studies with different levels of vehicle speed prediction errors to emulate CV settings are also presented. Preliminary work is published in [60] and a more detailed simulation and experimental work has been submitted for another publication.

In Chapter 4, A Hardware-in-the-Loop-System (HiLS) is proposed to offer the flexibility and accuracy of evaluating the performance of connected vehicle applications. The HiLS is comprised of a microscopic traffic simulator (VISSIM) and a laboratory powertrain research platform. VISSIM is used to simulate a traffic network while the powertrain research platform, which consists of a real engine, an engine-loading device (hydrostatic dynamometer) and a virtual powertrain model is used to represent a single vehicle. A connected vehicle application such as the Cooperative Adaptive Cruise Control (CACC) can be simulated in VISSIM, where a target vehicle is selected to be represented by the powertrain research platform. This is done by sending the simulated target vehicle speed and road condition information from VISSIM to the powertrain research platform in real-time during simulation. This information is used to calculate the vehicle load demand, which is realized by the engine and powertrain. Fuel consumption and emissions from the engine are measured by precise laboratory equipment. The HiLS utilizes a real engine for direct fuel and emission measurements. Furthermore, different vehicles can be tested quickly and flexibly by changing the engine and the load settings on the

dynamometer. The HiLS can also accommodate large precision measurement devices since it is built in a laboratory setting. Testing connected vehicle applications in a simulated but realistic traffic is more economical without having to instrument multiple vehicles. It is also safer and bypasses the legalities that would otherwise hamper the evaluation of connected vehicle applications in real traffic. Experimental results from the testbed are documented in in [61,62].

Chapter 5 outlines the summary and possible future works.

Chapter 2

PMP-Based Unconstrained HEV Powertrain Optimization

2.1 Motivation

Recent advances in Connected Vehicle (CV) paved ways to real-time information sharing between vehicles. With Inter-Vehicle-communication (IVC) and Vehicle-Infrastructure-Integration (VII), vehicle loads can be better predicted across a future time window, which can be utilized in vehicle energy management strategies (EMS) to save fuel. This is especially valuable for Hybrid Electrical Vehicle (HEV) where the powertrain is equipped with an additional power storage device, namely a battery. The battery could assist, store or replace the engine power to avoid the engine from operating at inefficient operation regions. Particularly, a HEV with power-split powertrain architecture employs a continuous-variable-transmission (CVT), which gives greater flexibility in selecting any desired engine operating points for any vehicle load. However, considering the physical constraints and nonlinearity of complex systems such as the internal combustion engine and the battery, optimizing the HEV powertrain using traditional optimization methods are computationally heavy. Furthermore, traffic environment is transient, where traffic states change all the time, affecting the vehicle load predictions from IVC and VII. Certainly, researches in this area have been done, but tradeoffs between optimality, driving-cycle sensitivity, speed of calculation and charge-sustaining (CS) conditions have not been cohesively addressed before. In light of this, a combined approach of a time-efficient powertrain optimization strategy, utilizing vehicle speed trajectory prediction based on IVC and VII is proposed.

2.2 Vehicle Speed Trajectory Prediction Using VII and IVC

With recent advances of "Intelligent Drive" utilizing IVC and VII, shared real-time traffic information can be made available between vehicles. Using information sharing between IVC and VII infrastructures, the proposed trajectory prediction model will mainly be based on Gipps' car following model if there is a leading vehicle within the communication range. If the leading vehicle is not in presence, the vehicle will utilize the classical cell transmission model (CTM) for trajectory predictions. The details of the trajectory prediction iteration are shown in Table 1.1.

Table 1.1. Online Iteration For Vehicle Speed Prediction

Prediction iterations for individual vehicle
1. Get current speed and position on the route.
2. If leading vehicle is available (within communication range of 300 meters), go to Step 3, else go to Step 5.
3. Retrieve predicted trajectory from leading vehicle, and use Gipps' model for prediction.
4. If predicted trajectory covers the whole trip to the destination, go to Step 7, else go to Step 5
5. Retrieve speed data along the route from traffic center (CTM).
6. Predict rest of trip based on speed data from RSU.
7. Broadcast the predicted trajectory, end this iteration.

Using the information provided by the transmitted future information, each vehicle then processes the information independently, by utilizing an optimization algorithm. To improve vehicle speed prediction accuracy, an offline parameter learning for individual driver with regards to the car following model is implemented. The predicted vehicle speeds are used to optimize the HEV powertrain, while the vehicle still follows the unperturbed actual vehicle speeds. Detailed models of the vehicle speed prediction can be found in [59], which is the work of a co-author of the journal.

2.3 Power-Split HEV Model

The vehicle dynamics in this work utilizes the Toyota Hybrid System (THS) power-split architecture [5], indicated in Fig. 2.1. The power-train uses a planetary gear set to connect the engine, and two electrical machines, both can either work as a motor or generator, to allow two degrees-of-freedom in meeting the speed and torque demands. The requested torque at the wheels is given by Eq. (2.1) and static equations relating the torques and speeds of the engine and the two electrical machines are given by Eq. (2.2)–(2.5) [23,63].

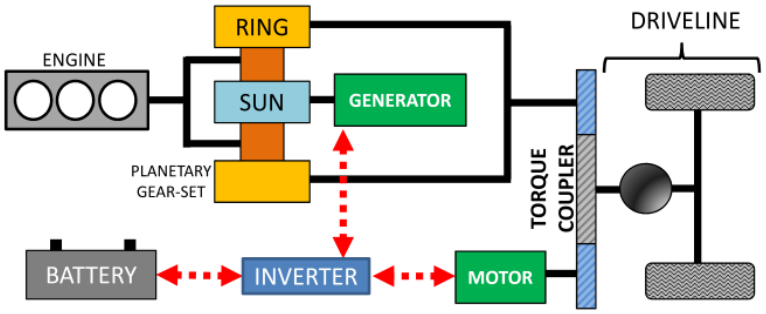


Figure 2.1. Toyota Hybrid System Power-Split Architecture

$$T_v = R_{tire} \cdot (f_{tire} M_v g + 0.5 \rho_{air} C_{drag} A_f \omega_v^2 R_{tire}^2 + M_v \dot{\omega}_v R_{tire}) \quad (2.1)$$

$$\omega_m = K \omega_v \quad (2.2)$$

$$\omega_e (R + S) = \omega_g S + \omega_m R \quad (2.3)$$

$$T_m = \frac{T_v}{K} - T_e \left(\frac{R}{S+R}\right) \quad (2.4)$$

$$T_g = -T_e \left(\frac{S}{S+R}\right) \quad (2.5)$$

Both electrical machines are connected to a battery through an inverter, where the net-power-from-battery is a function of the speeds and torques of both electrical machines.

$$P_{batt} = \eta_m^{km} \omega_m T_m + \eta_g^{kg} \omega_g T_g \quad (2.6)$$

k_m and k_g determine the efficiency terms of the electrical machines, whether to be multiplied or divided with the powers of the electrical machines, depending on the electrical machine operation; as a motor (k_m or k_g= -1) or a generator (k_m or k_g = +1). The power of the electrical machine is positive as a motor and negative as a generator. The dynamics of the battery state of charge is a function of P_{batt}.

$$S\dot{O}C = -\frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_{batt}P_{batt}}}{2R_{batt}Q_{batt}} \tag{2.7}$$

2.4 Optimization Problem and Framework

The Confined Optimal Operating Line (C-OOL) framework in [23] is used to find the correlation between the minimum $\dot{m}_{fuel}$ and P_{batt}. Given a time window $[t_0, t_f]$, fuel consumption cost can be written as

$$J = \int_{t_0}^{t_f} \dot{m}_{fuel} dt \tag{2.8}$$

where $\dot{m}_{fuel}$ is the mass fuel rate which depends on the engine operating points (T_e, ω_e), between $[t_0, t_f]$. Given ω_v, the value of T_v and ω_m can be calculated in Eq. (2.1, 2.2). At time t, $P_{batt}(t)$ candidates can be iterated by first setting the (T_e, ω_e) at the extremes of an engine map (from highest to lowest fuel consumption). Using the (T_e, ω_e) point, T_v, ω_m and Eq. (2.1)–(2.5), the motor/generator torques and speeds can be calculated, hence the $P_{batt}(t)$ limits $[P_{batt\_min}(t), P_{batt\_max}(t)]$ from Eq. (2.6). $P_{batt}(t,k)$ values are then iterated between these two limits.

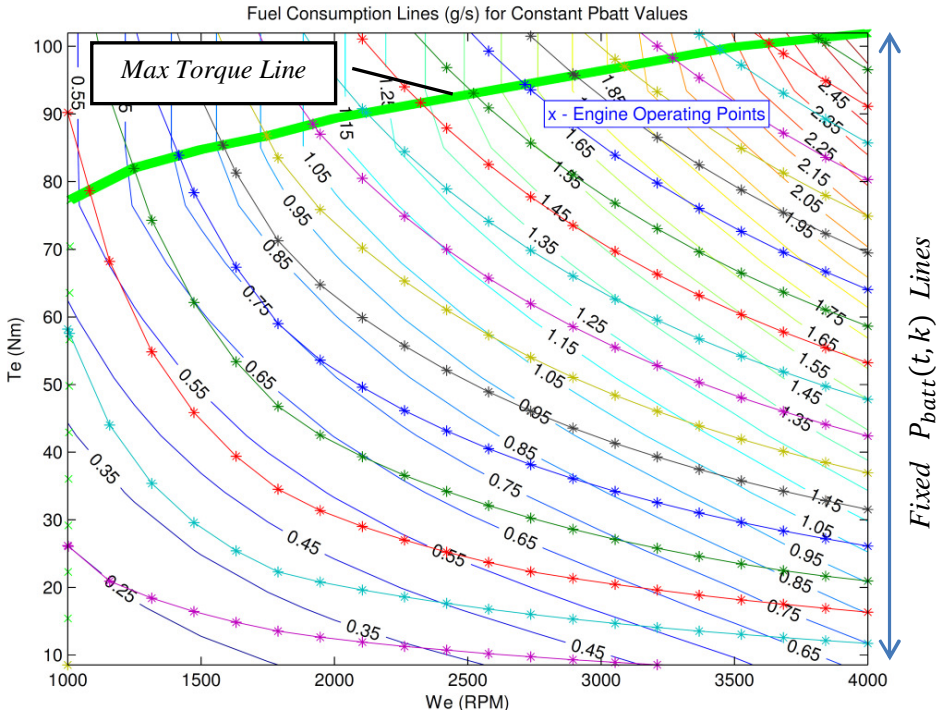


Figure 2.2. Fuel Consumption Map with fixed $P_{batt}(t,k)$ lines (with asterisks)

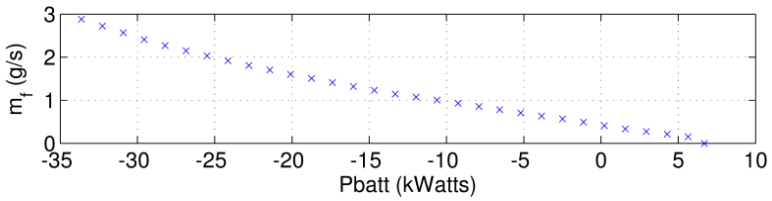


Figure 2.3. Minimum $\dot{m}_{fuel}$ vs P_{batt}

Given a $P_{batt}(t,k)$ value, Eq. (2.4)–(2.6) are combined to simultaneously solve the torques as in Eq. (2.9). Note that ω_e is iterated between the engine map speed limits, while possible combinations of k_m and k_g are also iterated to find the working operations of the electrical machines. Therefore, for a given $P_{batt}(t,k)$ value, ω_e can be iterated to find the pair (T_e, ω_e) with the lowest fuel rate $\dot{m}_{fuel}(t,k)$. Shown in Fig. 2.2 are the constant $P_{batt}(t,k)$ lines (with asterisks) when ω_e is iterated across the engine map. The engine fuel rate in grams/second is also shown by the contour lines without asterisks, while the bold green line is the maximum torque line. The process is repeated for all $P_{batt}(t,k)$ candidates between the $P_{batt}(t)$ limits. The resulting minimum $\dot{m}_{fuel}(t,k)$ vs $P_{batt}(t,k)$ plot is shown in Fig. 2.3. Another P_{batt} point is also calculated to represent engine shut-off, called $P_{batt\_shutoff}(t)$, by using $(T_e, \omega_e) = (0,0)$ in Eq. (2.1)–(2.6) with a fuel rate of zero, as shown by the lowest point in Fig. 2.3.

$$\begin{bmatrix} T_g \\ T_m \\ T_e \end{bmatrix} = \begin{bmatrix} \eta_g^{kg}\omega_g & \eta_m^{km}\omega_m & 0 \\ 0 & 1 & \left(\frac{R}{S+R}\right) \\ 1 & 0 & \left(\frac{S}{S+R}\right) \end{bmatrix}^{-1} \begin{bmatrix} P_{batt} \\ \left(\frac{T_v}{K}\right) \\ 0 \end{bmatrix} \tag{2.9}$$

Therefore, at every time step, there is a fixed mapping between minimum $\dot{m}_{fuel}$ and P_{batt}. As a result, Eq. (2.8) can be represented as

$$J = \int_{t_0}^{t_f} \dot{m}_{fuel}(P_{batt}(t))dt \tag{2.10}$$

$P_{batt}(t)$ is essentially the input which can be chosen at every time step to minimize the cost in Eq. (2.10) and to maintain the battery SOC in Eq.(2.7). An overview of the optimization principles and its implementation in this framework are presented next.

2.5 Pontryagin's Minimum Principles (PMP) Overview

This chapter will explain the optimization problem settings and the PMP necessary conditions as discussed in [64]. Readers are invited to refer to detailed derivations in Appendix 1. Consider an unconstrained optimization with open final time. The augmented cost function is given by

$$I = \phi(x_f, t_f, v) + \int_{t_0}^{t_f} L(x,u,t)dt \tag{2.11}$$

$L(x,u,t)$ is the cost function and $\phi(x_f, t_f, v)$ is the augmented terminal cost given by

$$\phi(x_f, t_f, v) = \phi(x_f, t_f) + v^T \psi(x_f, t_f) \tag{2.12}$$

where $\phi(x_f, t_f)$ and $v^T\psi(x_f, t_f)$ are the soft and hard terminal constraints. v^T is the Lagrangian multiplier that drives the terminal condition $\psi(x_f, t_f)$ to zero. In this case, v^T is constant because it is responsible to drive only the final state to zero. The states dynamics are given by

$$\dot{x} = f(x,u,t) \text{ with initial conditions } x(t_0) = x_0 \tag{2.13}$$

Therefore, the Hamiltonian is stated as

$$H = L(x,u,t) + \lambda^T f(x,u,t) \tag{2.14}$$

where λ^T is the co-states. Given the problem settings in Eq. (2.11)-(2.14), and assuming the problem is convex [23], the necessary conditions that minimize Eq. (2.11) are given by the adjoint, system and controls equations, including the terminal and transversality conditions in Eq. (2.15)–(2.19). Detailed derivations of these conditions can be found in Appendix 1.

$$\dot{\lambda} = -H_x^T \text{ and } \phi_{x_f} = \lambda^T(t_f) \tag{2.15}$$

$$\dot{x} = f(x,u,t) \text{ and } x(t_0) = x_0 \tag{2.16}$$

$$H_u = 0 \tag{2.17}$$

$$(\phi_t + H)|_{t_f} = 0 \tag{2.18}$$

$$\psi(x_f, t_f) = 0 \tag{2.19}$$

2.6 Implementation of PMP with Real-Time Traffic Information

A fast optimization algorithm is needed to optimize the powertrain with traffic predictions in real-time. Therefore, in this chapter, the optimization is carried out by first fitting the powertrain optimization framework in Chapter 2.4 into the PMP optimization problem in Chapter 2.5. Then the optimization problem is solved analytically using PMP necessary conditions in Eq. (2.15)-(2.19). The cost function in Eq. (2.10) is augmented by introducing a hard terminal constraint in Eq. (2.12) to ensure $SOC(t_f)$ reaches a desired SOC_f at the end of a prediction cycle

$$I = \beta[SOC(t_f) - SOC_f] + \int_{t_0}^{t_f} \dot{m}_{fuel}[P_{batt}(t)]dt \tag{2.20}$$

where the states equation is given by

$$\dot{x} = S\dot{O}C[P_{batt}(t)] = -\frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_{batt}P_{batt}(t)}}{2R_{batt}Q_{batt}} \tag{2.21}$$

with $x(t_0) = SOC_0$ as the initial conditions. Note that both the cost and the state dynamics are functions of the input $P_{batt}(t)$. The Hamiltonian in Eq. (2.14) can be expressed as

$$H = \dot{m}_{fuel}(P_{batt}(t)) + \lambda S\dot{O}C(P_{batt}(t)) \tag{2.22}$$

Using the Hamiltonian above in the adjoint Eq. (2.15) give

$$\dot{\lambda} = -H_x = -\left[\dot{m}_{fuel}(P_{batt}(t)) + \lambda S\dot{O}C(P_{batt}(t))\right]_{SOC} = 0 \tag{2.23}$$

$$\lambda(t_f) = \phi_{x_f} = \frac{d}{dSOC(t_f)}\left[\beta(SOC(t_f) - SOC_f)\right] = \beta \tag{2.24}$$

From Eq. (2.23) and (2.24), it can be deduced that

$$\lambda(t) = \beta = \text{constant} \tag{2.25}$$

The derivative of the Hamiltonian with respect to input $P_{batt}(t)$ in Eq. (2.17) gives

$$H_u = \frac{d}{dP_{batt}(t)}\left[\dot{m}_{fuel}(P_{batt}(t))\right] + \lambda \frac{d}{dP_{batt}(t)}\left[S\dot{O}C(P_{batt}(t))\right] \tag{2.26}$$

The first term cannot be derived analytically and has to use approximation. Recall in Chapter 2.4, the C-OOL mapping between minimum $\dot{m}_{fuel}$ and P_{batt} can be found at every time step. The relationship between minimum $\dot{m}_{fuel}$ and P_{batt} was found to be almost linear (Fig. 2.3)

$$\dot{m}_{fuel}(t,k) = a_0(t)P_{batt}(t,k) \tag{2.27}$$

$a_0(t)$ is the approximated linear slope for a requested vehicle load at time t, while k is the index of the iterated $P_{batt}(t)$ candidates (Chapter 2.4). In a sense, the approximated $a_0(t)$ represents the impact in fuel use $\dot{m}_{fuel}$ when choosing different amount of battery power, P_{batt}, for a given vehicle power demand. Therefore, a mapping of a_0-slope at different vehicle power demands, calculated from different vehicle speeds and accelerations, can be done offline (Fig. 2.4).

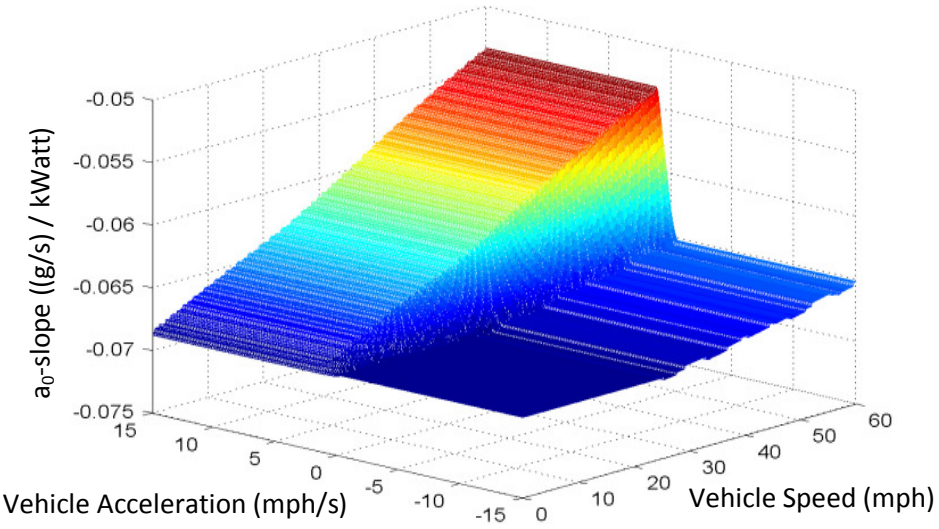


Figure 2.4. a_0-slope Look-up Map

The 2nd term in Eq. (2.26) can be differentiated analytically from Eq. (2.7). Setting Eq. (2.26) to zero and solving for $P_{batt}(t)$ gives

$$P_{batt}(t) = \frac{1}{4R_{batt}}\left[V_{oc}^2 - \left(\frac{\lambda}{a_0(t)Q_{batt}}\right)^2\right] \tag{2.28}$$

Note that in Eq. (2.28), the constant-λ from Eq. (33) has not been determined yet. λ however, can be calculated when charge sustaining operation in Eq. (2.29) is specified by setting the sum-of-$S\dot{O}C$ between $[t_0, t_f - dt]$ to be zero as in Eq. (2.30).

$$SOC(t_f) = SOC(t_0) \qquad (2.29) \quad \Big| \quad \sum_{t_0}^{t_f - dt} S\dot{O}C(t) = 0 \qquad (2.30)$$

Replacing $P_{batt}(t)$ in Eq. (2.28) into the SOC-dynamics in Eq. (2.7) gives

$$S\dot{O}C(t) = \left[\frac{\lambda}{2Q_{batt}^2 R_{batt} a_0(t)}\right] - \left[\frac{V_{oc}}{2Q_{batt}R_{batt}}\right] \qquad (2.31)$$

Equation (2.31) is the SOC-dynamics at a specific time. Summing all $S\dot{O}C(t)$ between $[t_0, t_f - dt]$

$$\sum_{t_0}^{t_f - dt} S\dot{O}C(t) = \left[\frac{\lambda}{2Q_{batt}^2 R_{batt}} \sum_{t_0}^{t_f - dt} \left(\frac{1}{a_0(t)}\right)\right] - \left[\left(\frac{t_f - t_0}{dt}\right)\frac{V_{oc}}{2Q_{batt}R_{batt}}\right] \qquad (2.32)$$

The left hand side of Eq. (2.32) represents the sum-of-$S\dot{O}C(t)$ at the end of the predicted trip. This term represent the amount of sum-of-$S\dot{O}C(t)$ that will be reached at the end of a predicted trip. This value would be zero for charge sustaining operation if there is only a single prediction cycle. However, in this work, the prediction will be updated repeatedly at a fixed time intervals before the driving cycle ends. Consequently, the sum-of-$S\dot{O}C(t)$ at the start of the current prediction time will be non-zero because of the sum-of-$S\dot{O}C(t)$ accumulated from the start of the trip, up to the current update time. The amount of sum-of-$S\dot{O}C(t)$ that need to be achieved by the end of the current prediction length, to compensate the balance from the previous cycles is therefore, the negative of the sum-of-$S\dot{O}C(t)$ from initial time to the update time. Therefore, the left hand side term can be replaced as the negative of the balance sum-of-$S\dot{O}C(t)$ at update time.

$$-\sum_{t_0} S\dot{O}C = \left[\frac{\lambda}{2Q_{batt}^2 R_{batt}} \sum_{t_0}^{t_f - dt} \left(\frac{1}{a_0(t)}\right)\right] - \left[\left(\frac{t_f - t_0}{dt}\right)\frac{V_{oc}}{2Q_{batt}R_{batt}}\right] \qquad (2.33)$$

Solving for λ from Eq. (2.33) yields

$$\lambda = 2Q_{batt}^2 R_{batt} \left[\sum_{t_0}^{t_f - dt} \left(\frac{1}{a_0(t)}\right)\right]^{-1} \left[\left(\frac{t_f - t_0}{dt}\right)\left(\frac{V_{oc}}{2Q_{batt}R_{batt}}\right) - \sum_{t_0} S\dot{O}C\right] \qquad (2.34)$$

Using Eq. (2.34), a constant λ is solved from future information of $a_0(t)$ between $[t_0, t_f - dt]$. λ is then used with the current $a_0(t)$ to calculate the optimal input at the current time $P_{batt}(t)$, given by Eq. (2.28). Note that in Eq. (2.33) and (2.34), the term t_0 now represents the current update time.

The main contributions of the proposed optimization method come from the explicit derivations of a constant λ in Eq. (2.34) and control input $P_{batt}(t)$ in Eq. (2.28) by combining the necessary conditions in Pontryagin's Minimum Principles in Eq. (2.23, 2.24, 2.26) and the charge-sustaining condition in Eq. (2.30). This is made possible by approximating the constant a_0-slopes, found in the C-OOL framework [23], in order to solve the first term in Eq. (2.26) analytically. In addition, generating a lookup map of a_0-slopes from different vehicle power demands offline also further reduces the optimization computation time. A fast optimization method is necessary to optimize the HEV powertrain in real time which utilizes predicted traffic information that changes at short time intervals. In [58], the measured max calculation time is 2.6s for a 100s prediction cycle with a 1s time-step, using Intel® 2.66 GHz Core-2-Duo™ processor. Given the speed, this method could potentially be used in real-time.

2.7 P_{batt} Constraints

The optimization method explained in Chapters 2.5 and 2.6 is based on an unconstrained optimization method. However, due to physical limitations of the engine, there exist constraints on input P_{batt}, depending on the power demand. Since P_{batt} is a measure of power drawn from the battery, the upper limit represents the maximum power that can be drawn from the battery to overcome a power request, or the engine shut-down point, $P_{batt\_shutoff}$. Slightly below the $P_{batt\_shutoff}$ is the P_{batt_max} point, where the engine is operating at its lowest point (lower left corner of engine map in Fig. 2.2). Note that the engine cannot operate between these two points. The lower limit, P_{batt_min} is calculated from the highest operating point of the engine (upper right corner of engine map in Fig. 2.2). P_{batt_min} value is negative, meaning the battery is being charged.

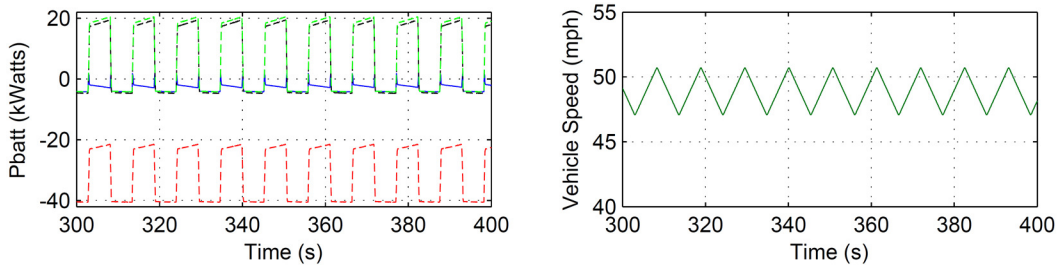


Figure 2.5. P_{batt} Limits and Vehicle Speed

Figure 2.5 shows a sample of the P_{batt} lines for a predicted speed. The green, black, blue and red lines represent $P_{batt\_shutoff}$, P_{batt_max}, calculated P_{batt} and P_{batt_min}.

2.8 P_{batt} Adjustment

Since we are using an unconstrained optimization method, some of the calculated P_{batt} points could exceed the $P_{batt\_shutoff}$ upper limit. This is because $a_0(t)$ values, which are used when calculating λ and P_{batt}, do not contain any information on P_{batt} limits. Although $P_{batt\_shutoff}$, P_{batt_max} and P_{batt_min} limits were calculated before, they were used only to define the range for P_{batt} iterations, in order to calculate the slope a_0 (Chapter 2.4). The slope a_0 represents the impact of changing input P_{batt} to fuel consumption for a given vehicle power request and the optimization procedure make use of this information in λ and P_{batt} calculations in Eq. (2.28) and (2.34) without considering the upper or lower limits of P_{batt}. It is interesting to see that if the calculated P_{batt} is higher than $P_{batt\_shutoff}$, the resulting $\dot{m}_{fuel}$ is negative. However, since the derivation of optimal P_{batt} is based on unconstrained optimization formulations, this physical limitation of the engine operation to the input P_{batt} is not being considered, as long as the calculated input P_{batt} maintain a charge sustaining condition. With the $P_{batt\_shutoff}$ constraints, P_{batt} points that exceed the $P_{batt\_shutoff}$ limit are lowered to the constraints values, causing the power drawn out from the battery to be smaller than calculated. Note that the calculated P_{batt} ensures charge sustaining and therefore, changing these values will cause a non-charge-sustaining condition. Therefore, the P_{batt} values need to be adjusted to ensure charge-sustaining operation.

2.8.1 P_{batt} Adjustment for Accurate Traffic Prediction

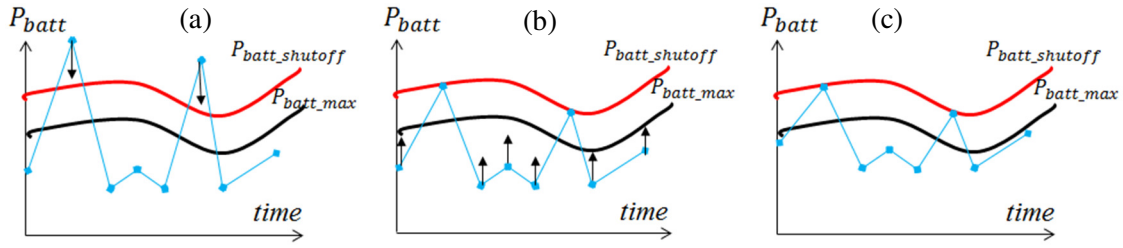


Figure 2.6. P_{batt} Adjustment Procedure

In this work, an accurate prediction case is considered first in order to benchmark the maximum fuel savings potential from the optimization method. Shown in Fig. 2.6 is the P_{batt} adjustment procedure if the calculated P_{batt} are above $P_{batt\_shutoff}$. The blue, red and black lines represent the calculated P_{batt} , $P_{batt\_shutoff}$ and P_{batt_max} respectively. In Fig. 2.6a, calculated P_{batt} points that exceed $P_{batt\_shutoff}$ limit are lowered down to the $P_{batt\_shutoff}$ line. This change in P_{batt} however will result in a non-zero sum-of-$\dot{SOC}$ or excess in battery charge. Therefore, other P_{batt} points that are below the P_{batt_max} line are moved higher to utilize the excess battery charge (Fig. 2.6b – 2.6c) to ensure charge sustaining condition. Note that higher P_{batt} means higher use of battery power and lower $\dot{m}_{fuel}$.

2.8.2 P_{batt} Adjustment for Traffic Prediction with Error

Predicting the behavior of a complex system, such as the speeds of interconnected vehicles will result in prediction errors. However, despite the inevitable presence of the prediction errors, it would be useful to identify how these errors would impact the optimization procedure. First, the $a_0(t)$ across the prediction length between $[t_0, t_f - dt]$ is interpolated from the look-up map using the predicted speeds and accelerations. λ is then calculated from Eq. (2.34) and the input $P_{batt}(t)$ is calculated from λ and the $a_0(t)$ at current time. Therefore, prediction errors will affect $a_0(t)$ interpolation, λ calculation and eventually, $P_{batt}(t)$ calculations.

Note that $a_0(t)$ at current time can be interpolated using actual vehicle speed and acceleration. However, $a_0(t)$ across the prediction length are needed to calculate λ, which introduces errors in P_{batt} calculations. If calculated $P_{batt}(t)$ is outside of the P_{batt} limits, it will be limited to the P_{batt} limit value. However, due to the absence of accurate prediction, the limits can only be approximated using predicted data. Consequently, the P_{batt} limits are inaccurate and in addition, cannot be used to adjust P_{batt} individually (Chapter 2.8.1). Since P_{batt} adjustment cannot be performed, the *SOC* will deviate to a high value, which is undesirable (battery overcharge).

To avoid battery overcharge, the parameter that can be adjusted is λ. Although P_{batt} cannot be individually adjusted due to the lack of accurate P_{batt} upper limit, predicted P_{batt} upper limit can be used to adjust λ to avoid the *SOC* from deviating too high. First,

λ is calculated from the predicted cycle using Eq. (2.34). Using this λ and the predicted cycle, a predicted P_{batt} for $[t_0, t_f - dt]$ is calculated using Eq. (2.28). In addition, the predicted upper limit $P_{batt\_shutoff}$ for $[t_0, t_f - dt]$ is also calculated. Any $P_{batt}(t)$ that exceeds $P_{batt\_shutoff}$ will be limited to the $P_{batt\_shutoff}$ value. Note from Eq. (2.7), P_{batt} is an input to $S\dot{O}C$. Therefore adjusting P_{batt} to $P_{batt\_shutoff}$ at time t will cause a change in $S\dot{O}C$.

$$\Delta S\dot{O}C(t) = S\dot{O}C\big(P_{batt}(t)\big) - S\dot{O}C\left(P_{batt\_shutoff}(t)\right) \tag{2.35}$$

The summation of all $\Delta S\dot{O}C$ due to P_{batt} adjustments at time $t = t_{adj}$, called $\sum_{t_{adj}} \Delta S\dot{O}C$, can therefore be treated as an additional sum-of-$S\dot{O}C(t)$ that need to be compensated to ensure charge sustaining, similar to the derivation of λ in Eq. (2.32)-(2.34). Therefore Eq. (2.34) can be modified as

$$\lambda = 2Q_{batt}^2 R_{batt}\left[\sum_{t_0}^{t_f-dt}\left(\frac{1}{a_0(t)}\right)\right]^{-1}\left[\left(\frac{t_f-t_0}{dt}\right)\left(\frac{V_{oc}}{2Q_{batt}R_{batt}}\right) - \sum_{t_0} S\dot{O}C - \sum_{t_{adj}} \Delta S\dot{O}C\right] \tag{2.36}$$

Using this λ, the process is repeated several times to reduce the amount of adjusted P_{batt} and hence reducing $\sum_{t_{adj}} \Delta S\dot{O}C$. The resulting λ is then used with $a_0(t)$ interpolated from actual drive cycle at current time to calculate the current input $P_{batt}\ (t)$.

2.9 Rule-Based HEV Powertrain Optimization

One way to benchmark the performance of the optimization strategy presented in Chapter 2.6 to Chapter 2.8 is by comparing with the Rule-Based method employed in HEV. In general, there is a two-level controller in a rule-based strategy. The first selects the most efficient engine operating points, and hence the operating points of the electrical machines, by interpolating the iso-power curves of the combined power demands from the wheels and the battery on the engine map [65-67]. The second decides whether or not to implement the commands from the first controller, depending on the vehicle's conditions [63,67,68]. For example, during an all-electric range operation, the second controller will over-ride the first controller decision by turning off the engine if the power demand is low and the *SOC*-level is enough. These methods are also embedded in a vehicle simulation package, ADVISOR 2003, developed by the National Renewable Energy Lab [69]. Simulation results with the UDDS driving cycle is

shown to be similar with simulations by [63]. Details of the Rule-Based method used in this work can be found in Appendix 2.

2.10 Simulation Setup

To test the performance of the proposed traffic prediction and powertrain optimization method, a simulation using microscopic traffic simulator VISSIM is conducted, based on artificially constructed signalized roads. VISSIM is a commercial microscopic traffic simulator, and is built on the Wiedemann's car following model [70]. It has been utilized extensively in various applications, mostly related with the traffic evaluation. We need to point out that, the simulated vehicle trajectories from the VISSIM program could only reflect simplified driving dynamics to a limited extent, and would not be capable to fully capture the complexity of human driving behaviors. Still, the simulation experiment is a much more cost-effective alternative over the real world experiment and is useful to gain insights regarding the performance of the powertrain optimization algorithm. Two cases are constructed with 6 miles and 15 miles roads. The link lengths are randomly selected ranging from 600 ft to 4000 ft. Fixed-time signal controllers are used with a common cycle length along the road. For simplicity, only one lane highway is simulated without turning vehicles at intersections. By doing so, the randomness from lane-changing, as well as the inflow and outflow of traffic are eliminated.

The layouts of both networks are shown in Fig. 2.7. Vehicles are simulated to travel through the networks circularly. The trajectory data of each vehicle during the first simulation run is collected for parameter learning only. The data from the second run with the trained parameters are then used for prediction and powertrain optimization. For comparison purposes, prediction and optimization without parameters learning are performed as well.

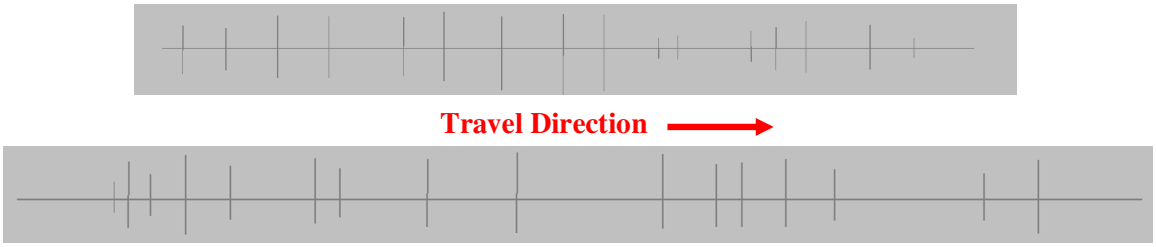


Figure 2.7. Network Layout in VISSIM (not scaled) for 6-mile (top) and 15-mile (bottom) Cases

In order to investigate the effects of optimization frequency, the trajectory prediction for every vehicle is updated every 10, 20, 30 and 40 seconds for both the 6-mile and 15-mile cases. 30 vehicles are selected to test the optimization effectiveness. Two categories of prediction update cases are investigated: accurate-prediction and prediction-with-error. In accurate-prediction cases, driving trajectories from the VISSIM traffic simulator, which represent the actual drive cycles, are used in the optimizations at every prediction cycle. The predictions based on Gipps' and CTM models are called the prediction-with-error, and is further divided into two categories, prediction with and without parameter calibrations. Finally, the prediction in the last 100 seconds is assumed to be accurate to allow a charge-sustaining condition for the prediction-with-error cases to get fair fuel economy comparisons with accurate-prediction and Rule-Based. In the Rule-Based method, the battery will start to charge or discharge to the initial capacity during the 100 seconds before the final time.

2.11 Simulation Results

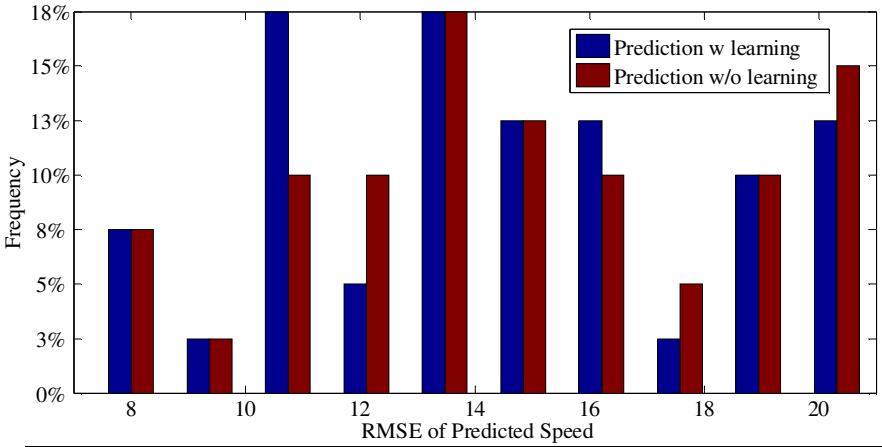
First, the errors from vehicle speed predictions are calculated. Simulation results for all 30 vehicles in the 6-mile and 15-miles cases are then discussed. Next, a single vehicle in the 6-mile case is selected for a more detailed analysis.

2.11.1 Vehicle Speed Prediction Performance

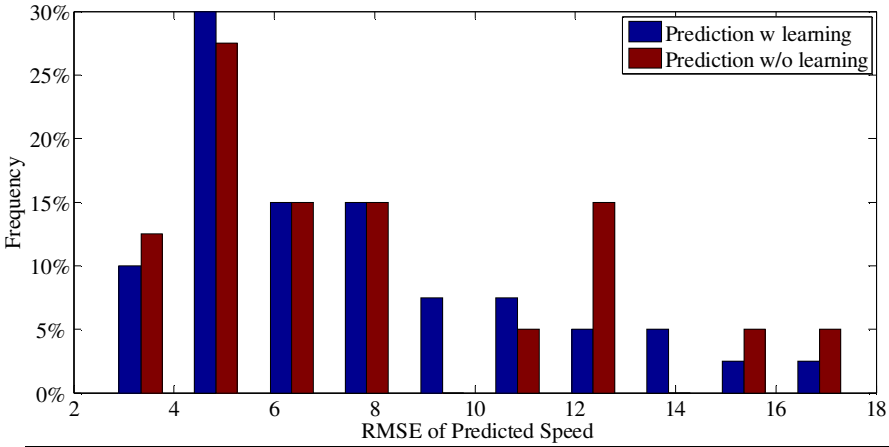
The RMSE of time series of the speed is calculated at every update time to quantify the traffic prediction error. Since the predictions for 20, 30 and 40 seconds update times are sampled in different frequencies from the predictions in the 10 seconds update time, only RMSE of 10 seconds update time is shown here.

For each vehicle, the RMSE along the trip are then averaged, and the results are shown in Fig. 2.8. The blue bars show the RMSE with parameter learning and the red bars without parameter learning. As shown in the figure, the improvements with parameter learning are very modest. The possible explanation is as the following. The prediction error for the first vehicle is from the CTM prediction. The error of this leading vehicle's trajectory then propagates to its followers' prediction and it will be hard to improve by the parameter learning for the followers. The RMSE distribution for the 6-miles case was observed to be larger than the 15-mile case, which is most likely due to the different densities of intersections in the two layouts (Fig. 2.7). In the 6-mile case, the signal

impact is more substantial, with more closely spaced intersections, resulting in more frequent speed changes than in the 15-mile case.



	Prediction w learning	Prediction w/o learning
Average RMSE	14.40	14.78



	Prediction w learning	Prediction w/o learning
Average RMSE	7.54	7.87

Figure 2.8. Average Predicted Vehicle Speed RMSE Comparisons for 6-mile (top) and 15-mile (bottom) Cases

To reveal more details of the prediction error, the predicted and observed speeds of vehicle 12, 6-mile case is highlighted in Fig. 2.9. The RMSE is 19.3 mph for this prediction, which is typical for the 6-mile case. In this case, two main factors contributing to the prediction error can be seen. First, the predicted speed is shifted from the observed speed, indicated by letter A, which could substantially contribute to the RMSE. This indicates delayed maneuver time instances between the prediction and observation, even though the predicted maneuvers, e.g. acceleration or deceleration,

are correct. Secondly, prediction of signal impact could be biased, especially for long term prediction. For instance, as indicated by letter B, the subject vehicle stopped at a red traffic signal, while the prediction indicates the vehicle arrived at a different arrival time, and passed the intersection smoothly. These types of details would be difficult to capture accurately on a microscopic level. However, even with relatively large RMSE, consistent patterns of accelerating, decelerating and cruising were found between the observed and predicted data. It has to be mentioned that the vehicle speed prediction results in this chapter are part of a co-author works in [59], but is presented in this thesis for clarity in the powertrain optimization results in the following chapters.

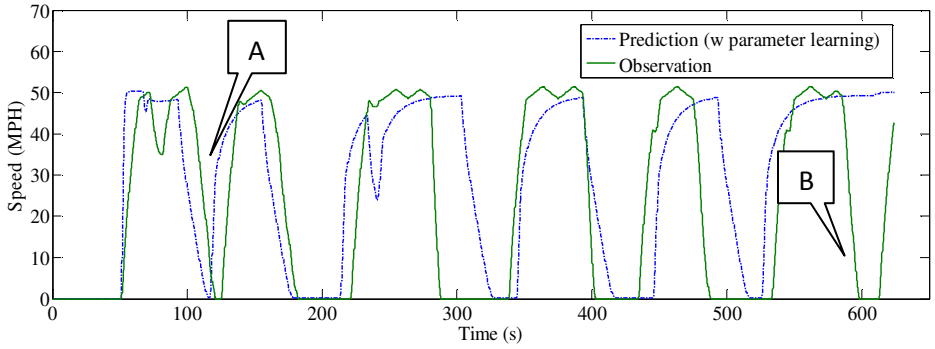


Figure 2.9. Comparison of Predicted and Observed Speeds for Vehicle 12, 6-mile Case

2.11.2 Optimization Performance

6-mile Case: Shown in Fig. 2.10 are the comparisons of average MPG for 30 vehicles for 10, 20, 30 and 40 seconds prediction update times and the Rule-Based method (denoted by RB) for the 6-mile case. Accurate-prediction shows roughly 9.6% improvement while Prediction with error cases show 1.6% to 4.3% improvements.

The MPG's between prediction without (Fig. 2.10a) and with (Fig. 2.10b) parameters calibration are roughly the same. This can be explained by calculating the Mean-Absolute-Percentage-Error for λ (λ–MAPE) between prediction-with-error and accurate-prediction in each vehicle, for every prediction update case. Fairly similar distributions of λ–MAPE, between prediction without and with parameters calibration, is shown for every prediction update case in Fig. 2.11. As explained in Chapter 2.8.2, prediction errors will affect λ calculation, which is used to calculate the input P_{batt}, causing deterioration in the MPG. Therefore, if the λ-error (represented by λ–MAPE) are similar between prediction

without and with parameters calibration, then the MPG's should also be similar. Also, λ–MAPE's are shown to be small, ranging from 0% to 7% (Fig. 2.11).

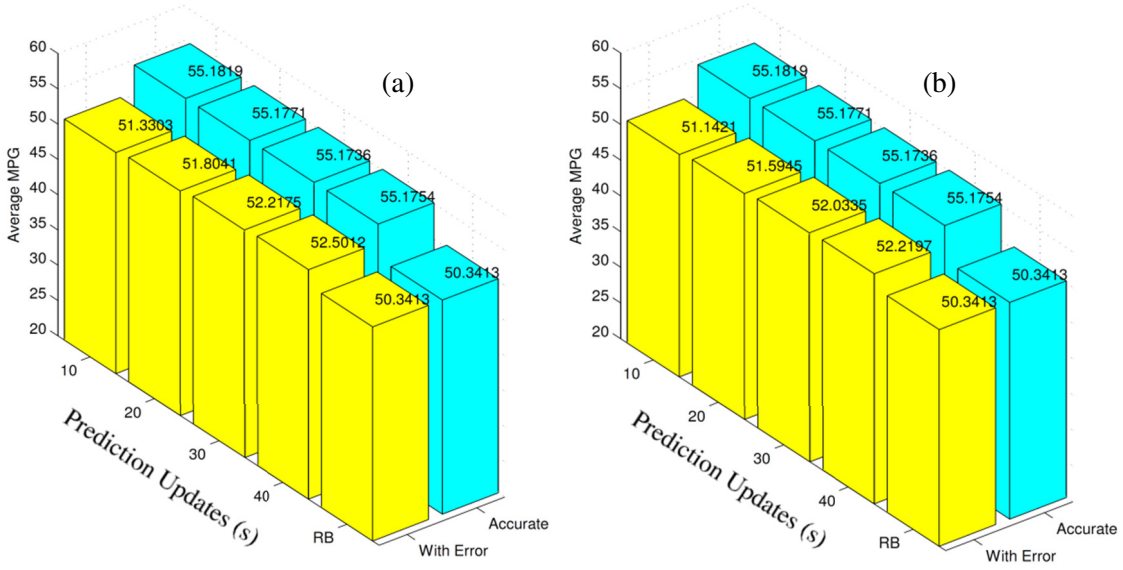


Figure 2.10. Ave. MPG for 6-mile Case without (a) and with (b) Prediction Parameters Calibration

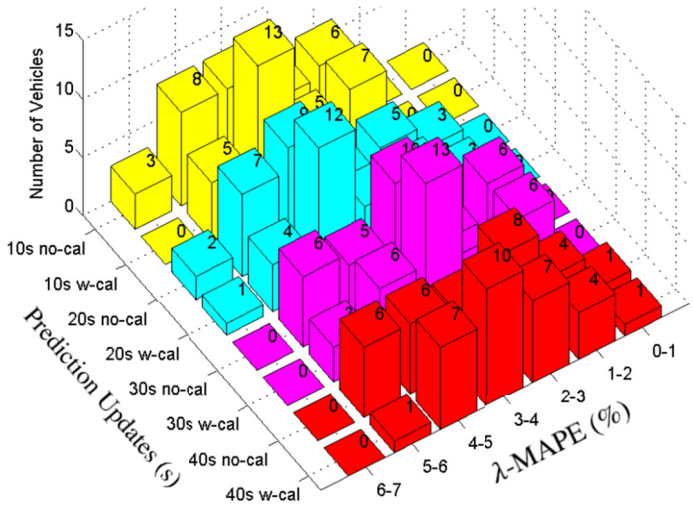


Figure 2.11. λ-MAPE distribution for 6-mile Case without and with Parameters Calibration

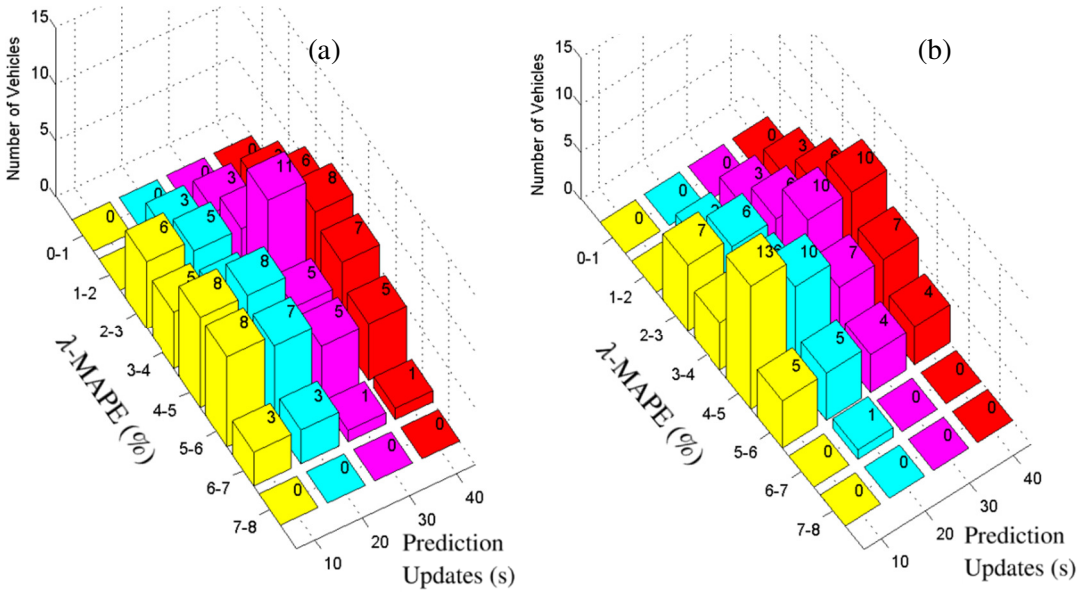


Figure 2.12. λ-MAPE against 10s-update for 6-mile Case without (a) and with (b) Parameters Calibration

Although parameters calibration has no significant effects on the results, the MPG's are slightly higher with longer update time (Fig. 2.10). The reason can be seen when comparing the λ-MAPE distributions for every prediction update case, which is calculated against the 10s accurate-prediction case (Fig. 2.12). The λ-MAPE is calculated relative to the 10s accurate-prediction case in order to compare the λ errors at 20s, 30s and 40s update times with the λ error at 10s update time. The figures show distribution at lower values with longer update time, indicating lower λ errors at longer update time, which is caused by prediction accuracy, as will be discussed next.

Figure 2.13a-2.13c shows the speed predictions at time instances of 50s, 60s and 70s, the actual speed and the initially predicted speed. Figure 2.13d shows λ for 10s-Prediction update (accurate and predicted) and 40s-Prediction update (predicted). The prediction at every 10s between 50s and 70s does not show much improvement when compared to the actual speed, although it captures the general trend (Fig. 2.13a-2.13c). This can happen if the vehicle becomes isolated or no new information is transmitted to improve the prediction at certain time intervals. Prediction error introduces error in λ, as shown in Fig. 2.13d at 50s. However, since the prediction error does not improve at 60s (Fig. 2.13b), λ error at 60s is almost the same as λ error at 50s for 10s-Prediction case (Fig. 2.13d). λ for 40s-Prediction case however does not change between 50s and 70s

since it is only updated every 40s. In this case, λ error for 40s-Prediction case is shown to be lower than 10s-Prediction within the time interval. Therefore, if the prediction error does not improve at every update time, there will be a steady-state λ error with respect to the accurate-prediction case, causing higher overall λ error, which will impact the MPG. However, λ error can be reduced if the speed prediction becomes more accurate at every update time.

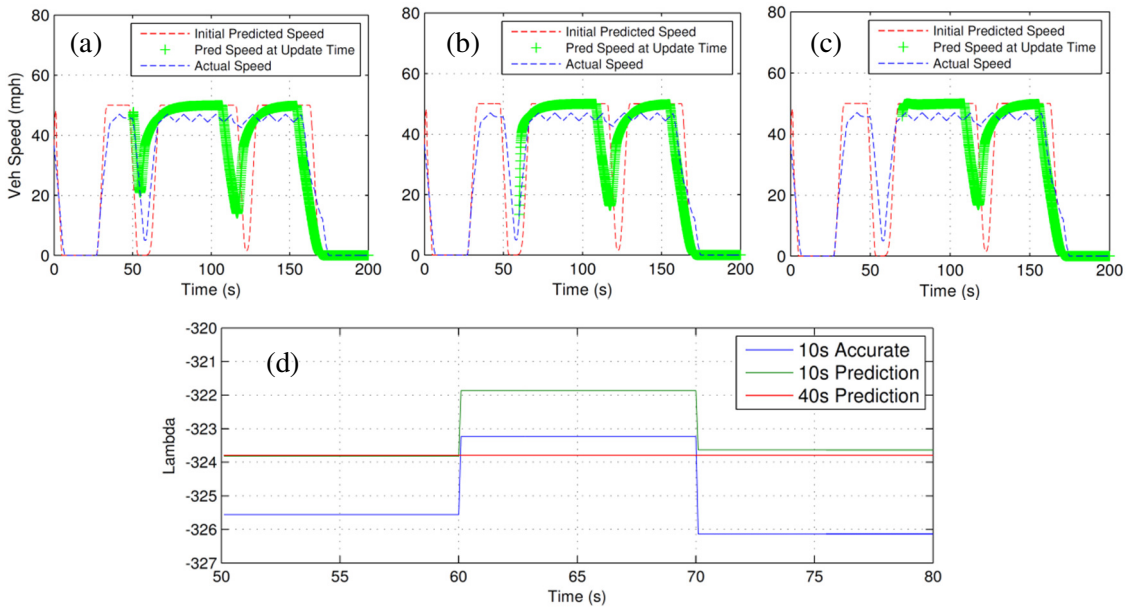


Figure 2.13. λ Comparisons between 10s-Accurate, 10s-Prediction and 40s-Prediction

15-mile Case: Shown in Fig. 2.14 are the comparisons of average MPG for 30 vehicles for 10, 20, 30 and 40 seconds prediction update times and the Rule-Based method for the 15-mile case. Accurate-prediction shows roughly 7% improvement while prediction with error shows 2.6% to 3.5% improvements. Similar behaviors found in the 6-mile case are also found in the 15-mile case, as shown by the λ-MAPE distributions in Fig. 2.15 and 2.16.

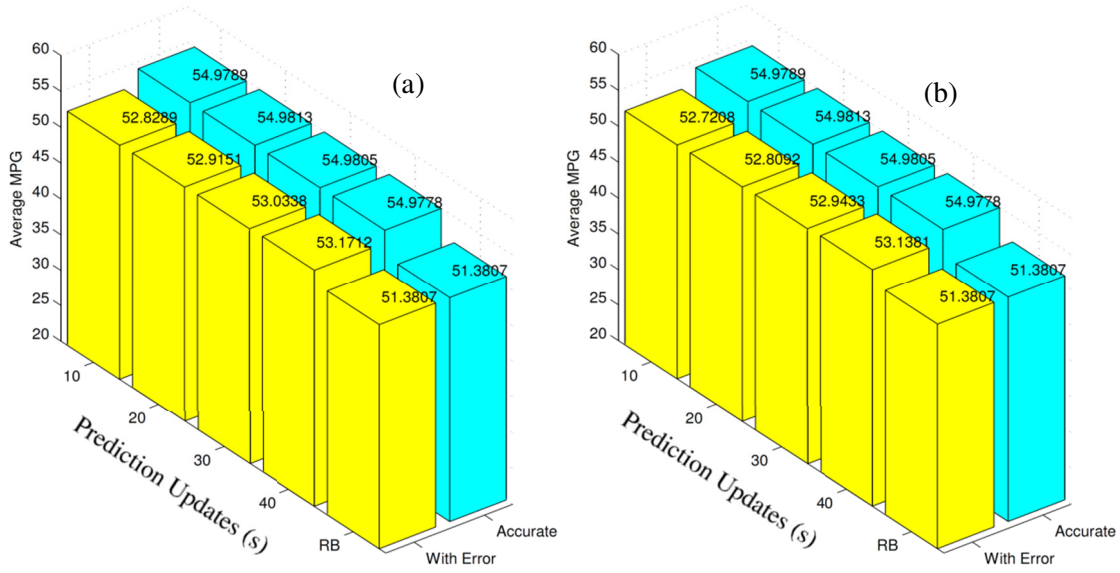


Figure 2.14. Ave. MPG for 15-mile Case without (a) and with (b) Prediction Parameters Calibration

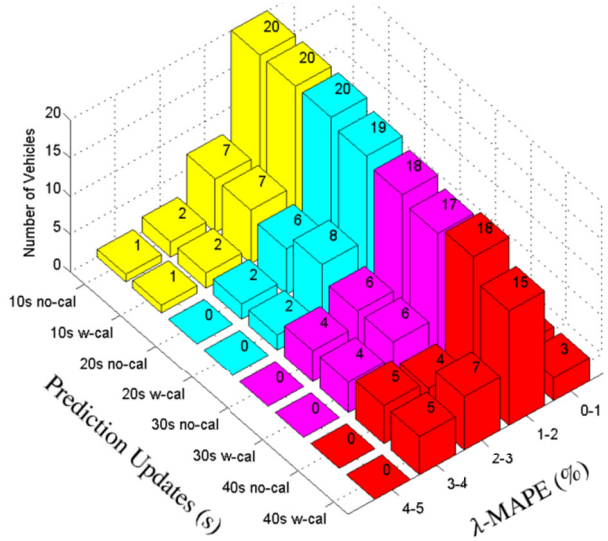


Figure 2.15. λ-MAPE distribution for 15-mile Case without and with Parameters Calibration

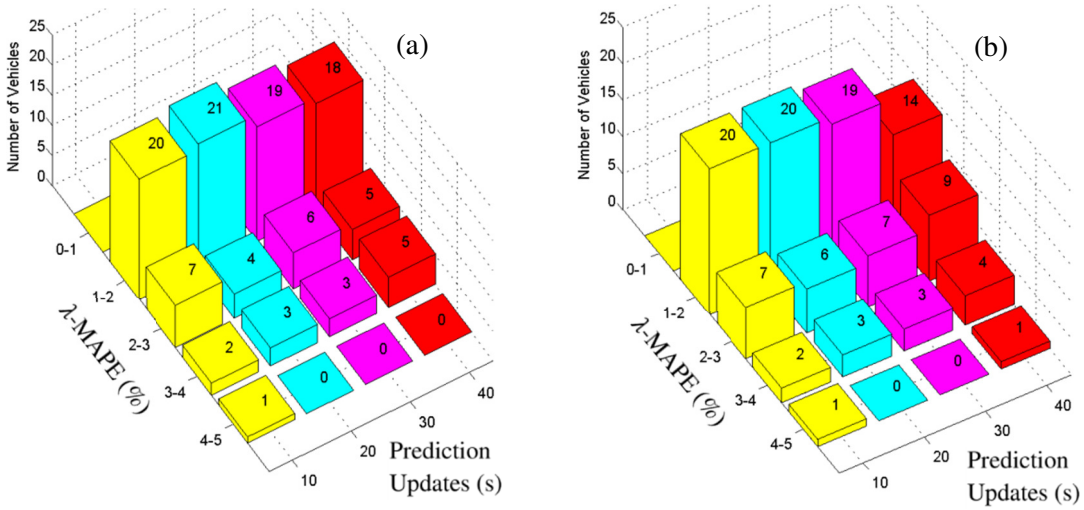


Figure 2.16. λ-MAPE against 10s-update for 15-mile Case without (a) and with (b) Parameters Calibration

2.11.3 Case Study: Single Vehicle Simulation

To further analyze the fuel savings, the dynamics of vehicle 28 with 10s update (6-mile case) is selected. Accurate-prediction (AP), Prediction-with-Error (PE) and Rule-Based (RB) achieved 47.7552 MPG, 47.6164 MPG and 42.4569 MPG respectively. Figure 2.17-2.19 show the vehicle speed, battery SOC, cumulative fuel consumption and engine torque-speed for all three cases.

Shown in Fig. 2.17, the SOC for PE case follows the AP closely, while the RB shows more battery use with lower SOC level. This is also reflected in the cumulative fuel consumption plot, where the rule-based consumed less fuel, but at the expense of lower battery SOC. Note that at the beginning of the simulation, with high vehicle speed requests, RB depletes the battery. In contrary, the AP and PE cases both charge the battery slightly in order to ensure charge-sustaining operation. The same pattern can also be seen between 100s and 140s. These actions allow the RB cumulative fuel consumption to be lower before the end of the cycle. At around 380s however, the SOC for RB becomes too low at 50%, which then triggers a battery-charging action that saw the SOC increased to 55%, and can be seen with higher engine torque and speed (Fig. 2.18–2.19). Around 650s, the RB is forced to charge the battery back to the initial SOC. Unfortunately at this point, there is a large acceleration, which therefore forces the RB to operate inefficiently to charge the battery and meet the high power demand. It can be

seen that the RB cumulative fuel starts to increase at a fast rate and becomes larger than the AP and PE cases around 670s. The AP and PE cases however maintained the SOC level around 57% before utilizing the future deceleration to charge the SOC back to 60% at final time.

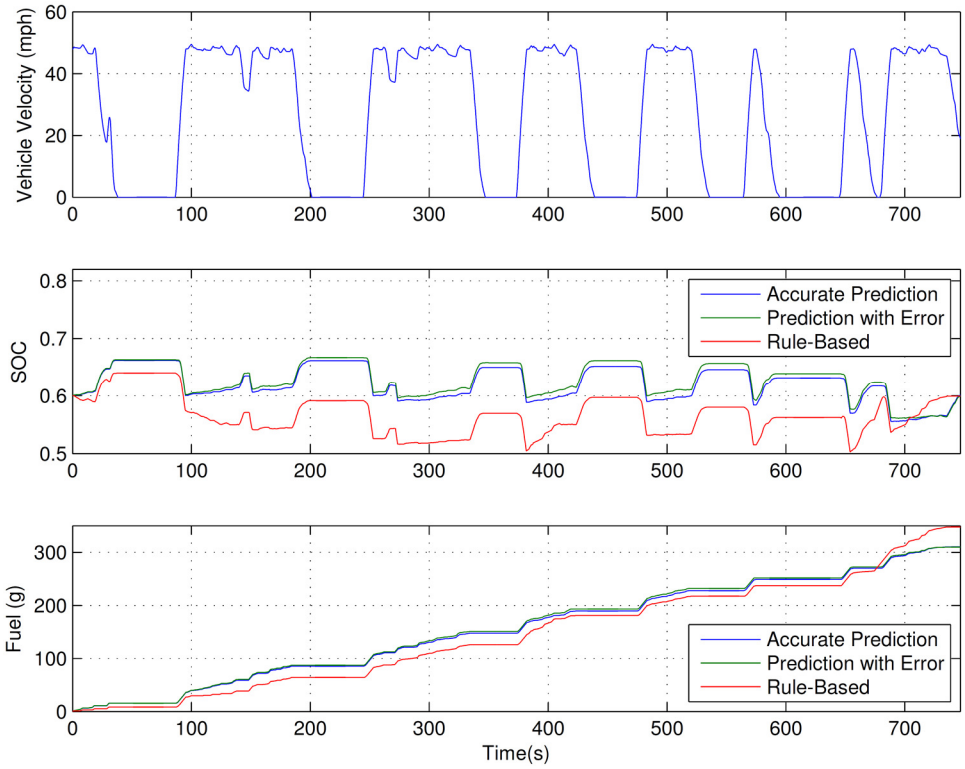


Figure 2.17. Vehicle Speed, Battery SOC and Cumulative Fuel Consumption

It can be seen throughout the cycle that the battery assisted the engine at high power demands and charges back the battery during decelerations. However, the amount of electric power used at a particular time to assist the engine plays a vital role to ensure an overall good fuel economy performance. Knowing the future information allows the AP and PE cases to make good judgment to avoid discharging the battery too much in the middle of the cycle and therefore allowing it to avoid the high fuel costs at the end of the cycle. It can also be seen that RB spends a significant amount of fuel towards the last 100 seconds in order to maintain a charge-sustaining condition. Since the ratio of the charge-sustaining interval over the total travel time for 6-mile case is bigger than 15-mile case, this could explain why the average MPG for RB is worse for 6-mile case than 15-mile case (Fig. 2.10 and Fig. 2.14).

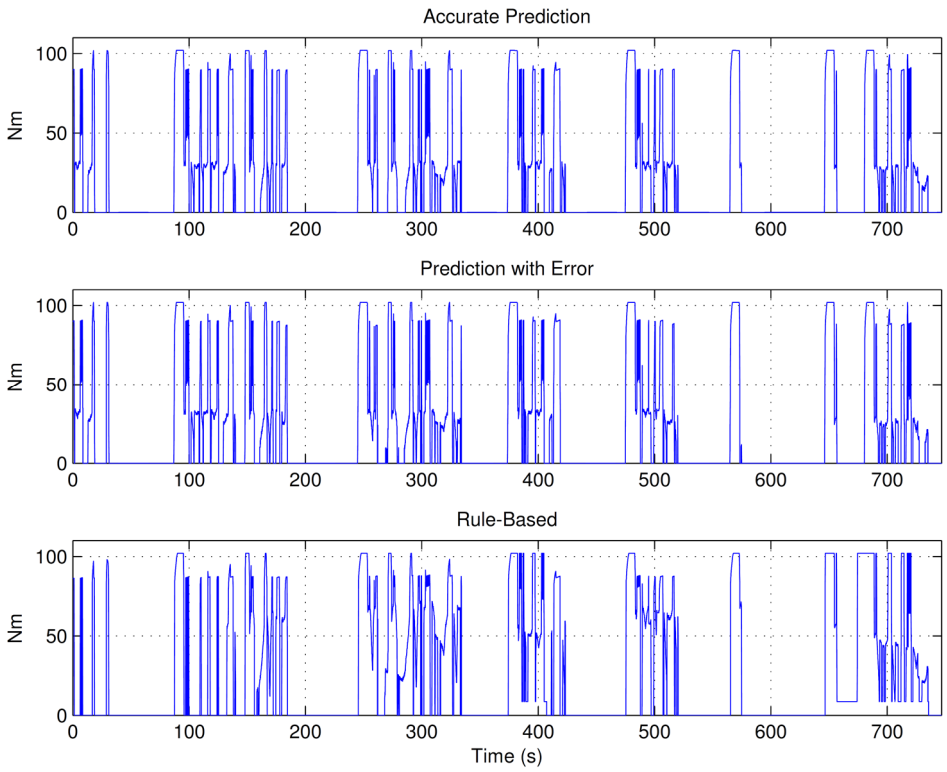


Figure 2.18. Engine Torque for Accurate, Predicted and Rule-Based Cases

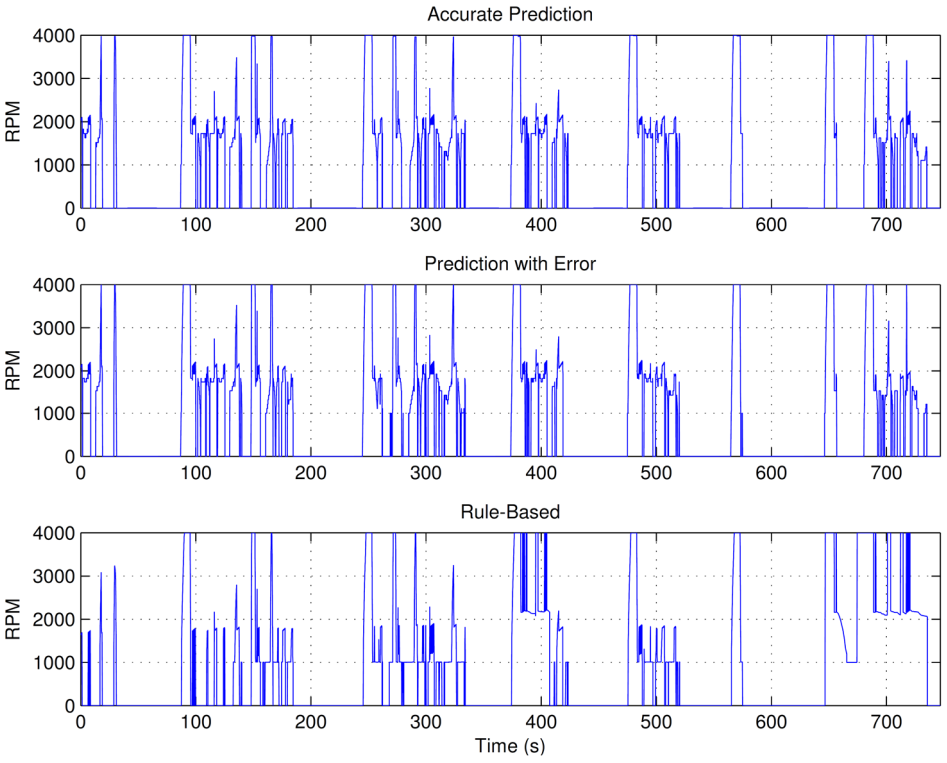


Figure 2.19. Engine Speed for Accurate, Predicted and Rule-Based Cases

2.12 Conclusion

In this work, a real-time hybrid-electrical-vehicle optimization method utilizing a traffic prediction model that predicts the vehicle trajectory is proposed. The prediction is done by assuming availability of inter vehicle communication (IVC) and vehicle infrastructure integration (VII) information. The prediction, optimization and update structure is simulated and several conclusions can be made. First, the traffic model parameter calibration shows minimal improvements to prediction accuracy due to the error from the leading vehicle and has little effects on the MPG of the vehicles. Secondly, results show that a frequent update time does not necessarily means better MPG performance. If the vehicle trajectory prediction error at an update time does not improve from previous update time, the λ-error will be inherited, affecting MPG performance. This however is a limitation of the IVC and VII network, where the vehicle interconnection with the whole prediction network depends on the vehicle's location. There will be certain durations where the information cannot be sent to the vehicle, causing the vehicle's prediction to be unchanging. Therefore, it would better to know when to resume or stop updating λ when a new prediction is received, depending on the prediction's accuracy. Thirdly, accurate prediction records an averaged 7% to 9.6% MPG's improvements over Rule-Based method, which could be significant when translated to a network level savings [71], but is modest for a single vehicle. The reason is because an unconstrained optimization method is used on a system with physical limitations. Therefore, manually constraining the input may not be the best solution and a better result could be gained using a constrained optimization derivation. Finally, the impact of prediction errors to the optimization comes from λ calculation. Therefore, a more robust method to calculate λ is needed in the presence of these errors.

Chapter 3

SP-Based Constrained HEV Powertrain Optimization

3.1 Motivation

Connected Vehicle (CV) technology allows traffic information sharing between vehicles. With Inter-Vehicle-communication (IVC) and Vehicle-Infrastructure-Integration (VII), vehicle loads can be better predicted across a future time window and can be utilized in vehicle powertrain optimization to save fuel. This is especially true for Hybrid Electrical Vehicle (HEV) where the powertrain is equipped with an additional power storage device, namely a battery. The battery could assist, store or replace the engine power to avoid the engine from operating at inefficient operation regions. Particularly, a HEV with power-split powertrain architecture employs a continuous-variable-transmission (CVT), which gives greater flexibility in selecting any desired engine operating points for any vehicle load. However, considering the physical constraints and nonlinearity of complex systems such as the internal combustion engine and the battery, optimizing the HEV powertrain using traditional optimization methods are computationally heavy. Furthermore, traffic environment is transient, where traffic states change all the time, affecting the vehicle load predictions from IVC and VII. In light of this, there is a need for a fast HEV powertrain optimization that can be implemented with repeated speed prediction updates due to the transient nature of traffic. For practical implementation, the engine operating range and the engine transient dynamics also have to be considered. Previous researches in this area have not cohesively address the optimality, engine constraints, speed of calculation and charge-sustaining (CS) conditions. In this chapter, a fast HEV powertrain optimization that considers the associated constraints and engine transient dynamics is proposed for practical implementation.

3.2 Power-Split HEV Model

The powertrain dynamics in this work utilizes the Toyota Hybrid System (THS) power-split architecture [5] (Fig. 3.1). The powertrain uses a planetary gear set to connect the engine, and two electrical machines, both can either be a motor or generator, to allow two degrees-of-freedom in meeting the speed and torque demands. Static equations relating the torques and speeds of the engine and the two electrical machines are given by Eq. (3.1–3.5). Notations *m* and *g* are used to differentiate between the electrical machines. *T* is the torques, ω is the rotational speeds (notations *e*, *g* and *m* denote the engine, generator and motor) while *S* is the sun-gear teeth, *R* is the ring gear teeth and *K* is the final gear ratio between the motor and the driveline. T_v in Eq. (3.5) is the requested vehicle torque calculated from the rolling resistance, aerodynamic drag and vehicle acceleration respectively, while ω_v is the vehicle speed.

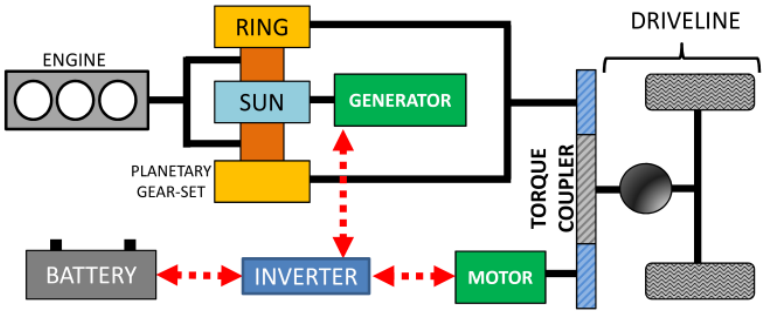


Figure 3.1. Toyota Hybrid System Power-Split Architecture

$$\omega_e(R+S) = \omega_g S + \omega_m R \quad (3.1)$$

$$\omega_m = K\omega_v \quad (3.2)$$

$$T_m = \frac{T_v}{K} - T_e\left(\frac{R}{S+R}\right) \quad (3.3)$$

$$T_g = -T_e\left(\frac{S}{S+R}\right) \quad (3.4)$$

$$T_v = R_{tire}\cdot(f_{tire}M_v g + 0.5\rho_a C_{drag} A_f \omega_v^2 R_{tire}^2 + M_v \alpha_v R_{tire}) \quad (3.5)$$

Both electrical machines are connected to a battery through an inverter, where the net-power-to-battery is a function of the speeds and torques of both electrical machines.

$$P_{batt} = \eta_m^{km}\omega_m T_m + \eta_g^{kg}\omega_g T_g \quad (3.6)$$

η_m and η_g are the efficiencies of the electrical machines (assumed constant). *km* and *kg* value can either be -1 or +1, depending on the operation of the electrical machines (-1 if motor or +1 if generator). An electrical machine is either in motoring or generating mode if the product of torque and speed is either negative or positive respectively. The dynamics of the battery State-of-Charge (*SOC*) is a function of P_{batt} [68].

$$\dot{SOC} = -\frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_{batt}P_{batt}}}{2R_{batt}Q_{batt}} \tag{3.7}$$

3.3 Optimization Problem and Framework

Given a time window [t_1, t_N], fuel consumption cost can be written as

$$J = \sum_{t_1}^{t_N} \dot{m}_{fuel} dt \tag{3.8}$$

where $\dot{m}_{fuel}$ is the mass fuel rate which depends on the engine operating points (T_e, ω_e) while *dt* is the time step. With HEV, fuel optimization is achieved by optimizing the power split between the engine and the battery for a given vehicle load. For a selected battery power, the engine operating point can be optimized to achieve minimum fuel consumption while meeting the vehicle load requirement. This correlation between minimum fuel consumption, $\dot{m}_{fuel}$ and input to the battery, P_{batt} for any given vehicle load is referred to as the Confined-Optimal Operating Line in [23] which will be discussed briefly.

$$\begin{bmatrix} T_g \\ T_m \\ T_e \end{bmatrix} = \begin{bmatrix} \eta_g^{kg}\omega_g & \eta_m^{km}\omega_m & 0 \\ 0 & 1 & \left(\frac{R}{S+R}\right) \\ 1 & 0 & \left(\frac{S}{S+R}\right) \end{bmatrix}^{-1} \begin{bmatrix} P_{batt} \\ \left(\frac{T_v}{K}\right) \\ 0 \end{bmatrix} \tag{3.9}$$

Given ω_v and α_v, ω_m and T_v can be calculated in Eq. (3.2,3.5). P_{batt} candidates can be iterated by first setting the (T_e, ω_e) at the extremes of an engine map (from highest to lowest fuel consumption). Then with Eq. (3.1–3.5) the motor/generator torques and speeds can be calculated, hence the P_{batt} limits. P_{batt} values are then iterated between

the P_{batt} limits. Given a P_{batt} value, Eq. (3.3–3.6) are combined to simultaneously solve the torques in Eq. (3.9).

Note that ω_e is iterated between the engine map speed limits, while possible combinations of k_g and k_m, are also iterated to find the working operations of the electrical machines. Therefore, in this sense, for a given P_{batt} value, ω_e can be iterated to find the pair (T_e, ω_e) with the minimum fuel rate $\dot{m}_{fuel}$. The process is repeated for all P_{batt} candidates between the P_{batt} limits. The resulting minimum $\dot{m}_{fuel}$ vs P_{batt} plot for a given vehicle load is shown in Fig. 3.2. In essence, for a specific vehicle power demand at a time instance, the minimum fuel consumption $\dot{m}_{fuel}$ for different levels of battery charging and discharging in a power-split HEV given by P_{batt} is represented by this fixed nonlinear relationship. This relationship can be calculated offline and stored as a lookup table for all possible vehicle power demand values.

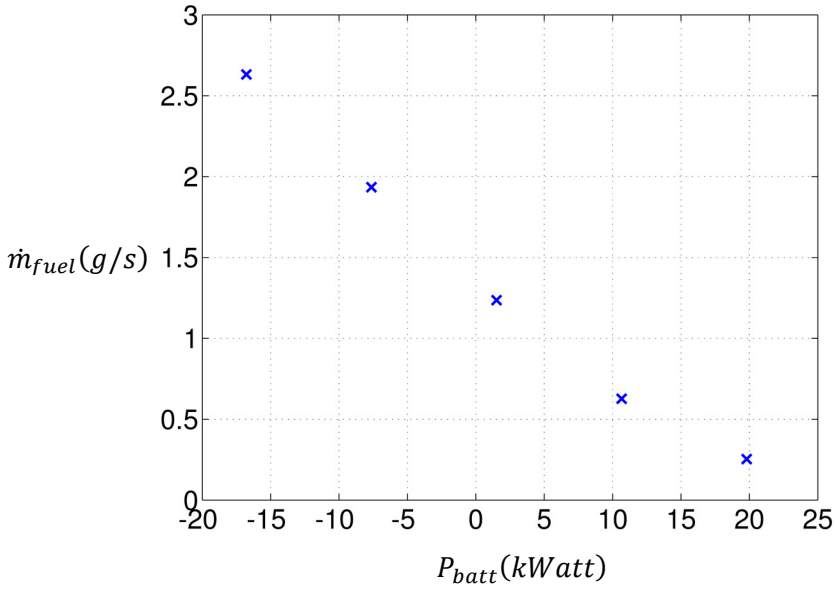


Figure 3.2. Minimum $\dot{m}_{fuel}$ VS P_{batt}

Therefore, P_{batt} in Eq. (3.10) is the input that can be chosen between the known constraints P_{batt_lo} and P_{batt_hi} at every time instance t_i to minimize the cumulative fuel consumption cost represented in Eq. (3.11).

$$P_{batt}(t_i) = \left[P_{batt\_lo}(t_i), P_{batt\_hi}(t_i) \right] \quad \forall t_i : [t_1, t_N] \tag{3.10}$$

$$J = \sum_{t_1}^{t_N} \dot{m}_{fuel}\left(P_{batt}\left(t_i\right)\right)dt \tag{3.11}$$

Intuitively, to minimize fuel use, the HEV can operate in all-electric mode until the battery is depleted. However, since the recommended battery operating range is limited [72], and for fair comparisons between different optimization strategies, the battery charge sustaining (CS) condition is introduced to ensure the battery *SOC* is maintained, $SOC(t_N) = SOC(t_1)$.

$$\sum_{t_1}^{t_{N-1}} \dot{SOC}\left(P_{batt}\left(t_i\right)\right) = 0 \tag{3.12}$$

The battery *SOC* dynamics is given in Eq. (3.7). The CS condition represents the constraint in the optimization problem. Therefore, the nonlinear cost function is given by Eq. (3.10), the input constraints are given by Eq. (3.11), a nonlinear constraint as a function of the input is given by Eq. (3.12) and the state dynamics is given by Eq. (3.7).

3.4 Input Rate Constraint

The optimization problem defined so far will suffice in simulations. However, in order to implement the optimized engine operating points in actual engine, the engine speed and engine torque cannot change too fast between two time steps. Since there is direct correlation between input P_{batt} and engine operating point, P_{batt} jump between two time steps is therefore constrained.

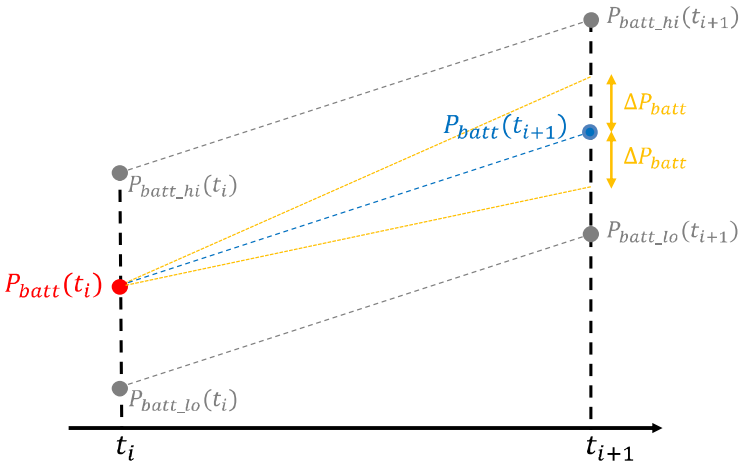


Figure 3.3. P_{batt} Jump Constraint

Figure 3.3 shows the maximum P_{batt}-jump ΔP_{batt} between two time steps. As discussed before, at every time step P_{batt} is iterated between P_{batt_lo} and P_{batt_hi}, which are determined based on the highest and lowest engine operating points. Therefore, the relative position of $P_{batt}(t_i)$ between $P_{batt\_lo}(t_i)$ and $P_{batt\_hi}(t_i)$ at time step (t_i) and the relative position of $P_{batt}(t_{i+1})$ between $P_{batt\_lo}(t_{i+1})$ and $P_{batt\_hi}(t_{i+1})$ at time step (t_{i+1}) can be directly related to the engine operating points at the two time instances. Restricting the P_{batt}-jump between these two relative P_{batt} positions, as shown in Fig. 3.3, will restrict the engine transient operation. This constraint is expressed as

$$\left|P_{batt\_rel}(t_{i+1}) - P_{batt}(t_i)\right| \le \Delta P_{batt} \quad \forall t_i : [t_1, t_{N-1}] \tag{3.13}$$

where $P_{batt\_rel}(t_{i+1})$ is the $P_{batt}(t_{i+1})$ between $P_{batt\_lo}(t_{i+1})$ and $P_{batt\_hi}(t_{i+1})$ relative to $P_{batt}(t_i)$ between $P_{batt\_lo}(t_i)$ and $P_{batt\_hi}(t_i)$

$$P_{batt\_rel}(t_{i+1}) = \beta\left(P_{batt}(t_{i+1}) - P_{batt\_lo}(t_{i+1})\right) + P_{batt\_lo}(t_i)$$

$$\text{where} \quad \beta = \frac{P_{batt\_hi}(t_i) - P_{batt\_lo}(t_i)}{P_{batt\_hi}(t_{i+1}) - P_{batt\_lo}(t_{i+1})}$$

The inequality in Eq. (3.13) can be equally written as 2 sets of (N-1) linear constraints

$$P_{batt\_rel}(t_{i+1}) - P_{batt}(t_i) \le \Delta P_{batt} \qquad \forall t_i : [t_1, t_{N-1}] \tag{3.14}$$

$$-P_{batt\_rel}(t_{i+1}) + P_{batt}(t_i) \le \Delta P_{batt} \qquad \forall t_i : [t_1, t_{N-1}] \tag{3.15}$$

Note in Eq. (3.14, 3.15), if the constraint is enforced by one of the inequalities, the other inequality will be complimentary for all positive and negative values of *Pbatt*. For the specific engine used in this work, a ΔP_{batt} value of 0.5 kWatts is found to be a good value to ensure feasible engine dynamics.

Overall, the HEV powertrain optimization problem is solved in two steps. First, given the flexibility of the power-split HEV powertrain, the relationship between the minimum fuel consumption and power-split levels defined by P_{batt} is calculated and stored offline for all possible vehicle power demands. This relationship ensures the engine is operating at the most optimal point for every possible HEV power-split level for a given vehicle power demand. Secondly, by utilizing the separable structure found in the reformulated

problem, Separable Programming (SP) is used as a fast and effective optimization solution to the problem, as will be discussed next.

3.5 Separable Programming Implementation

First we consider the optimization problem defined by Eq. (3.10-3.15). Note the cost function in Eq. (3.11), the equality constraint in Eq. (3.12) and inequality constraints in Eq. (3.14, 3.15) are additive over time where input P_{batt} is the sole decision variable. At each additive term in the cost and constraints functions, input P_{batt} for a specific time instance t_i appears alone and uncoupled with input P_{batt} from other time instances, which is common for many dynamical system. Since the cost and constraints functions are additive over time and input P_{batt} appears separately in each additive term, the problem therefore has a separable structure and suitable for Separable Programming implementation.

In SP, the nonlinear cost function and state dynamics are approximated with piecewise linear functions with the introduction of new variables λ's. The λ-method uses actual data points for piecewise linear functions fitting [73]. In the example below, sampling points of M = 5 is used for better visualization, but M = 10 and 20 are used in the optimization. Note however the there is a tradeoff when choosing the number of sampling points. The cost and state dynamics are better represented with more sampling points, but will increase the dimension of the piecewise-linear problem.

Figure 3.4 shows the minimum fuel rate interpolated from the engine map as $P_{batt}(t_i)$ is sampled from $P_{batt\_lo}(t_i)$ to $P_{batt\ \_hi}(t_i)$ using the procedures in Chapter 3.3. Note that the shape of the cost function is convex, which guarantees the adjacency criterion for cost minimization that requires the actual nonlinear cost to be lower than the approximated piecewise-linear cost between the sample points. The battery *SOC* dynamics is calculated for each of the $P_{batt}(t_i)$ sample point and plotted as shown in Fig. 3.5. The sampling values are stored as a lookup table to provide the cost and state-dynamic relationship with the input for a given vehicle load at a specific time.

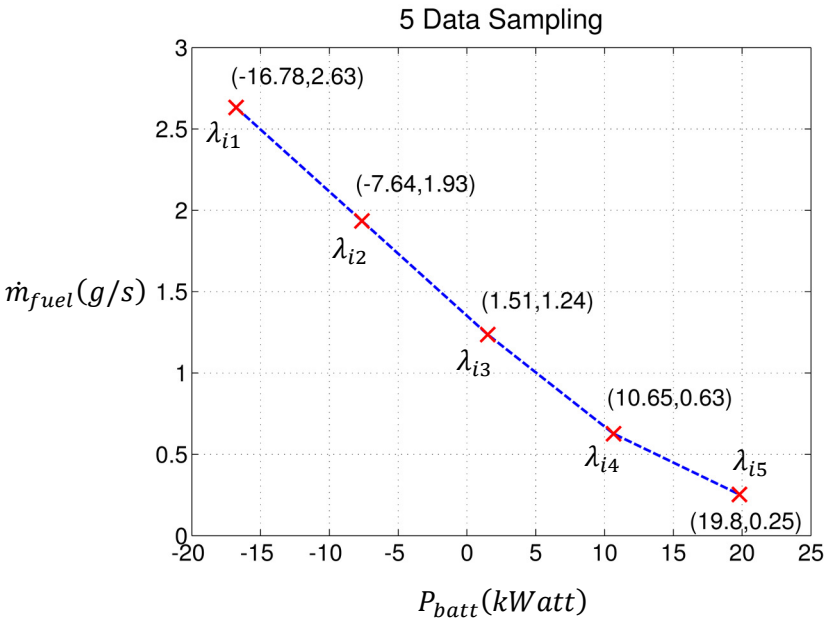


Figure 3.4. Linear Piecewise Approximation of Cost

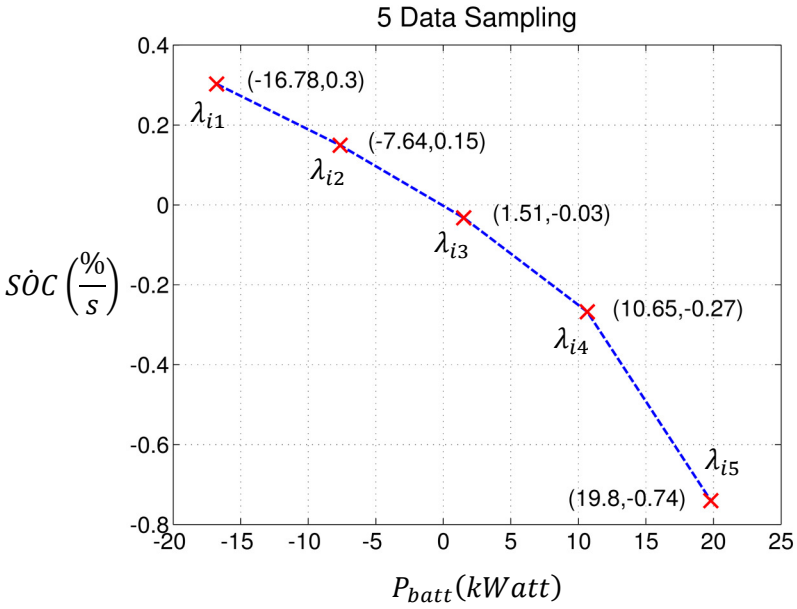


Figure 3.5. Linear Piecewise Approximation of $S\dot{O}C$

The cost, state-dynamic and input are then represented as linear functions of λ_{ij} at time step (t_i) in Eq. (3.16-3.18). The sampling values from Fig (3.4-3.5) are replaced as ($\dot{m}_{ij}$, $S\dot{O}C_{ij}$, P_{batt_ij}) in Eq. (16-18). Equation (3.19) represents the constraints of the linearizing variables [73]. Time instances are represented by subscript *i* while sampling

points are represented by subscript *j*. Vehicle load is incorporated at every time step from w_v and T_v in Eq. (3.2, 3.5) when deriving the minimum $\dot{m}_{fuel}$ vs P_{batt} plot.

$$\dot{m}_{fuel}\left(P_{batt}\left(t_i\right)\right) = \sum_{j=1}^{M=5} \dot{m}_{ij}\lambda_{ij} \tag{3.16}$$

$$S\dot{O}C\left(P_{batt}\left(t_i\right)\right) = \sum_{j=1}^{M=5} S\dot{O}C_{ij}\lambda_{ij} \tag{3.17}$$

$$P_{batt}\left(t_i\right) = \sum_{j=1}^{M} P_{batt\_ij}\lambda_{ij} \tag{3.18}$$

$$\sum_{j=1}^{M=5} \lambda_{ij} = 1 \text{ where } \lambda_{ij} > 0 \tag{3.19}$$

From the introduction of new variables λ_{ij}, the problem in Eq. (3.10-3.15) can be rewritten as a linear problem.

$$J = \sum_{t_1}^{t_N} \dot{m}_{fuel}\left(P_{batt}\left(t_i\right)\right)dt = \sum_{i=1}^{N}\sum_{j=1}^{M} \dot{m}_{ij}\lambda_{ij}dt \tag{3.20}$$

$$\sum_{t_1}^{t_{N-1}} S\dot{O}C\left(P_{batt}\left(t_i\right)\right) = \sum_{i=1}^{N-1}\sum_{j=1}^{M} S\dot{O}C_{ij}\lambda_{ij} = 0 \tag{3.21}$$

$$P_{batt}\left(t_i\right) = \sum_{j=1}^{M} P_{batt\_ij}\lambda_{ij} \quad \forall t_i : \left[t_1, t_N\right] \tag{3.22}$$

$$\sum_{j=1}^{M} \lambda_{ij} = 1 \text{ where } \lambda_{ij} > 0 \qquad \forall t_i : \left[t_1, t_N\right] \tag{3.23}$$

The linear inequality constraints for $\forall t_i : \left[t_1, t_N\right]$ is given by

$$P_{batt\_low}(t_i) \le \sum_{j=1}^{M} P_{batt\_ij}\lambda_{ij} \le P_{batt\_hi}(t_i) \tag{3.24}$$

The linear inequality constraints due to input rate constraint for $\forall t_i : \left[t_1, t_{N-1}\right]$ are expressed as

$$P_{batt\_rel}(t_{i+1}) - \sum_{j=1}^{M} P_{batt\_ij}\lambda_{ij} \le \Delta P_{batt} \tag{3.25}$$

$$-P_{batt\_rel}(t_{i+1}) + \sum_{j=1}^{M} P_{batt\_ij}\lambda_{ij} \le \Delta P_{batt} \tag{3.26}$$

$P_{batt\_rel}(t_{i+1})$ is given by

$$P_{batt\_rel}(t_{i+1}) = \beta\left(\partial P_{batt}(t_{i+1})\right) + P_{batt\_lo}(t_i)$$

$$\text{where} \quad \beta = \frac{P_{batt\_hi}(t_i) - P_{batt\_lo}(t_i)}{P_{batt\_hi}(t_{i+1}) - P_{batt\_lo}(t_{i+1})}$$

$$\partial P_{batt}(t_{i+1}) = \sum_{j=1}^{M} P_{batt(i+1)j}\lambda_{(i+1)j} - P_{batt\_lo}(t_{i+1})$$

i and *N* represent the time index and total time, while *j* and *M* represent the sampling index and total sample. Equations (3.20-3.23) are the cost function, battery charge sustaining constraint, input representation at every time step t_i, and the linearizing variables constraints. ($\dot{m}_{ij}$, $S\dot{O}C_{ij}$, P_{batt_ij}) values can be obtained offline based on Chapter 3.3 and stored as a lookup table for any given vehicle power demand. Equations (3.24) and Eq. (3.25, 3.26) are the input constraint and the input-rate constraints at every time instance t_i. In this linear form, the problem can be solved with fast and efficient Linear Programming methods, such as the interior-point method.

3.6 Comparisons with Other Optimization Methods

Dynamic Programming, PMP-Based and Rule-Based HEV Powertrain Optimizations are implemented as benchmarks to assess the performance of the proposed optimization strategy.

3.6.1 Dynamic Programming HEV Powertrain Optimization

Dynamic Programming is an offline exhaustive search method to represent the maximum achievable MPG to measure the performance of the proposed optimization method. It involves iterating all possible inputs for all state candidates at every time step, starting from the final time to initial time, and stores the input "path" with minimum cost-

to-go from all time steps to the final time, to guarantee a global optimal solution [68]. The cost, state and input are given by the total fuel consumption in Eq. (3.11), battery *SOC* in Eq. (3.7) and P_{batt} in Eq. (3.6, 3.10). In order to enforce battery charge-sustaining, all possible $S\dot{O}C$ are iterated to calculate input P_{batt} candidates using Eq. (3.7). Minimum $\dot{m}_{fuel}$ for an input P_{batt} can be found following the procedures in Chapter 3.3, but it involves extremely heavy computation when implemented in DP. Therefore, to reduce computation time, a lookup table that correlates the minimum $\dot{m}_{fuel}$ and P_{batt} for any given vehicle load is calculated offline and is defined by 20 P_{batt} sampling points between P_{batt_lo} and P_{batt_hi}. Minimim $\dot{m}_{fuel}$ cost can then be linearly interpolated from this table using the P_{batt} candidates in DP. Considering that SP uses only 10 sampling points in this work, DP has better cost estimate with more sampling points and therefore can be used as the baseline performance measurement.

3.6.2 PMP-Based HEV Powertrain Optimization

The proposed method is also compared with online optimization based on the analytical solution of Pontryagin's Minimum Principles (PMP) [59]. In PMP, in order to derive an analytical solution, an unconstrained input case is first assumed. The input P_{batt} is then manually constrained, unlike Linear Programming where the input constraints can be explicitly incorporated in the solver. The main results are summarized here for reference and readers are encouraged to refer to Chapter 2 for more detailed derivations. The Hamiltonian of the optimization problem is defined as

$$H = \dot{m}_{fuel}\left(P_{batt}(t)\right) + \lambda S\dot{O}C\left(P_{batt}(t)\right) \tag{3.27}$$

In the first term, correlation between minimum $\dot{m}_{fuel}$ and $P_{batt}(t)$ for a vehicle load demand is approximated by a linear slope $a_0(t)$ using 20 sampling points. $a_0(t)$ for different vehicle loads are pre-calculated as a lookup table and is used when predicted vehicle loads are given in the simulation. The co-state λ is defined in the second term and $S\dot{O}C$ is a nonlinear function of $P_{batt}(t)$ in Eq. (3.7). λ is then derived in terms of $a_0(t)$ from the battery charge-sustaining condition. The unconstrained optimal analytical input is solved by first differentiating the Hamiltonian with respect to input $P_{batt}(t)$, setting the equation to zero and solving for $P_{batt}(t)$ as a function of $a_0(t)$ array [59] which depends on the predicted vehicle loads.

3.6.3 Rule-Based HEV Powertrain Optimization

The optimization strategy presented in Chapter 3.3 to Chapter 3.5 is also benchmarked with ad-hoc Rule-Based method [65]. In general, there is a two-level controller in a Rule-Based strategy. The first selects the most efficient engine operating points by interpolating the iso-power curves of the combined power demands from the wheels and from the battery on the engine map [65,67]. Engine transient dynamics is also considered to avoid unfeasible engine speed or torque jumps between two time steps. The second contains a set of logic controllers that decides whether or not to implement the commands from the first controller [65]. For fair fuel comparisons, Rule-Based method will try to achieve CS operation within the last 50s of the simulation. Details of the Rule-Based method are explained in Appendix 2.

3.7 Small-Scale Traffic Simulation

3.7.1 Simulation Network

A traffic network with a total length of 6 miles and 16 intersections which are spaced between 600ft and 4000ft randomly, and a speed limit of 50 mph is built in VISSIM software. VISSIM is a commercial microscopic traffic simulator which is based on the Wiedemann's car following model [70] and has been used extensively in various applications related to traffic evaluation. Fixed-time traffic signal controllers are used with a common cycle length along the road. For simplicity, only one lane highway is simulated without turning vehicles at intersections to eliminate the randomness from lane-changing, as well as traffic inflow and outflow. All vehicles have total travel times between 650s and 800s.

3.7.2 Optimization Implementation

The optimization is implemented in a receding horizon manner. First, 20 vehicles speed trajectories are extracted from VISSIM simulation. For each vehicle, input P_{batt} is first calculated for the whole vehicle speed trajectory. Then, the first 40s of the optimized input P_{batt} are saved. The optimization is then repeated for vehicle trajectory from 40s to the final time, where input P_{batt} are then saved between 40s and 80s. Input P_{batt} is recalculated every 40s until the final 50s of the vehicle trajectory to represent the repeated optimizations when new vehicle speed trajectory is predicted and updated

every 40s. The proposed optimization takes less than 3s for a speed trajectory of 800s with 0.1s time step. Note that the optimization calculation time becomes quicker as the speed trajectory horizon becomes shorter towards the end of the driving cycle. Additional linear constraint is also added in SP, with and without input rate constraint, to ensure P_{batt} value at the end of 40s in the previous cycle is the same as P_{batt} value at the start of the current cycle.

3.7.3 Vehicle Speed Prediction

In a CV environment, first an ideal case of actual vehicle speed prediction is used to study the maximum potential benefits from the optimization. Then, considering that any prediction model is prone to errors, different levels of normally-distributed random-noise, between 0 to 1 mph, 0 to 3 mph and 0 to 5 mph are added to the actual speed trajectories to represent prediction uncertainties. The noises are injected at 5s intervals, where the speed trajectory are then re-interpolated based on the perturbed points. These predicted vehicle speeds are used by SP to optimize the powertrain, while the vehicle still follows the unperturbed actual vehicle speeds.

3.8 Simulation and Experimental Results

For comparison purposes, SP is implemented with and without the input rate constraints (IRC), denoted as SP-IRC and SP-NoIRC respectively. Note that IRC is given by Eq. (3.25, 3.26). First, the result for SP-NoIRC is compared with Dynamic Programming (DP) for a short driving cycle. Then, SP-IRC, SP-NoIRC, PMP and Rule-Based (RB) methods are implemented and compared for 20 vehicles. Comparisons between PMP and SP are then discussed. Next, the performance of SP-IRC, PMP and RB for a single vehicle is analyzed. Subsequently, the effects of different levels of vehicle speed prediction uncertainties on SP-IRC are presented. Finally, test results using SP-IRC reference engine operating points are presented.

3.8.1 Comparisons with Dynamic Programming

DP is computationally expensive and takes time to complete. Therefore, only partial driving cycle from one of the vehicles is used for comparison with SP-NoIRC. For a short driving cycle of 210s with a time step of 0.1s, DP takes 140 hours to solve using Intel i7-

4790 3.6GHz processor with 24GB of RAM. In contrast, SP-NoIRC only takes about 1.2 seconds to complete. The results are shown in Fig. (3.6-3.8).

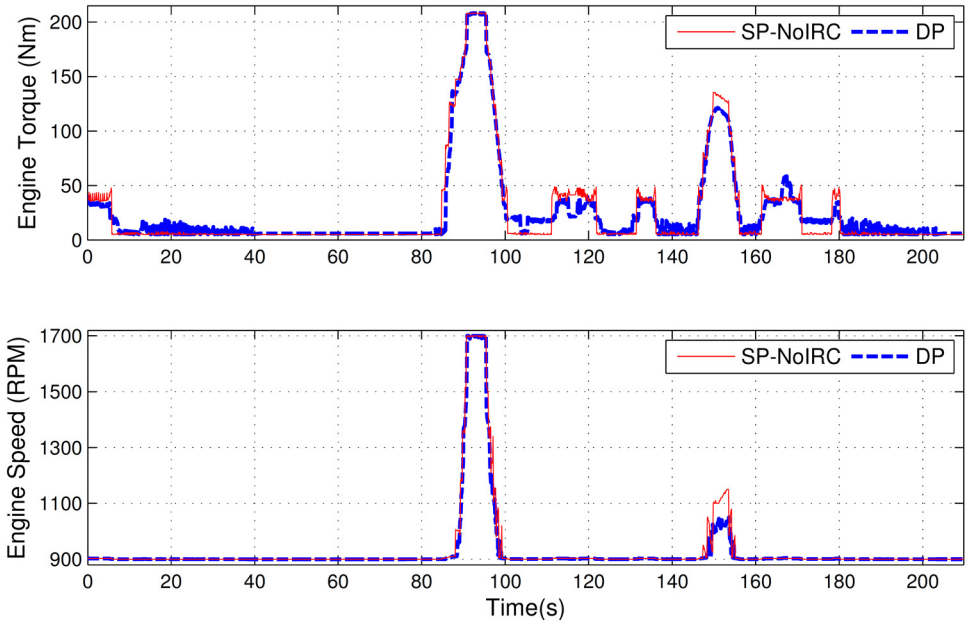


Figure 3.6. SP and DP Engine Torque and Speed

SP and DP achieve CS operation with comparable cumulative fuel consumption of 82.95 and 82.37 grams, or 64.16 MPG and 64.61 MPG respectively. DP performance is highly dependent of the iteration size of input P_{batt} candidate, which was calculated from $S\dot{O}C$ using Eq. (3.7) to enforce the charge-sustaining condition. $S\dot{O}C$ was incremented by 1×10^{-4} percent-per-second. A finer resolution of $S\dot{O}C$ increments will improve DP result, but would also increase the computation burden and time. With a small enough input iteration for DP, it is shown that SP-NoIRC performance is comparable to DP.

From Fig. 3.6, the engine operating points for DP and SP-NoIRC have an almost similar trend, but there are some differences. SP-NoIRC operates at the highest P_{batt_hi} line, or engine idling point, during high-speed-low-deceleration and deceleration regions at 100-110s and 170-178s, while DP operates at engine idling speed with higher engine torques. This therefore causes the battery *SOC* for SP-NoIRC to dip faster than DP in these regions from using all-electric mode as shown in Fig. 3.7. Similarly, SP-NoIRC also operates at engine idling points during high decelerations at 15-40s, 135-145s, and 180-200s. DP on the other hand operates at slightly higher engine operating points as shown by the higher *SOC* charging rate at these regions. SP-NoIRC balances this lower

SOC by operating at higher engine operating points than DP at high-acceleration regions at 80-100s and 150-155s, and high-speed-low-acceleration regions at 110-120s and 130-138s.

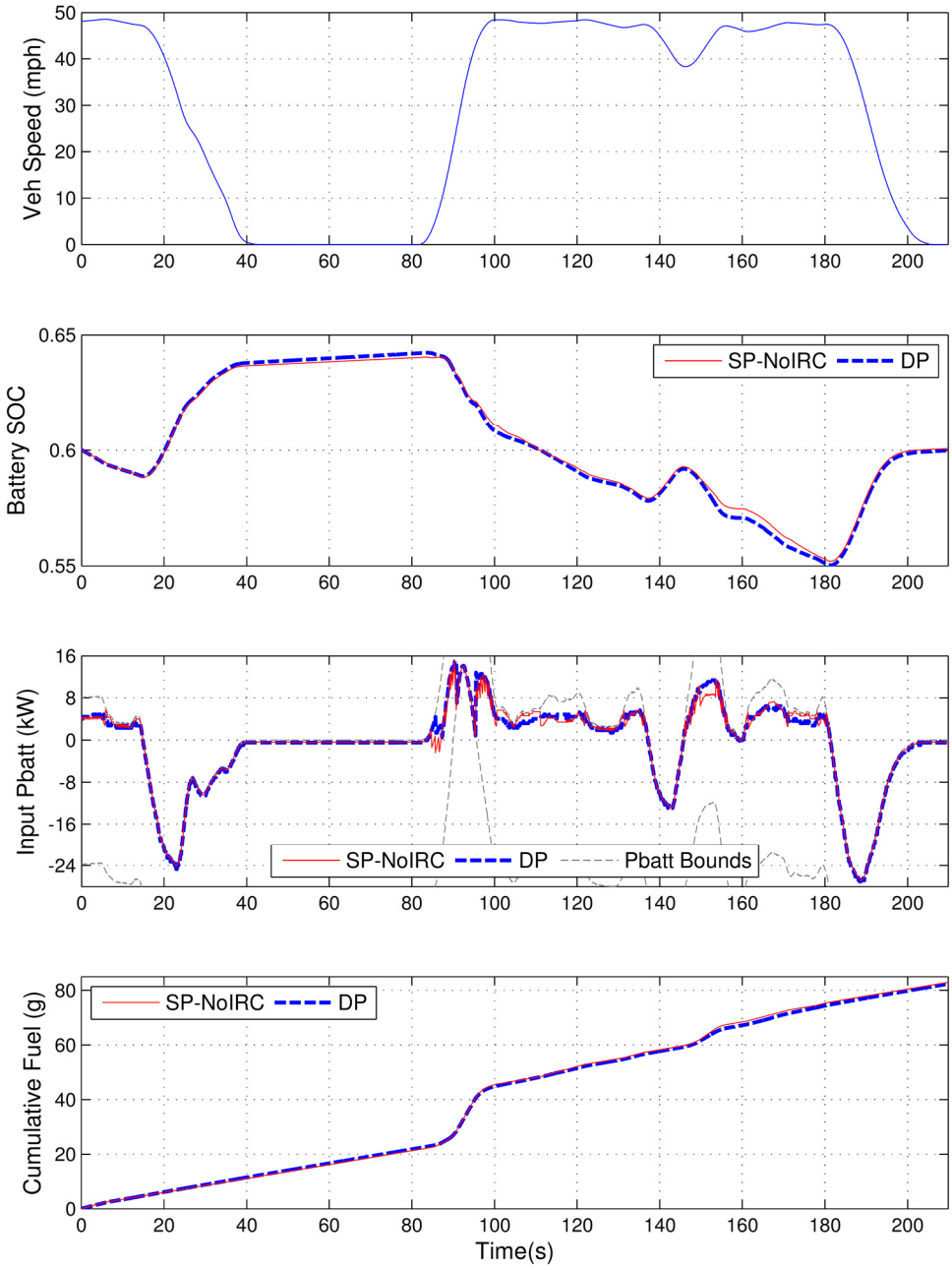


Figure 3.7. Vehicle Speed, Battery *SOC*, P_{batt} Input and Fuel Consumption

Figure 3.8 shows the engine operating points of SP-NoIRC and DP, bounded above by the maximum torque line. First, note that for SP-NoIRC, groups of engine operating

points are isolated in different clusters across the map, which shows the engine operating points are allowed to move freely due to unconstrained input P_{batt} rate. The input P_{batt} rate in DP is not constrained, but is being limited by the maximum jump of the iterated SOC candidates between two time steps, chosen at 1 percent-per-second, which is twice as large as the SOC jump from SP-NoIRC result and 10^4 times larger than the iterated $S\dot{O}C$ for DP. Scattered data, for example, along the maximum torque line, shows the transition points of DP engine operating points. DP operates less frequently than SP-NoIRC at less efficient engine idling points as shown in Fig. 3.6. This can be attributed by a more accurate estimation of the relationship between minimum $\dot{m}_{fuel}$ cost and input

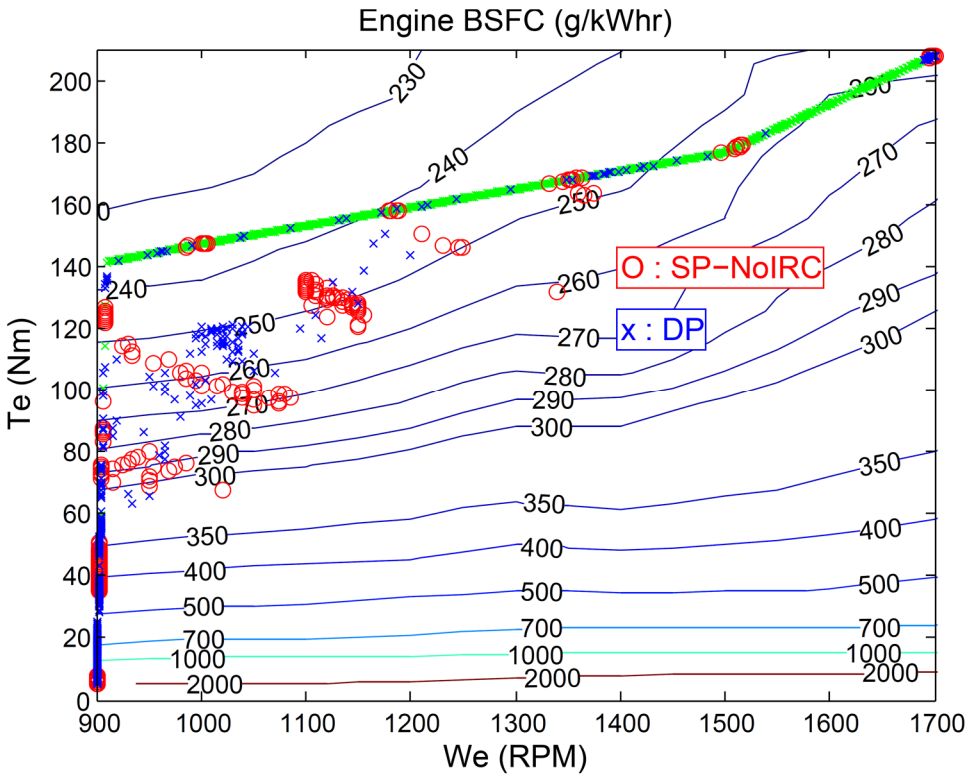


Figure 3.8. SP-NoIRC and DP Engine Operating Points

P_{batt} for DP by using more sampling points than SP-NoIRC. DP has lower cost estimates for sampling points between the SP-NoIRC sample points due to the convexity shown in Fig. 3.4. Therefore, P_{batt} values for engine operating points slightly above engine idling has lower cost estimate for DP than SP-NoIRC. However, overall SP-NoIRC also operates at slightly higher engine torque and speed than DP, which are more efficient. A

combination of high and low efficiency operating points of SP-NoIRC therefore offers a comparable performance with DP, with significantly less computation time.

3.8.2 Comparisons with PMP and Rule-Based Optimizations

Overall, SP-NoIRC and SP-IRC results show MPG improvements over PMP and RB in Fig. 3.9. On average, SP-NoIRC is 4.89 MPG (10.9%) higher than RB, 2.11 MPG (4.4%) higher than PMP and 0.21 MPG (0.4%) higher than SP-IRC. SP-IRC is also 4.68 MPG (10.4%) and 1.90 MPG (4.0%) higher than RB and PMP respectively. The improvements of SP-IRC and SP-NoIRC over RB method are expected, but the improvements over PMP are discussed in the following.

First, analytical solution from PMP in [59] is derived by assuming input P_{batt} is unconstrained. Therefore, out-of-bound inputs are manually adjusted to stay within the boundaries while maintaining CS operation. On the other hand, Separable Programming uses Linear Programming methods such as Interior-Point to incorporate the input boundaries in the problem formulation, therefore yielding better optimized inputs than PMP.

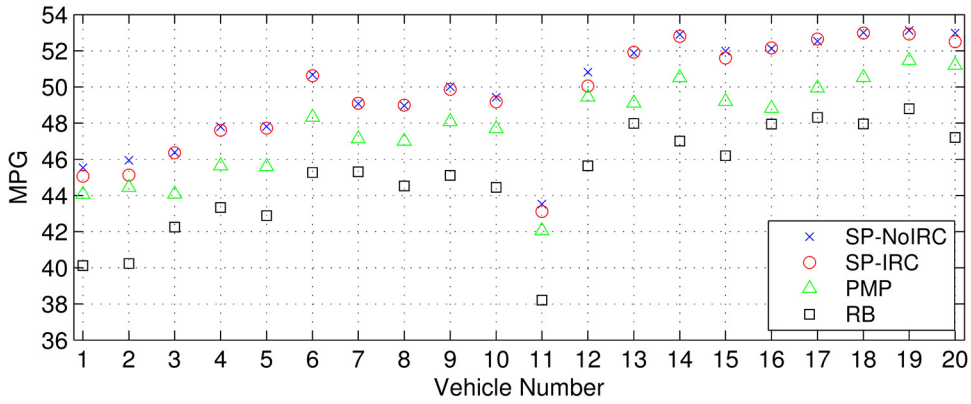


Figure 3.9. MPG Results from Simulation

In PMP [59], the optimized unconstrained input P_{batt} mainly violates the upper bound P_{batt_hi}, which is the engine idling point. These points are then constrained at P_{batt_hi} line, but the summation of $S\dot{O}C$ using the reduced P_{batt} yields a non charge-sustaining (CS) result. To enforce a CS operation, other P_{batt} within the bounds are incrementally increased until a CS operation is achieved. However, forcing the P_{batt} that violates the

upper-bounds to be exactly on the P_{batt_hi} line may not be beneficial in terms of fuel efficiency, as discussed next.

Figure 3.10 shows the minimum $\dot{m}_{fuel}$ for each P_{batt} values iterated between P_{batt_lo} and P_{batt_hi}, representing the highest and lowest engine operating points respectively. For every constant P_{batt} value, represented by the grey dashed line, the engine speed is iterated from 900RPM to 1700RPM to find the minimum $\dot{m}_{fuel}$. P_{batt} is higher at the bottom of the engine map, where the engine power is low, because it represents the portion of power from the battery to meet the vehicle demand. The minimum $\dot{m}_{fuel}$ engine operating points are then plotted on the Engine Fuel map as shown in Fig. 3.10. The engine operating points are bounded above by the maximum torque line. PMP manual-tuning forces P_{batt} that violates the P_{batt_hi} bound, or engine idling, to stay on the bound which has the lowest efficiency as also shown in Fig. 3.8. This can be seen in Chapter 3.8.3 where PMP operates more at engine idling compared to SP, therefore making it less efficient.

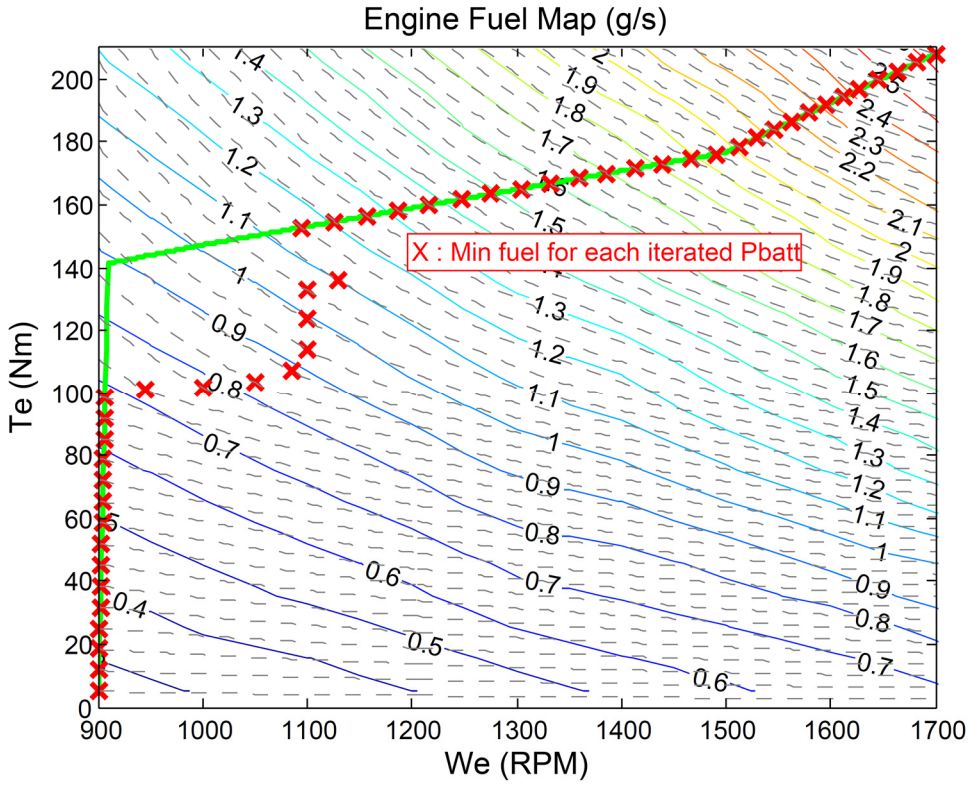


Figure 3.10. Minimum $\dot{m}_{fuel}$ for Every P_{batt} Iterate

In general, SP achieves better fuel economy than PMP, but the calculation time takes relatively longer. On average, for an 800s drive cycles with 0.1s time step, PMP takes less than 1s to solve, while SP takes less than 3s which may offset the time when the input can be executed. However, this delay is still relatively small compared to the 40s update time.

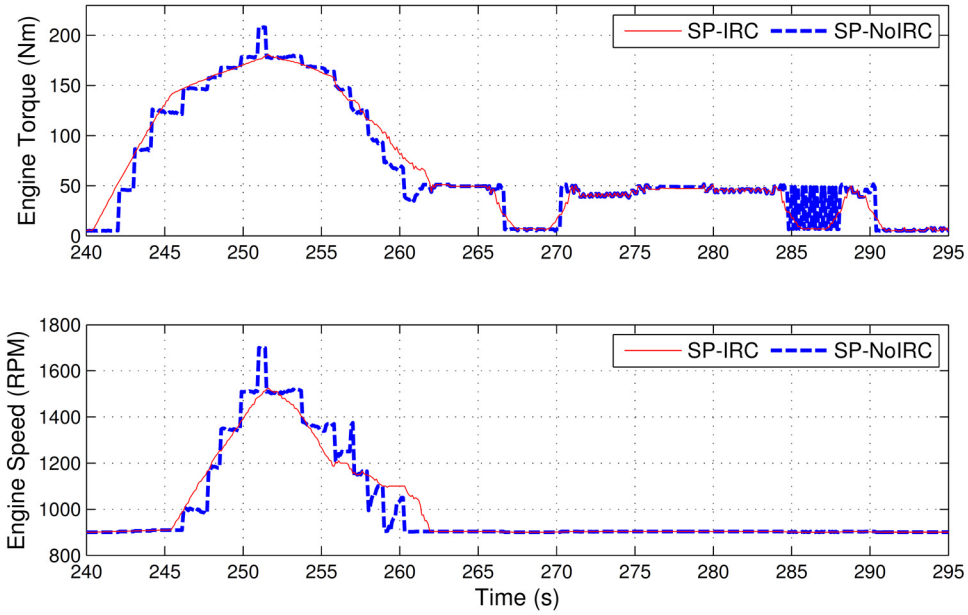


Figure 3.11. SP-IRC and SP-NoIRC Engine Operating Points Comparison

On average, SP-IRC performs only slightly worse than SP-NoIRC. This can be explained from the engine operating points comparison for vehicle 11 in Fig. 3.11. SP-NoIRC resulted in an unconstrained engine dynamics where the engine speed and torque change abruptly. The input rate limit in SP-IRC dampens out the transients, which resulted in an averaged engine torque and speed dynamics of the SP-NoIRC case. Due to the averaging effect of the engine operating points, the fuel consumptions between the two cases are similar. Input P_{batt} and relative P_{batt}-jump between two time steps for SP-IRC and SP-NoIRC are shown in Fig. 3.12. The change in P_{batt} is more restricted, resulting in slower and more practical engine transients, by implementing the constraints in Eq. (3.25, 3.26) with ΔP_{batt} value of 0.5 kWatts. Note there are slight violations of relative P_{batt}-jump due to the tolerance value used in the solver, which however does not adversely affect the engine dynamics.

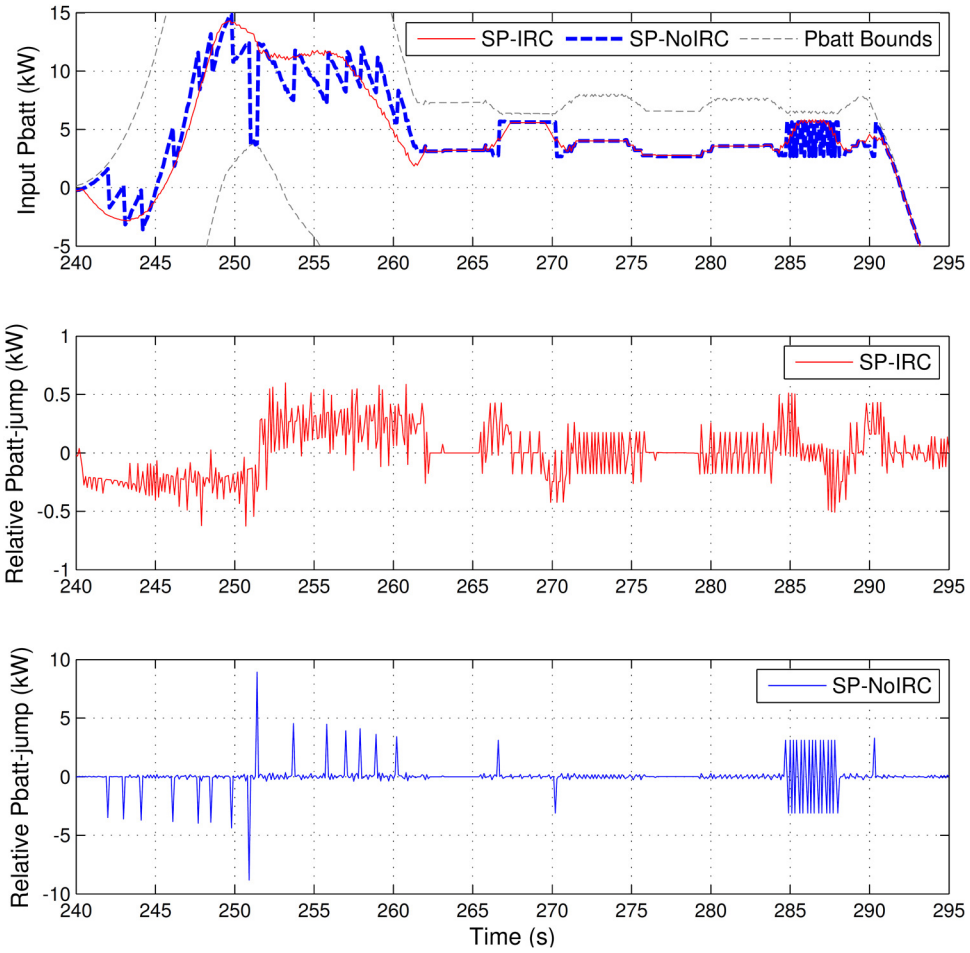


Figure 3.12. SP-IRC and SP-NoIRC Input P_{batt} and Relative P_{batt} Jump

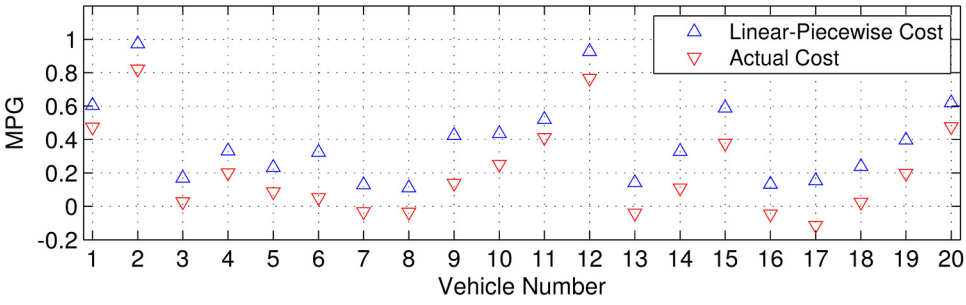


Figure 3.13. SP-IRC and SP-NoIRC MPG Difference

Figure 3.13 shows the comparisons of MPG difference between SP-No-IRC and SP-IRC, calculated using linear-piecewise cost function and using actual fuel map. SP-NoIRC achieved higher MPG for all vehicles using the approximated piecewise-linear cost. However, when translating the optimized input on actual fuel map, SP-IRC achieved slightly higher MPG for vehicle 7, 8, 13, 16 and 17, which is attributed by the

approximation error in the cost function. Nevertheless, the difference is less than 0.2 MPG as shown in Fig. 3.13, which is negligible.

3.8.3 Single Vehicle Simulation and Experimental Results

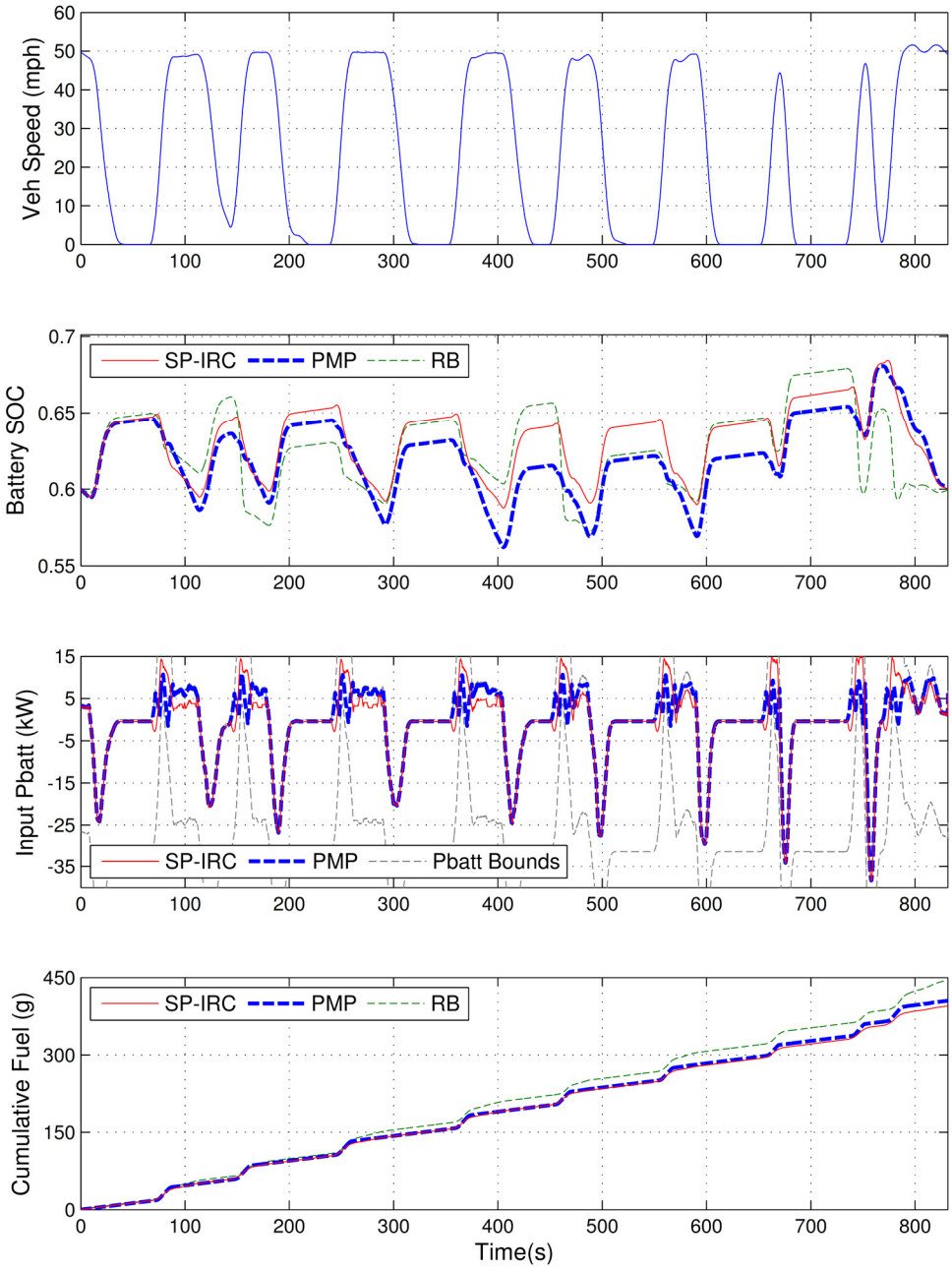


Figure 3.14. Vehicle Speed, Battery *SOC*, P_{batt} Input and Fuel Consumption

Vehicle 11, which has a driving cycle of 831s with 7 stops, is selected to analyze and compare the different optimization procedures. Simulation results showed SP-IRC, PMP

and RB achieved 43.11 MPG, 42.05 MPG and 38.2 MPG respectively. Figure 3.14 shows the driving cycle, *SOC* trajectories, input P_{batt} and cumulative fuel consumptions.

As an ad-hoc method, RB can only rely on the current vehicle load and could not use future information to optimize the engine. For example, without knowing the driving cycle would end at 831s, RB depletes the battery charge to 60% during an acceleration event at 770-800s and has to use engine-only mode the remaining of the cycle to maintain a CS operation, causing cumulative fuel to increase drastically in the last 30s. In contrast, knowing the future a-priori, SP-IRC and PMP discharges the battery moderately at 770-831s for CS operation.

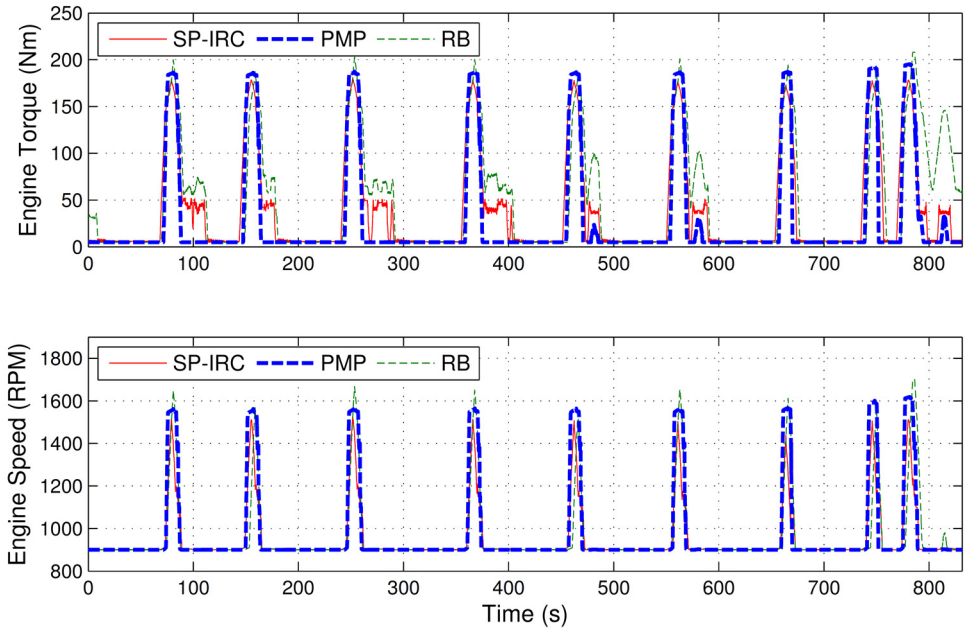


Figure 3.15. Vehicle 11 Engine Operating Points

Shown in Fig. 3.15 are comparisons of engine operating points between SP-IRC and PMP which is zoomed-in at 240-295s in Fig. 3.16. Unlike PMP, SP-IRC is shown to avoid running at engine idling, which is less efficient for the engine, during high-speed-low-acceleration regions around 100s, 170s, 270s, 375s, 480s, 570s and 800s. In these regions, SP-IRC operates with an engine torque of around 50Nm, while PMP operates at the lowest engine operating points. As a result, SP-IRC generally has lower peak engine speed and torque compared to PMP during high vehicle accelerations by utilizing more battery. SP-IRC however operates the engine more during cruising speed to recharge the battery, for example between 261s and 290s, as shown in Fig. 3.16. RB shows

higher engine operating points during vehicle cruising in Fig. 3.15. This is because at initial vehicle acceleration, where the demanded vehicle power is low, RB utilizes all-electric operation which keeps the engine operating points low, as shown at the beginning of Fig 3.16, which however depletes the battery charge. RB has to recharge the battery more afterwards, leading to higher engine operating points.

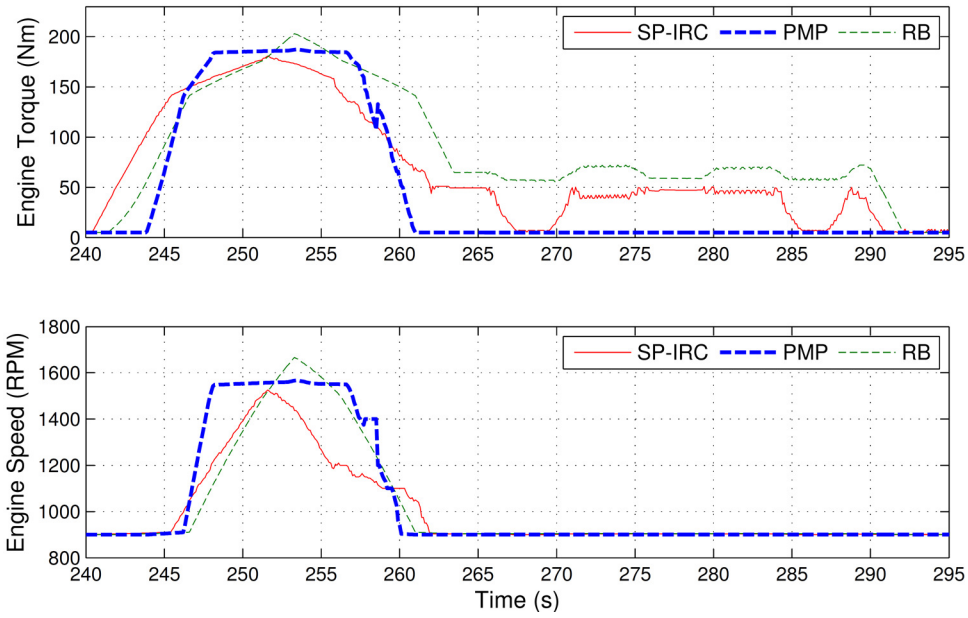


Figure 3.16. Vehicle 11 Engine Operating Points between 240s and 295s

Comparisons of engine operating points between SP-IRC, PMP and RB between 240s and 295s are shown in Fig. 3.16. Unlike PMP, SP-IRC is shown to avoid running at inefficient engine idling during vehicle cruising speeds. RB operates higher during this period to maintain battery *SOC* which is low compared to SP-IRC and PMP at 240s, as shown in Fig. 3.14. The engine operating points for SP-IRC and PMP are plotted against the engine efficiency map in Fig. 3.17. RB operating points falls along the maximum torque line and therefore are not shown. The operating points are bounded above by the maximum torque line. PMP engine operating points are concentrated in the extreme regions, at the top right and bottom left corners, of the engine map. SP-IRC engine operating points are similarly distributed but are more concentrated slightly towards the top-left corner, where the engine efficiency is higher. This shows that SP-IRC minimizes fuel by operating the engine at higher efficiency regions by utilizing more battery during vehicle acceleration, hence lowering peak engine operating points and recharging the

battery back during low vehicle demands such as cruising speeds, hence avoiding engine idling by increasing the lowest engine operating points.

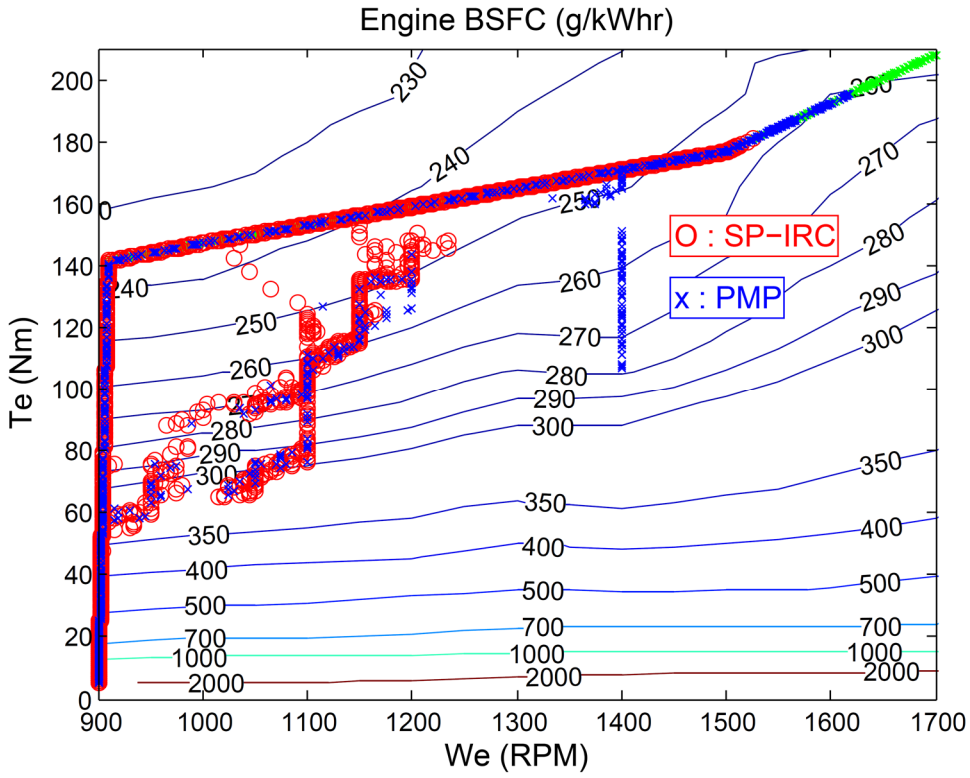


Figure 3.17. SP-IRC and PMP Engine Operating Points

In order to show the feasibility of the optimizations, results for SP-IRC and RB are tested on a 4.5L John Deere diesel engine as part of a powertrain research platform [61,74] that is capable of tracking transient engine operating points. Test results show good tracking, with tracking errors of about 10% and 1% for the engine torque and engine speed respectively, as shown in Fig. 3.18-3.19. Actual fuel consumption measured for SP-IRC and RB are 401.5g and 450.8g respectively, which are only 1.67% and 1.26% higher than simulation results. The test results shows that the engine operating points from SP-IRC optimization are feasible and can be implemented accurately on a real engine, which also shows fuel improvements over an ad-hoc RB optimization method.

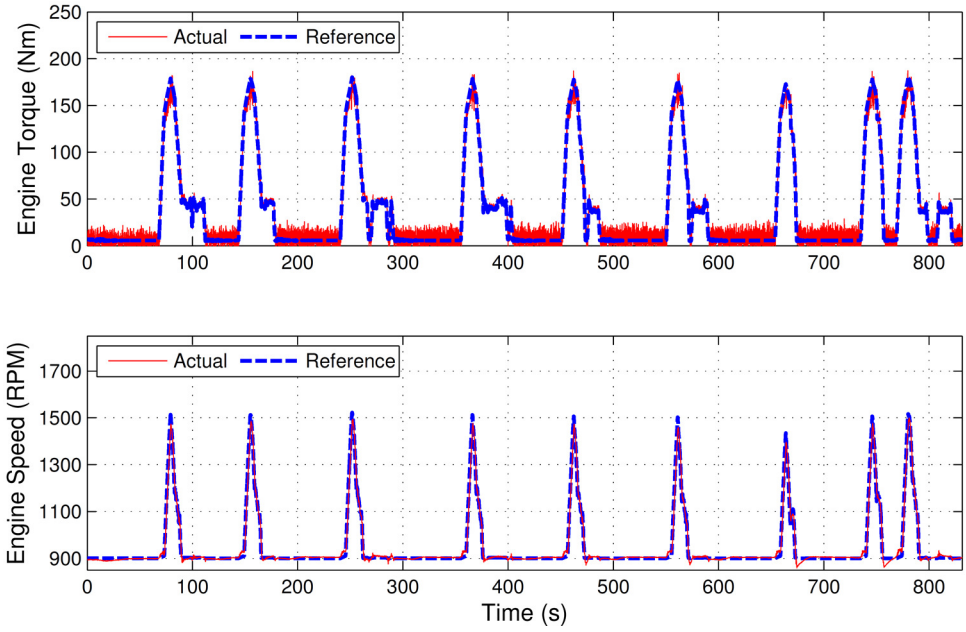


Figure 3.18. Test-Engine SP-IRC Operating Points

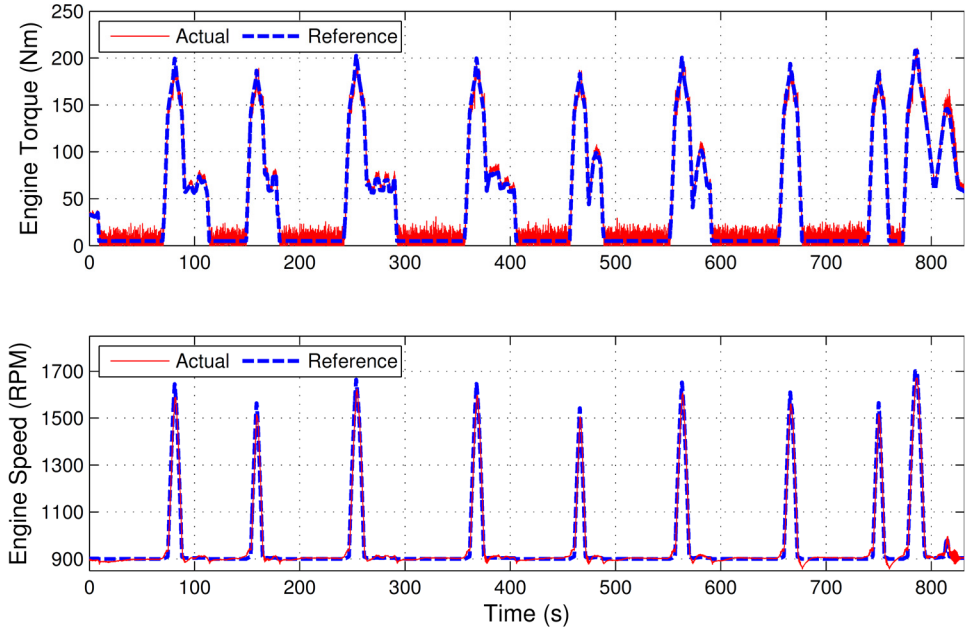


Figure 3.19. Test-Engine RB Operating Points

3.8.4 Effects of Prediction Uncertainties

Figure 3.20 shows the average MPG for SP-IRC without and with different levels of normally distributed vehicle speed errors added to the actual speeds of 20 vehicles. With 0-1 mph, 0-3 mph and 0-5 mph added speed errors, the average Mean-Absolute-

Deviation-Percentage (MADP) calculated against actual vehicle speed are 1.9%, 2.2% and 2.7% respectively.

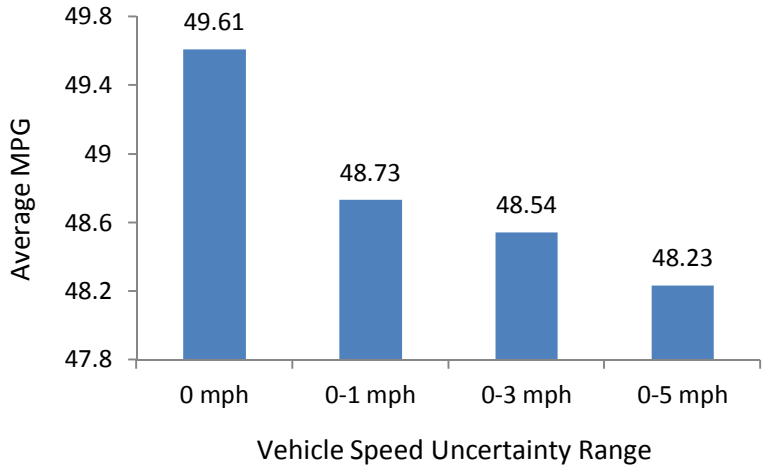


Figure 3.20. Effects of Prediction Uncertainties

Fuel benefits over RB (44.93MPG) for 0 mph, 0-1 mph, 0-3 mph and 0-5 mph speed errors are 10.4%, 8.5%, 8.0% and 7.3% respectively, which shows deterioration as the speed errors increase. SP-IRC is formulated without considering prediction uncertainties, resulting in evident performance degradation as prediction errors are increased.

3.9 Conclusion

In this work, the HEV powertrain optimization problem is solved in two steps. First, given the flexibility of the power-split HEV powertrain, the relationship between the minimum fuel consumption and the power-split levels between the engine and battery is calculated and stored offline for all possible vehicle power demands. This relationship ensures the engine is operating at the most optimal point for every possible HEV power-split level for a given vehicle power demand. Then, the nonlinear fuel cost and battery dynamics are approximated by linear piecewise functions and formulated as a Separable Programming (SP) problem. The piecewise-linear functions introduce new dimensionless variables which are solved as a large-dimension constrained linear problem with efficient Linear Programming (LP) solvers. The engine operating range and engine transient dynamics are represented as linear constraints in the LP problem to ensure the engine operating points are feasible. Traffic information from CV is integrated

in the optimization by integrating the driving cycle prediction into the powertrain optimization. Fast calculation time allows the optimization method to be implemented repeatedly in a CV environment with repeated speed prediction updates due to the transient nature of traffic.

Fuel economy of SP with and without input-rate-constraint (SP-IRC and SP-NoIRC) are measured against Dynamic Programming (DP), Pontryagin's Minimum Principle (PMP) and Rule-Based (RB) methods using driving cycles generated in VISSIM traffic simulator. First, it is found that SP-NoIRC gives comparable fuel economy as DP for a short driving cycle. Then a fleet of 20 vehicles are simulated and optimization results are compared between SP-NoIRC, SP-IRC, PMP and RB. It is found that the SP-NoIRC achieves average improvements of 4.4% and 10.9% compared to PMP and RB respectively. SP-IRC on average is only 0.4% lower than SP-NoIRC due to engine dynamics averaging effects from the input rate constraint. SP-IRC test results show the feasibility of the optimized engine operating points and the fuel benefits over RB method.

Despite having better performance, numerical SP optimization takes relatively longer to solve than analytical PMP solution. Furthermore, vehicle speed prediction uncertainties are shown to deteriorate the performance of SP. Therefore, as future works, reduced number of strategically placed sampling points to accurately characterize the cost and battery state-of-charge can be used to reduce the number of dimensionless variables in the LP solver, hence the SP calculation time. In addition, a robust optimization strategy that can handle vehicle speed prediction uncertainties, while maintaining cost optimality, is needed to guarantee superior performance in a CV environment.

Chapter 4

Hardware-in-the-Loop Testbed for Evaluating Connected Vehicle Applications

4.1 Motivation

Technologies associated with Inter-Vehicle Communications (IVC) and Vehicle-Infrastructure Integration (VII) have not only gained traction in research communities, but also policy makers with the Department of Transportation's proposal for installing communication devices in new vehicles in the near future. Traffic information sharing between vehicles, also known as connected vehicle, is therefore seen as the future of road transportation to improve traffic mobility and safety. Connected vehicle technology also allows better optimization of a vehicle's fuel economy and emissions by utilizing traffic information such as the traffic light Signal-Phase-and-Timing (SPaT) and surrounding vehicles speed information. Researches in utilizing connected vehicle technology to optimize fuel use and emissions are mostly done in simulations, while actual testing on real vehicles is limited due to safety, cost and technical challenges. Consequently, the simulation results may not represent the actual fuel and emissions benefits precisely. Currently the performance of a vehicle's fuel economy and emissions in traffic is measured through either simulation or by instrumenting the vehicle. First, a simulation-based approach replaces the engine with steady-state fuel-use and emission maps and therefore may not be accurate compared to actual measurements. Secondly, instrumenting vehicles is time consuming and expensive since it requires major modifications of the vehicles. In addition, equipping large precision measurement devices on small passenger vehicles is challenging for testing purposes. Therefore, A Hardware-in-the-Loop-System (HiLS) is proposed to offer the flexibility and accuracy of evaluating the performance of connected vehicle applications.

4.2 Objective and Scope

First, the main objective of the HiLS is to merge a powertrain research platform with a traffic simulator, VISSIM. Using the powertrain research platform, a real engine can be loaded through an engine-loading device while fuel and emissions are measured in real time. VISSIM on the other hand provides the dynamics of a selected vehicle and road environment to calculate the engine load, while simultaneously simulating the background traffic dynamics. Secondly, remote communication between the powertrain research platform and VISSIM has to occur in real-time. Flexible operation of VISSIM in a remote location will allow researchers from different locations to conduct traffic simulations and collect real fuel and emission measurements quickly without being present at the powertrain research platform facility. Real-time communication is needed to ensure accurate tracking of the vehicle speed. Utilizing a small enough time-step, tracking error is minimized in case of communication delay. Finally, the testbed must be capable of tracking various vehicle speed profiles under different traffic scenarios accurately through the real engine and virtual powertrain to ensure various connected vehicle applications, such as the CACC, can be tested.

4.3 HIL System Architecture

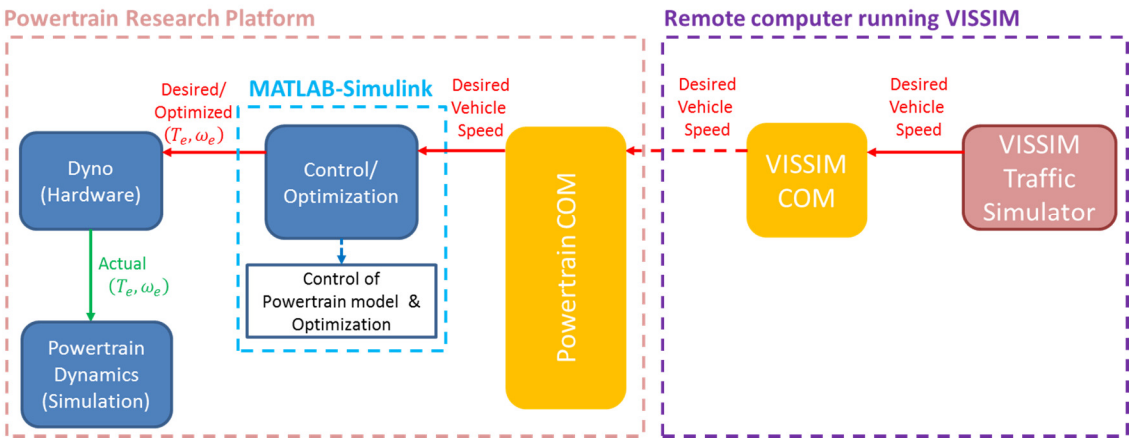


Figure 4.1. HiLS Architecture

The HiLS architecture is shown in Figure 4.1. In the diagram, the powertrain research platform is located remotely from a computer running VISSIM simulation. VISSIM simulation executions and data transfer are handled by VISSIM-COM, which is a separate program coded in C# language. VISSIM-COM has three functions:

1. Controls the execution timing of VISSIM simulation at fixed time-step intervals to ensure real-time execution.

2. Extracts traffic data from VISSIM simulation at every time-step.

3. Sends traffic data from remote computer to powertrain research platform through internet network at every time-step.

Details of execution timing, data extraction and data sending for VISSIM-COM will be explained in the HiLS Middleware section (Chapter 4.5). On the powertrain research platform side, traffic data is received and handled by the Powertrain-COM, which is also coded in C#. The Powertrain-COM has two functions:

1. Receives traffic data from VISSIM-COM through the internet network.

2. Sends traffic data to the hardware controller in MATLAB-Simulink at fixed time-step intervals.

The traffic data received is used to calculate the vehicle load where a powertrain optimization method is then used to optimize the engine torque and speed. The desired optimal engine operating point is then tracked by the hardware controllers, while fuel consumption and emissions from the engine are measured. Actual engine operating point readings are then used to calculate the realized vehicle speed in the virtual powertrain dynamics. If needed, the realized vehicle speed can also be sent back to VISSIM through the middleware to reflect (1) the powertrain dynamics and constraints, such as engine and gear shift delay and battery state-of-charge limits, and (2) the realized vehicle power from powertrain optimization. Details of the hardware controllers will be further discussed in the HiLS Components section (Chapter 4.4).

In the powertrain research platform, the hardware operates continuously, while Powertrain-COM feeds updated traffic data at fixed time-step intervals to the hardware controller. This is unlike VISSIM-COM, which controls the executions of VISSIM simulation at every time-step. The two COM-softwares also ensure VISSIM and the powertrain research platform operations are synchronous. Details of the powertrain research platform and VISISM are discussed in the HiLS Components section (Chapter 4.4). The HiLS Middleware, consisting of the Powertrain-COM, VISSIM-COM and network communication between them, is discussed subsequently.

4.4 HIL System Components

The main components of the HiLS are the Powertrain Research Platform, which represents the target vehicle being tested, and VISSIM traffic simulator, which provides the target vehicle dynamics and road conditions.

4.4.1 Powertrain Research Platform

The proposed HiLS will utilize an existing powertrain research platform that was developed in the University of Minnesota [74-77] as shown in Figure 4.2. The platform was developed to expedite the investigation of vehicle powertrain architectures and control strategies to engine's fuel consumption and emissions. The platform consists of a real engine and a hydraulically actuated engine loading device (dynamometer), while the vehicle powertrain dynamics and controls are captured virtually through simulation. A real engine is used because the combustion and emission behavior of an engine is too complex to be modeled accurately [35,36] for real-time application, while the dynamics of a powertrain can be captured accurately with well-developed models.

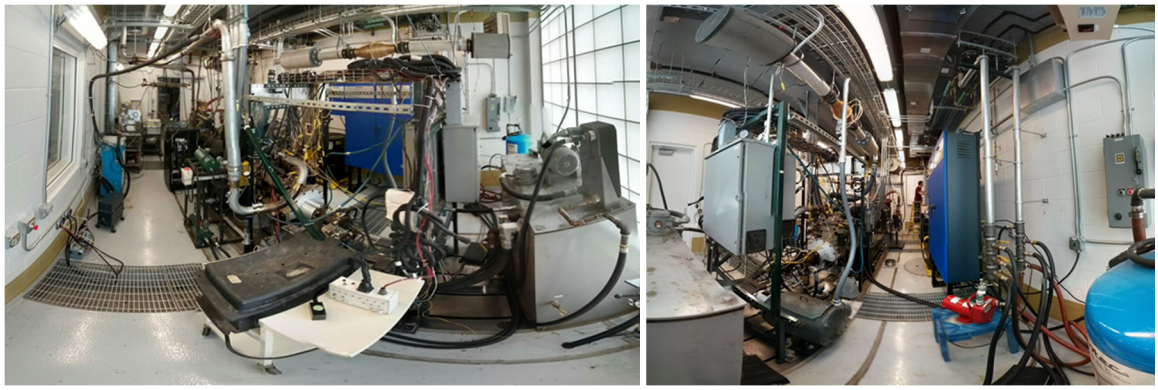


Figure 4.2. Powertrain Research Platform

The control and simulation is defined by a three-level closed-loop architecture [74-77] as shown in Fig. 4.3. In the high-level controller, given a power demanded from the vehicle, the user-defined energy management system (EMS) will select a reference engine operating point that optimizes fuel-use and emissions. In the middle-level controller, the virtual-torque-controller will control the powertrain torques that realizes the reference engine torque from the high-level controller. Well-developed models are used to simulate the dynamic responses of the powertrain components, which include the desired engine

loading torque. In the low level controller, the dynamometer is then controlled to track the desired engine loading torque from the middle-level controller. Fuel consumption and emissions from the engine can then be measured by precision measurement instruments.

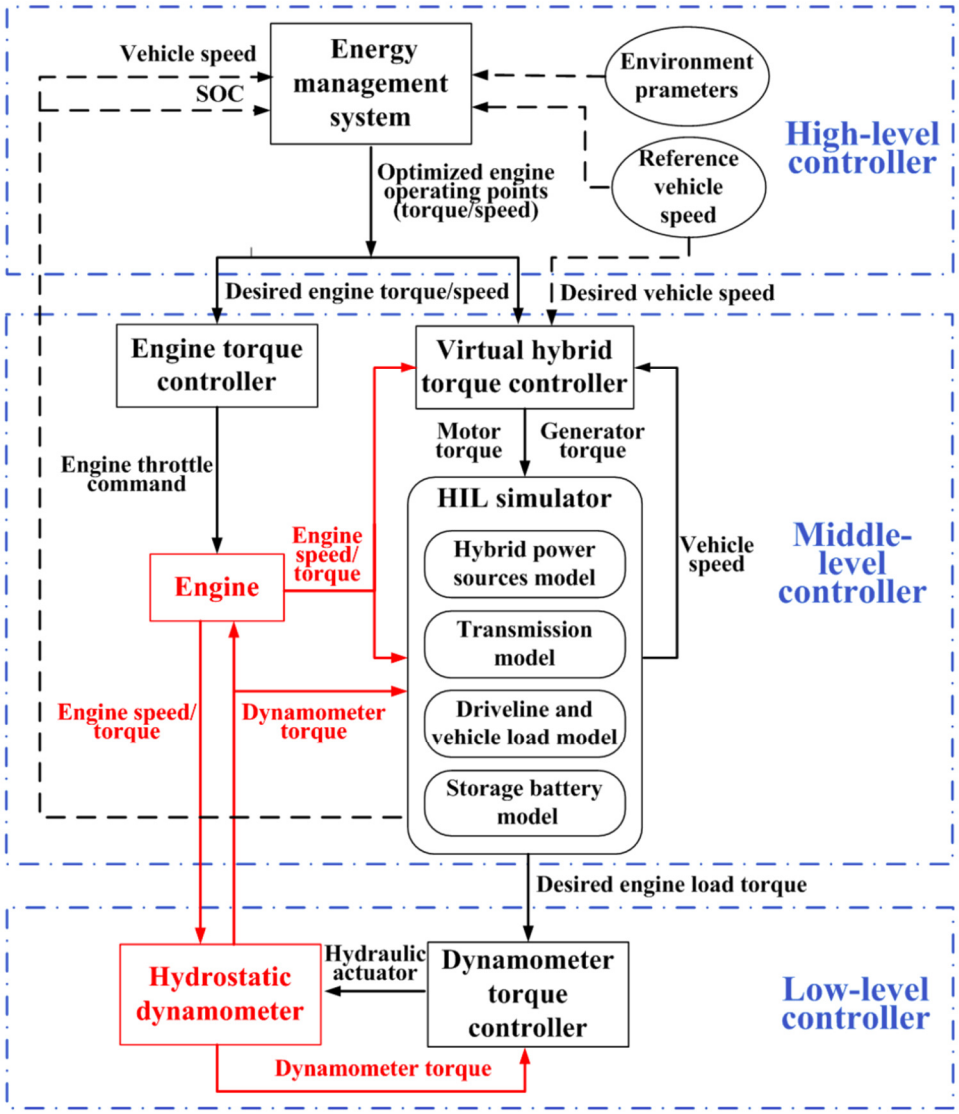


Figure 4.3. Three-Level Controller of Powertrain Research Platform [74]

Fuel consumption is measured by AVL's Fuel Measurement System Model P402 with measurement uncertainty of 0.1% and output frequency of up to 80kHz. The emissions are measured using AVL's SESAM-FTIR, which can measure up to 25 components of

exhaust gas from engine combustion including NO_X, CO, CO_2 and HCHO with a sampling rate of 1Hz.

4.4.2 VISSIM Microscopic Traffic Simulator

VISSIM is a commercial microscopic traffic simulator which is based on the Wiedemann's car following model [70] and has been used extensively for various applications related to traffic evaluations. The software allows users to access traffic simulation states, such as vehicle speed, road conditions and signal phase and timing, at every simulation time-step. A simple network communication model is programmed as a dynamic-link-library (DLL) in VISSIM, by fitting experimental data from the Safety Pilot Model Deployment Program [12], to simulate the BSM packet drop in V2V communication. The model provides the probability of BSM data transmission success based on the distance between two communicating vehicles. Data transmission success or failure between every two vehicles at each time step is then determined as a random occurrence based on the probability.

4.5 HIL System Middlewares

The HiLS middleware consists of the Powertrain-COM and VISSIM-COM. First, local communications which define the interactions of VISSIM-COM with VISSIM simulation and Powertrain-COM with the powertrain research platform controller are discussed. Then, internet network communication between Powertrain-COM and VISSIM-COM is explained. Finally, data synchronization from VISSIM simulation data extraction to data transfer to the hardware controller is discussed.

4.5.1 Local Communication

Both VISSIM-COM and Powertrain-COM utilize the Component-Object-Model (COM) interface which is a standard for inter-software communication [78]. Software packages that are built with COM capability allow predefined objects in the software packages to be readable and writeable by an external program, which is usually written in a programming language.

VISSIM is a COM-capable software. Traffic data, such as the target vehicle speed and road angle, are predefined as COM objects in VISSIM and therefore accessible from an external program (VISSIM-COM) after each simulation time-step. In addition, the

execution of VISSIM simulation-run at every time-step can be triggered by VISSIM-COM. Therefore, when executed, VISSIM-COM performs the following:

1. Initialize VISSIM software and load the simulation files.
2. Perform a single time-step simulation run.
3. Extract the desired traffic data and send it to Powertrain-COM over internet network.
4. Wait until the end of the real-clock time-step.
5. Repeat Step 2 to 4 until the end of simulation.

For faster extraction of large traffic data (Step 3), instead of accessing COM objects, a DLL that runs internally in VISSIM sends the desired traffic data to a User-Datagram-Protocol (UDP) virtual network port on the local computer that can be accessed by VISSIM-COM. For connected and autonomous vehicle applications, VISSIM-COM can transfer the large traffic data to an external connected vehicle controller to calculate the desired speeds for selected vehicles. The desired vehicles speeds are then sent back to VISSIM and the Powertrain-COM through VISSIM-COM. Figure 4.4 shows the time measured to extract traffic data from VISSIM using DLL. Extraction time was measured at every simulation time-step of 200ms, from 0 to 89 vehicles entering the traffic. Extracted traffic data for every vehicle include vehicle ID, road-lane ID, vehicle longitudinal and latitudinal coordinate, vehicle speed and vehicle acceleration. Fast data extraction time duration of less than 0.06ms for up to 89 vehicles is shown.

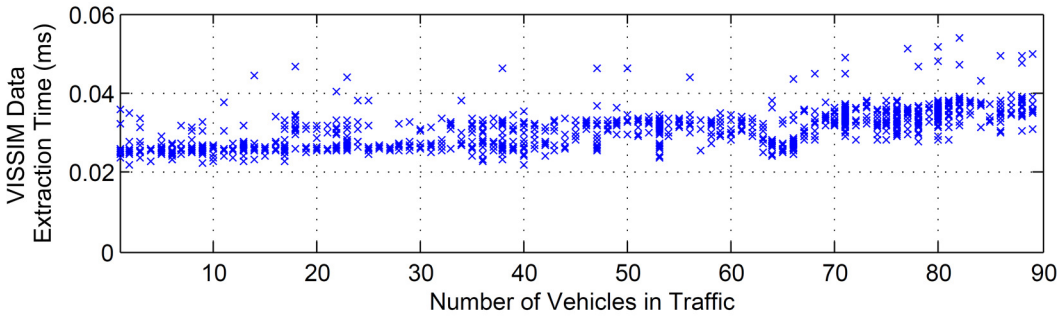


Figure 4.4. VISSIM Traffic Data Extraction Time as Vehicles Enters Simulated Traffic

Details of the Powertrain-COM are discussed as follows. COM is also enabled in MATLAB through the MATLAB COM Automation Server. Unlike VISSIM whose COM objects are predefined, MATLAB COM Automation Server allows access to parameters in MATLAB-Simulink blocks and user-defined MATLAB workspace variables. With this feature, the Powertrain-COM performs the following:

1. Requests traffic data from VISSIM-COM over the network.
2. When traffic data is received, it updates the parameters in the high-level powertrain controller in MATLAB-Simulink.
3. Maintains the value of traffic data until current real-clock time-step is over while keep requesting for updated traffic data.
4. Use updated traffic data received in Step 3 in the next time-step and repeat Step 2 to 4 until the end of VISSIM simulation.
5. Throttles down the real engine once VISSIM-COM sends signal to indicate the end of simulation.

Note that before the Powertrain-COM is initiated, the hardware is already running at a predefined constant vehicle load. This load will then change when updated traffic data is received.

4.5.2 Network Communication

Transmission-Control-Protocol (TCP) is used as the network communication transport protocol because of the reliability of data transfer and ordered data delivery. In TCP, buffer memories are allocated on VISSIM-COM side for data sending and on Powertrain-COM side for data retrieval. These buffers ensure that data is not lost during the transfer. TCP also ensures the order of data is preserved on the receiving side, which is important in the HiLS application to distinguish the traffic data contents. User-Datagram-Protocol (UDP) transport protocol is faster than TCP, but is not used due to unreliable data transfer and disordered data. Reliable and ordered data delivery is important because data loss will affect the accuracy of the tests, while elements of the received traffic data have to be distinguished for calculation purposes. Although TCP is relatively slower than UDP, it is fast enough for the HiLS application.

At start-up, Powertrain-COM is designed to continuously send requests for a connection with VISSIM-COM. In order to establish a connection, VISSIM-COM opens a port in the socket of the remote computer running VISSIM to accept connection request from the Powertrain-COM. The socket address is defined by the internet protocol (IP) address of the remote computer and the port number. Therefore, socket connection is established as soon as VISSIM-COM opens the port.

Utilizing TCP, VISSIM-COM and Powertrain-COM sends and retrieves data from their respective buffer memories. However, since network connection is established between the two buffers, the COM-softwares will not be informed if interruption occurs in the internet network. It is therefore a common practice in TCP applications to include a keep-alive data to check the status of the internet connection between the buffers. Utilizing the keep-alive data, the Powertrain-COM will throttle down the engine if it detects a severe network interruption to ensure the engine is at a suitable operating point before shutting down for safety purposes and to avoid hardware damage.

When VISSIM simulation is completed, VISSIM-COM will close the network socket port and notify Powertrain-COM to throttle down the engine for hardware shutdown.

4.5.3 Data Synchronization

The process of communication at every time-step is depicted in Figure 4.5. Fixed time-steps are maintained on both VISSIM and the powertrain research platform by VISSIM-COM and Powertrain-COM respectively to ensure data synchronicity. Data transfer has to be maintained at every time-step to ensure a real-time execution. At the same time, the smallest possible time-step needs to be maintained by both VISSIM-COM and Powertrain-COM to avoid data aliasing for test accuracy. Note from Fig. 4.5, certain tasks need to be executed by both COM-softwares. The time durations to execute these tasks need to be considered when choosing the appropriate time-step to be maintained by both COM-softwares. For VISSIM-COM, the tasks to be completed in a single time-step are running a single VISSIM simulation time-step and extracting traffic data (using COM) from VISSIM, which takes about 5ms and 7ms respectively. Note however the times taken to perform the tasks are dependent on the performance of the computer running VISSIM and VISSIM-COM, in our case an Intel® i5 with 2.5GHz processor and 8GB of RAM. Traffic data is then sent to the Powertrain-COM over internet network. The

transfer time takes less than 0.1ms using the University of Minnesota network at a distance of 2 miles between the COM-softwares. Once the traffic data is received, Powertrain-COM sends the data to the high-level controller in MATLAB-Simulink, which takes about 40ms. This latency is due to priority conflict with the executions of data plotting in the Graphical User Interface (GUI) used by the powertrain research platform controller. The data is then used to calculate the vehicle load for engine operating point optimization.

From the above, the total time to complete all tasks does not exceed 100ms. A time-step of 100ms may be used, but a 200ms time-step is instead chosen to take into consideration of other possible delays in future HiLS applications. For example, a more complicated engine optimization method may increase the processing time on the powertrain research platform side, while increased traffic data extraction will delay the VISSIM-COM side. Therefore, a wait-time is implemented in both Powertrain-COM and VISSIM-COM to maintain a time-step of 200ms. Note however that the processing times can also be shortened with faster computers. Figure 4.5 shows the HiLS data synchronization between the two COM-softwares.

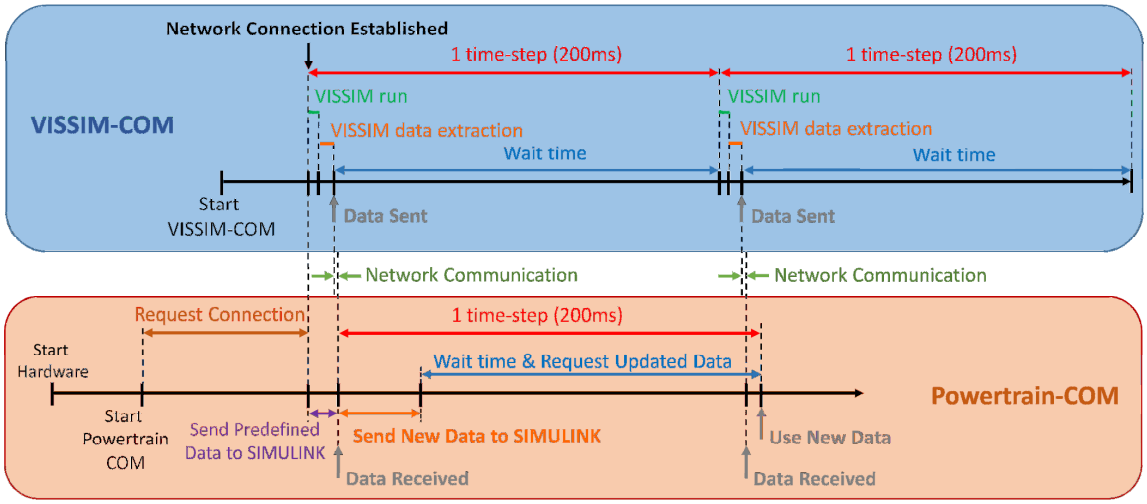


Figure 4.5. HiLS Data Synchronization

4.6 Experimental Results

Tests were conducted to show the capability of the HiLS. First the experimental setup is explained. Then the test results are presented and discussed. Finally, the conclusion of current works and future directions of the testbed are presented.

On the powertrain research platform, a Rule-Based optimization [65] is used in the high-level controller to optimize the engine operating points based on the vehicle load. The test engine is a 115 horsepower, turbocharged diesel engine, which is a representative of a small, lightweight diesel vehicle. The virtual powertrain is a hybrid electrical power-split with a planetary gear set, two electrical motors/generators and a battery as the main components. Details of the virtual powertrain model, control strategies and hardware setup are documented in the references [74-77].

4.6.1 Simple Traffic Network

A simple unidirectional 1700m traffic network with zero grade angle was built in VISSIM with 7 fixed-timing traffic-light intersections at every 200m from 300m to 1500m. VISSIM simulations are conducted beforehand to identify the target vehicle ID with the desired number of stops (no-stop, 1-stop, 2-stops and 3-stops) as shown in Figure 4.6.

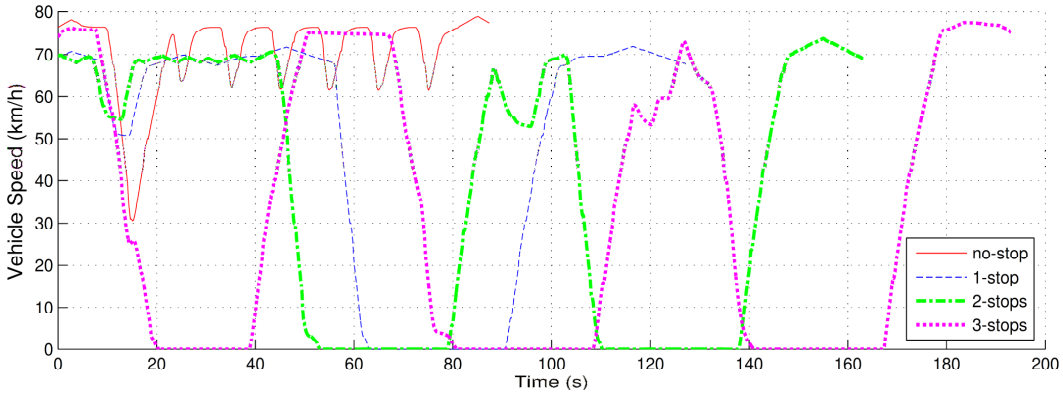


Figure 4.6. Vehicle Speeds for Simple Traffic Network

4.6.1a Vehicle Dynamics and Measurements of Fuel Consumption and Emissions

Figure 4.7 shows the reference vehicle speed that is received and tracked by the powertrain research platform, the tracked engine operating points, the dynamics of the virtual hybrid powertrain and the measured fuel consumption for the 3-stops case. Note

the torques and speeds of the generator and motor are generated from the virtual hybrid powertrain model. Figure 4.8 shows the measured emissions. Test results show accurate tracking of the desired vehicle speed, engine speed and engine torque with Mean Absolute Percentage Deviation (MAPD) of 0.4%, 2% and 14% respectively. The transient nature of actual engine torque is reflected in the relatively higher MAPD.

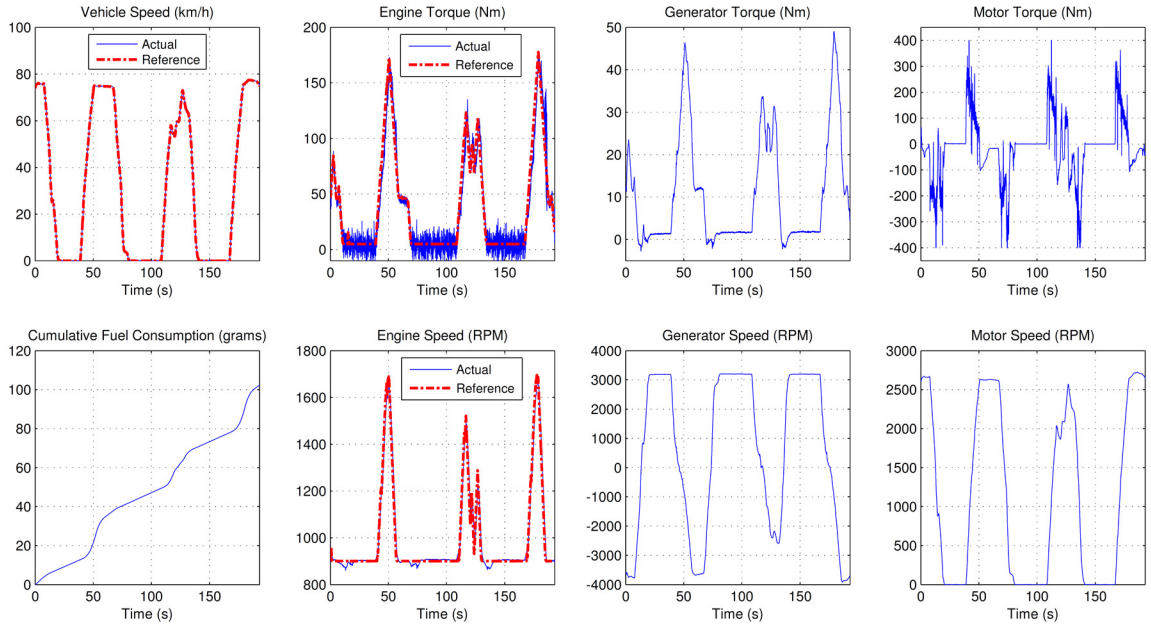


Figure 4.7. Vehicle-Powertrain Dynamics and Measured Fuel (3-stops, Simple Traffic Network)

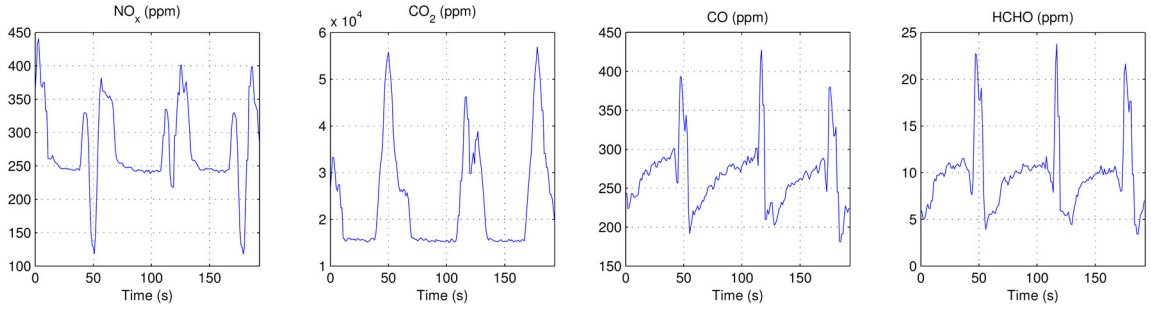


Figure 4.8. Measured Emissions (3-stops, Simple Traffic Network)

4.6.1b Total Fuel Consumptions

Measured fuel consumptions are shown in Figure 4.9. As the number of stops increases, the fuel consumption also increases due to high vehicle acceleration-deceleration and engine idling events that waste fuel and energy. The effect of driving behavior is shown

between the no-stop and 1-stop cases, where the benefit of having less stops is only about 5% due to an aggressive driving behavior of the no-stop case shown in Figure 4.6. The fuel use increases around 32-34% between 1-stop, 2-stops and 3-stops cases, all of which have similar level of driving aggressiveness.

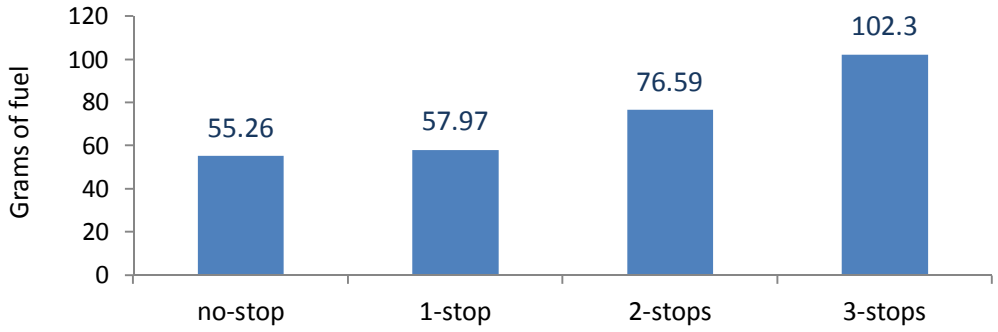


Figure 4.9. Total Mass of Measured Fuel Consumed for Simple Traffic Network

4.6.1c Total Mass of Measured Exhaust Gas Components

Each constituent of the exhaust gas is measured in terms of concentrations (ppm). Therefore, the formula below is used to convert the unit to mass-rate in grams per second.

$$\frac{g}{s} = \left(\frac{PPM}{10^6 moles\ of\ exhaust\ gas}\right) \times \left(\frac{exhaust\_mass\_rate}{exhaust\_molar\_mass}\right) \times constituent\_molar\_mass \qquad (4.1)$$

PPM reading of a constituent is its micromole concentration per mole of the exhaust gas. The exhaust gas mole-rate (in moles per second) is found from the exhaust mass-rate (in grams per second) and the exhaust molar-mass for diesel fuel (29.4 grams per mole). The exhaust mass-rate is the summation of the measured intake-air and fuel mass-rates. Multiplying the micromole concentration of the constituent with the exhaust mole-rate gives the mole-rate of the constituent. The mass-rate of the constituent is then found as the product of the mole-rate and the molar-mass of the constituent. Figure 4.10 shows the total mass of the measured exhaust gas components. It can be seen that as the number of stops increases, the amount of emissions also increases.

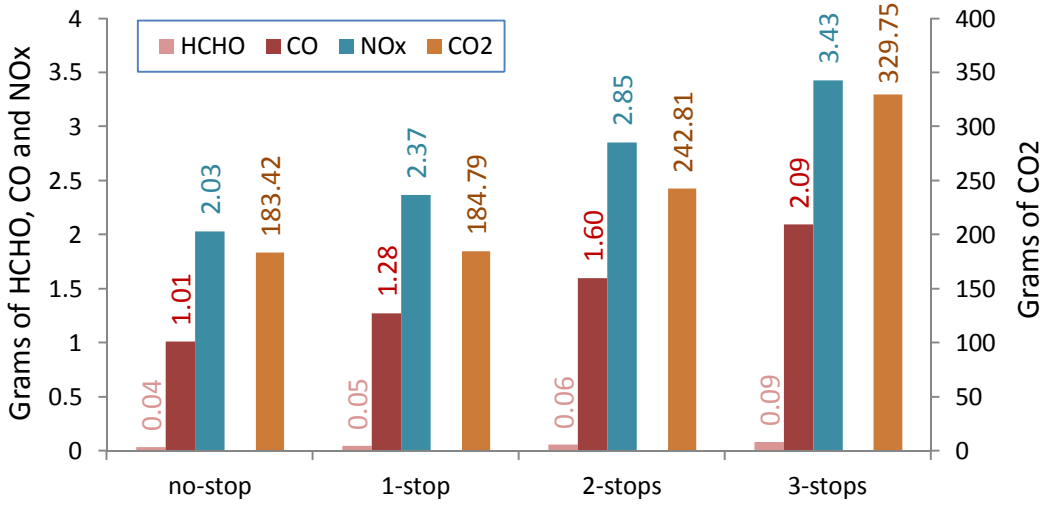


Figure 4.10. Total Mass of Measured Exhaust Gases for Simple Traffic Network

4.6.2 Complex Traffic Network

A complex traffic scenario was used from VISSIM's example demo BRT-priority-Texas. The main arterial network is based on a 3.5km stretch on Medical Drive between the intersections of Babcock Road and Fredericksburg Road in San Antonio, Texas as shown in Figure 4.11. The complexity of the network include (1) multiple vehicle types such as cars, busses and trucks (2) multiple lanes in each direction with vehicles lane-changing and bus stops (3) varying speed limits for roads and lanes (4) seven signalized intersections and six non-signalized intersections (5) reduced vehicle speeds, right-of-ways and pedestrian crossings at intersections (6) Ring-and-Barrier Controllers (RBC) and right-turn on red at signalized intersections (7) stop signs at non-signalized intersections.

The traffic simulation features are detailed enough that calibration works can be done to fit the parameters with real-world data. However, for the purpose of testing the testbed, the default parameters in the VISSIM demo are used. Two vehicles with 2-stops and 3-stops traversing the same path in Figure 4.11 are selected as test cases. Figure 4.12 shows the vehicle speed trajectories.

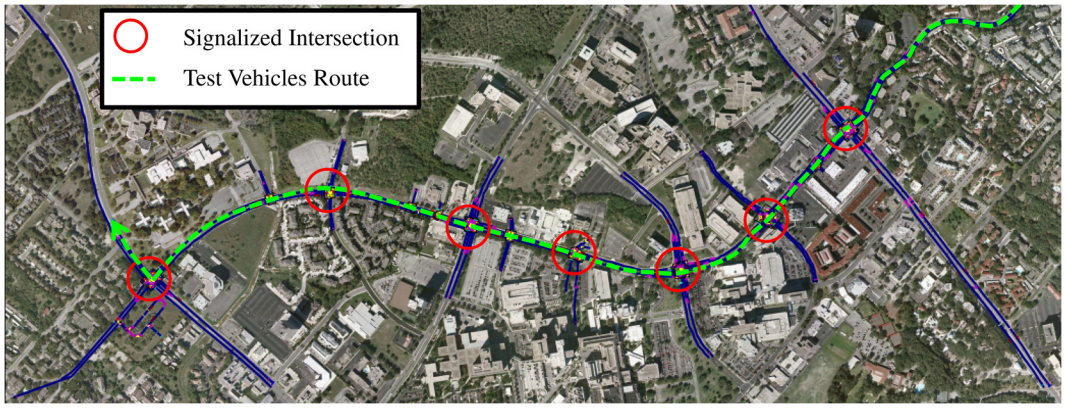


Figure 4.11. BRT-priority-Texas Traffic Demo from VISSIM

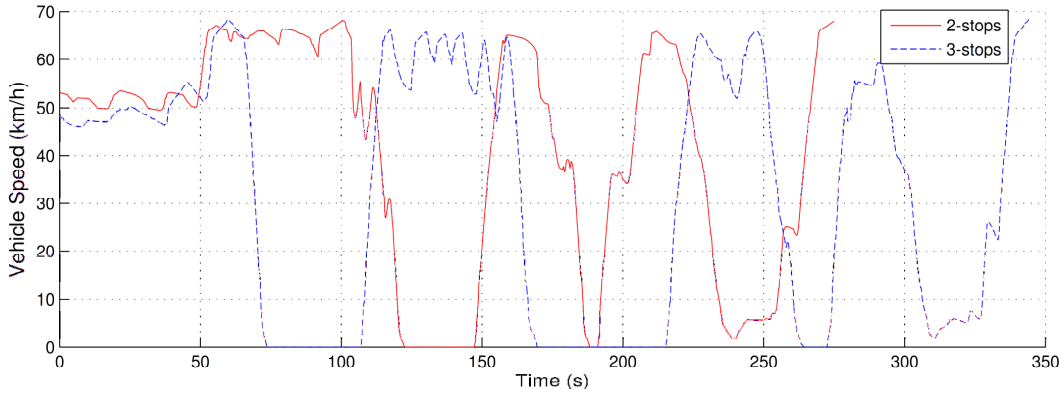


Figure 4.12. Vehicle Speeds for Complex Traffic Network

4.6.2a Vehicle Dynamics and Measurements of Fuel Consumption and Emissions

Figure 4.13 shows the vehicle, engine and virtual powertrain dynamics and the measured fuel of the 3-stops case in a complex traffic network with more transient vehicle dynamics than the simple case. The desired vehicle speed, engine torque and engine speed are tracked accurately with MAPD of 0.4%, 1.5% and 16% respectively. The engine torque MAPD is slightly higher than the simple traffic due to the higher dynamics of the reference. Emissions measurements are presented in Figure 4.14.

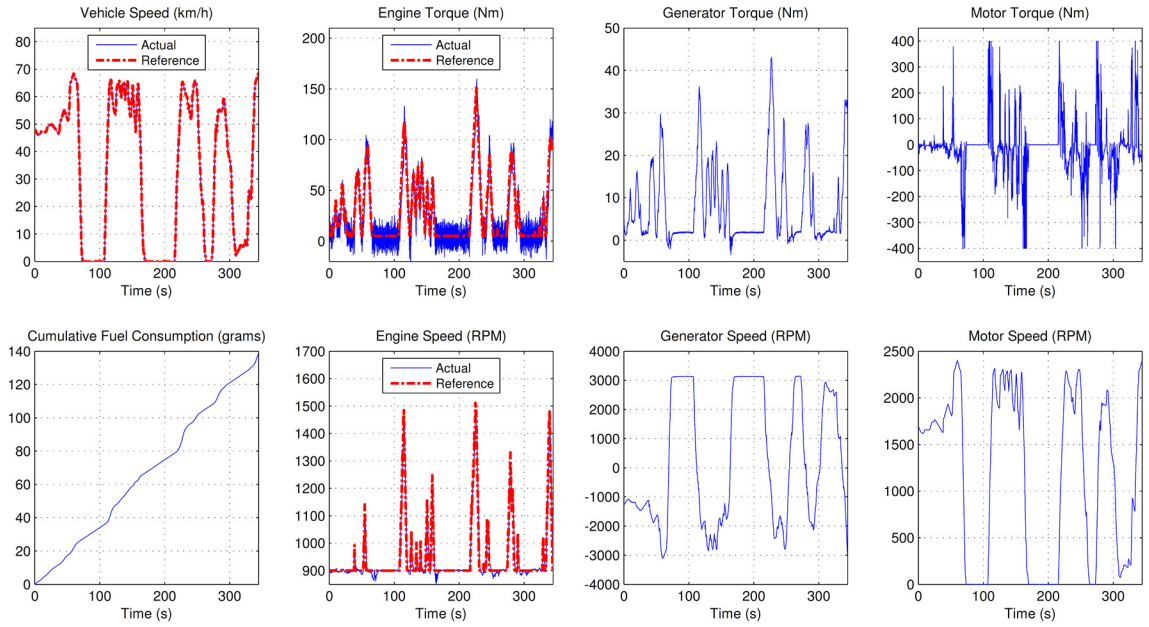


Figure 4.13. Vehicle-Powertrain Dynamics and Measured Fuel (3-stops, Complex Traffic Network)

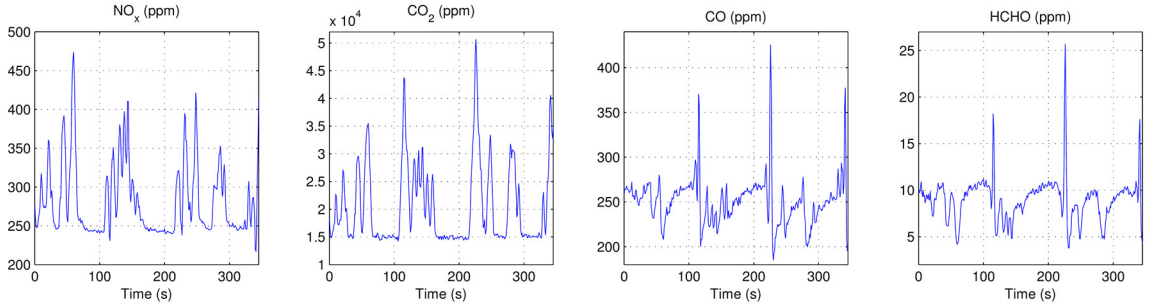


Figure 4.14. Measured Emissions (3-stops, Complex Traffic Network)

4.6.2b Total Fuel Consumptions

The total fuel consumptions measured for the 2-stops and 3-stops cases are shown in Figure 4.15. The 3-stops case consumes 23% more fuel than the 2-stops case, which is about 10% less than the comparison between the 2-stops and 3-stops cases in the simple traffic network. This can be attributed to the more transient vehicle dynamics in the complex traffic case, where the stop and go is not the only dominant dynamics.

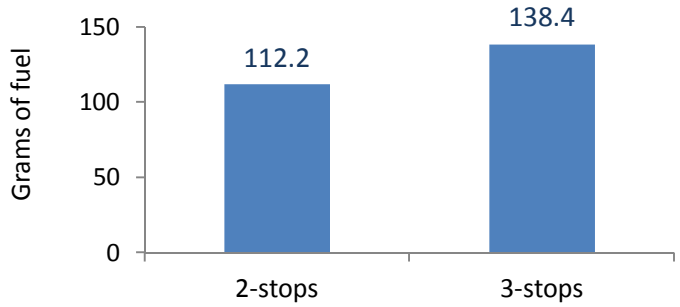


Figure 4.15. Total Mass of Measured Fuel Consumed for Complex Traffic Network

4.6.2c Total Mass of Measured Exhaust Gas Components

Figure 4.16 shows the total mass of exhaust gas components for the 2-stops and 3-stops cases in the complex traffic network. As shown, all exhaust gasses mass increase as the number of stops increases. Most notable are NO_x and CO_2 with 33% and 24% increases respectively in the 3-stops case.

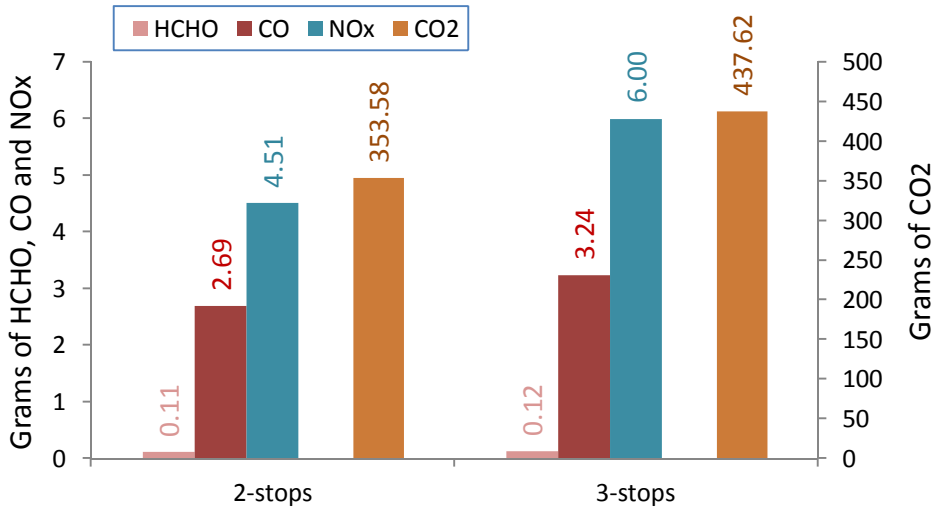


Figure 4.16. Total Mass of Measured Exhaust Gases for Complex Traffic Network

4.6.3 Comparison with Simulation Results

Simulations were performed to calculate total NO_X and CO_2 emissions using a software that estimates vehicle emissions on a policy level, but the results showed significant difference compared to measured emissions. This is attributed to the estimation errors from using averaged vehicles emissions data for a class of vehicles in the software, which is inaccurate to represent a specific vehicle. Furthermore, NO_X emission for the 2-

stops complex traffic case is interpolated using a well calibrated steady-state emission map for the specific test engine. Total NO_X from simulation was found to be 4.83 grams, which is 7% higher than the measured emission in Figure 4.16. This shows that even with a well calibrated map, simulation results are still inaccurate due to transient engine behaviors that cannot be captured by a steady-state map.

4.7 Conclusion

The HiLS has been tested and shown to achieve the followings:

1. Vehicle data from a remote traffic simulation have been successfully extracted and transferred in real-time to the powertrain research platform over the internet through COM interfaces and socket programming.

2. Different vehicle speed profiles are accurately tracked by the powertrain research platform to represent the target vehicle in VISSIM simulation.

3. Rule-Based optimization method has been successfully employed in the powertrain research platform to optimize engine operating points in real-time, which can be extended to other optimization methods utilizing traffic data [59] in the future.

4. Actual fuel consumption and emissions measurements are recorded, which can be used to evaluate various connected and autonomous vehicle applications.

Chapter 5

Conclusion and Future Work

5.1 Research Summary

The first objective of the thesis is to derive a computationally efficient energy management strategy for hybrid electrical vehicle that utilizes vehicle speed prediction to optimize fuel consumption while maintaining a battery charge-sustaining condition. Secondly, the energy management strategy has to also consider the operating range and transient dynamics limitation of an actual engine to ensure the optimized engine operating points are feasible. The final objective is to develop a Hardware-in-the-Loop System (HiLS) for evaluation of connected vehicle applications.

In Chapter 2, a combined approach of a time-efficient powertrain optimization strategy, utilizing vehicle speed trajectory prediction based on IVC and VII is proposed. A computationally efficient charge-sustaining (CS) HEV powertrain optimization strategy is analytically derived and simulated, based on the Pontryagin's Minimum Principle and a CS-condition constraint. A 3D lookup-map, generated offline to interpolate the optimizing parameters based on the predicted speed, is also utilized to speed up the calculations. Simulations are conducted for 6-mile and 15-mile cases with different prediction update timings to test the performance of the proposed strategy against a Rule-Based (RB) control strategy. The prediction, optimization and update structure is simulated and several conclusions can be made. First, the traffic model parameter calibration shows minimal improvements to prediction accuracy due to the error from the leading vehicle and has little effects on the MPG of the vehicles. Secondly, results show that a frequent update time does not necessarily means better MPG performance. If the vehicle trajectory prediction error at an update time does not improve from previous update time, the λ-error will be inherited, affecting MPG performance. This however is a limitation of

the IVC and VII network, where the vehicle interconnection with the whole prediction network depends on the vehicle's location. There will be certain durations where the information cannot be sent to the vehicle, causing the vehicle's prediction to be unchanging. Therefore, it would better to know when to resume or stop updating λ when a new prediction is received, depending on the prediction's accuracy. Thirdly, accurate prediction records an averaged 7% to 9.6% MPG's improvements over Rule-Based method, which could be significant when translated to a network level savings [71], but is modest for a single vehicle.

In Chapter 3, a fast charge-sustaining HEV optimization strategy that considers the operating range and transient dynamics of an engine is proposed. The HEV powertrain optimization problem is solved in two steps. First, given the flexibility of the power-split HEV powertrain, the relationship between the minimum fuel consumption and the power-split levels between the engine and battery is calculated and stored offline for all possible vehicle power demands. This relationship ensures the engine is operating at the most optimal point for every possible HEV power-split level for a given vehicle power demand. Then, the nonlinear fuel cost and battery dynamics are approximated by linear piecewise functions and formulated as a Separable Programming (SP) problem. The piecewise-linear functions introduce new dimensionless variables which are solved as a large-dimension constrained linear problem with efficient Linear Programming (LP) solvers. The engine operating range and engine transient dynamics are represented as linear constraints in the LP problem to ensure the engine operating points are feasible. Traffic information from CV is integrated in the optimization by integrating the driving cycle prediction into the powertrain optimization. Fast calculation time allows the optimization method to be implemented repeatedly in a CV environment with repeated speed prediction updates due to the transient nature of traffic. Fuel economy of SP with and without input-rate-constraint (SP-IRC and SP-NoIRC) are measured against Dynamic Programming (DP), Pontryagin's Minimum Principle (PMP) and Rule-Based (RB) methods using driving cycles generated in VISSIM traffic simulator. First, it is found that SP-NoIRC gives comparable fuel economy as DP for a short driving cycle. Then a fleet of 20 vehicles are simulated and optimization results are compared between SP-NoIRC, SP-IRC, PMP and RB. It is found that the SP-NoIRC achieves average improvements of 4.4% and 10.9% compared to PMP and RB respectively. SP-IRC on average is only 0.4% lower than SP-NoIRC due to engine dynamics averaging effects

from the input rate constraint. SP-IRC test results show the feasibility of the optimized engine operating points and the fuel benefits over RB method.

In Chapter 4, a Hardware-in-the-Loop-System (HiLS) is proposed to offer the flexibility and accuracy of evaluating the performance of connected vehicle applications. The HiLS is comprised of a microscopic traffic simulator (VISSIM) and a laboratory powertrain research platform. VISSIM is used to simulate a traffic network while the powertrain research platform, which consists of a real engine, an engine-loading device (hydrostatic dynamometer) and a virtual powertrain model is used to represent a single vehicle. A connected vehicle application such as the Cooperative Adaptive Cruise Control (CACC) can be simulated in VISSIM, where a target vehicle is selected to be represented by the powertrain research platform. This is done by sending the simulated target vehicle speed and road condition information from VISSIM to the powertrain research platform in real-time during simulation. This information is used to calculate the vehicle load demand, which is realized by the engine and powertrain. Fuel consumption and emissions from the engine are measured by precise laboratory equipment. The HiLS has been tested and shown to achieve the followings. First, vehicle data from a remote traffic simulation have been successfully extracted and transferred in real-time to the powertrain research platform over the internet through COM interfaces and socket programming. Secondly, different vehicle speed profiles are accurately tracked by the powertrain research platform to represent the target vehicle in VISSIM simulation. Thirdly, Rule-Based optimization method has been successfully employed in the powertrain research platform to optimize engine operating points in real-time, which can be extended to other optimization methods utilizing traffic data [59] in the future. Finally, actual fuel consumption and emissions measurements are recorded which can be used to evaluate various connected and autonomous vehicle applications.

5.2 Future Work

In Chapter 2, the impact of prediction errors to the optimization was found to be coming from the co-state λ calculation. To take into account the impact of prediction uncertainties on optimization performance, a more robust method to calculate λ is needed. One way to address this is to first quantify the speed prediction errors statistically [79]. Typically, a 95% confidence interval is used, indicating a 95% chance of occurrences between the selected bounds. Then, a robust optimization procedures that

considers the speed prediction uncertainties as bounded disturbances can be implemented.

In Chapter 3, despite having better performance, numerical separable programming optimization takes relatively longer to solve than analytical PMP solution. Therefore, as future works, reduced number of strategically placed sampling points to accurately characterize the fuel cost and battery state-of-charge can be used to reduce the number of dimensionless variables in the linear programming solver, hence the separable programming calculation time. Furthermore, vehicle speed prediction uncertainties are shown to deteriorate the performance of separable programming. Similar to Chapter 2, in this chapter a robust optimization strategy that can handle vehicle speed prediction uncertainties, while maintaining cost optimality, is needed to guarantee superior performance in a CV environment. Also as part of future works, the emissions for an engine can also be optimized together with the fuel using separable programming. First, the combined fuel and emissions cost function is normalized. Then, following the procedures in Chapter 3, the minimum emissions for different power split levels for every vehicle power demand are calculated offline using steady-state emissions maps [80]. With these done, the same procedures in Chapter 3 can be repeated to optimize the combined fuel and emission optimization problem using separable programming.

For both Chapter 2 and 3, the fuel optimization only considers the HEV powertrain. However, with the advent of connected vehicle technologies, the vehicle dynamics, which determines the vehicle load can also be optimized. There are two ways the combined optimization can be implemented. First is by implementing the optimizations in serial, which has been done for HEV powertrain in [81], where the vehicle dynamics is optimized first before optimizing the powertrain. Secondly, both the vehicle dynamics and powertrain can be optimized simultaneously, as implemented in [82] for an automatic gear transmission. Combined vehicle dynamics and powertrain optimization will increase the optimizing inputs or variables, thus increasing the problem dimension if it were to be solved numerically. This may pose a difficulty in achieving real time implementation, which however can be fixed by decreasing the prediction window. In this thesis, the prediction window is in the range of 700 to 800 seconds with 0.1 second time-step. If the prediction window is shortened to less than 100 seconds, the size of the

problem can be significantly reduced, allowing for a faster integrated vehicle dynamics and HEV powertrain optimization.

For the Hardware-in-the-Loop System, several improvements and additional functionalities are planned to improve the functionality of the testbed. First, a more comprehensive traffic simulations in VISSIM will be developed, which will be evaluated and calibrated with data collected from actual roads. For example, using live data from SMART-SIGNAL on Trunk Highway 55, the HiLS can be calibrated. Since SMART-SIGNAL logs data nonstop, different traffic scenarios can also be used for calibration purposes. The information such as time-stamped signal phase change, vehicle detector actuation, queue length, delay, arterial travel time and turning movement count are stored in a SQL database which can be accessed online [83].

For added functionality, the HiLS will include multiple Road-Side-Units (RSU) which are traffic data collection devices instrumented along real roads. Using the RSU, traffic data such as signal phase and vehicle detector count can be accessed and transferred to the VISSIM traffic simulator in real-time to emulate real traffic. In addition, the RSU will also be paired with On-Board-Units (OBU) which are installed in on-road vehicles. OBU can broadcast vehicle dynamics data to the RSU, which can then send the data to the HiLS. The real vehicle dynamics can then be used to represent a vehicle in VISSIM simulation, for example as the preceding vehicle to the vehicle represented by the powertrain research platform. With RSU and OBU devices, communication between vehicles (OBU to OBU) and between vehicle (OBU) and infrastructure (RSU) will also be realistic. Overall, with calibrated traffic parameters, real-time traffic data and real vehicle dynamics, the HiLS can be used to evaluate different connected vehicle applications realistically. An external connected vehicle controller is linked with the HiLS to control selected vehicles in VISSIM simulation, as shown in Appendix 3. Finally, with the HiLS, connected vehicle applications, such as eco-approach, cooperative adaptive cruise control, eco-driving and speed harmonization can be tested. Fuel and emissions can be measured in lab, where the benefits for each connected vehicle application can be assessed realistically using a real engine.

Bibliography

[1] EIA, 2010. “Annual Energy Outlook 2010 with Projections to 2035”, DOE/EIA-0383(2010). Energy Information Administration.

[2] Schrank, D., Lomax, T. and Eisele, B., 2011. “TTI’s 2011 Urban Mobility Report”, Texas Transportation Institute, Sept.

[3] NHTSA, 2012. DOT and EPA Propose CAFÉ and GHG Emission Standards, 2017-2025: Fact Sheet. National Highway Traffic Safety Administration. <http:// www.nhtsa.gov/staticfiles/rulemaking/pdf/cafe/2017-25_CAFE_NPRM_Factsheet .pdf> (accessed on 30.01.13).

[4] Shinichi, A., 2000. “Development of the Hybrid Vehicle and its Future Expectation”, SAE 2000-01-C042.

[5] Muta, K., Yamazaki, M. and Tokieda, J., 2004. “Development of New-Generation Hybrid System THS II – Drastic Improvement of Power Performance and Fuel Economy”, SAE Technical Paper 2004-01-0064.

[6] Emadi et al, 2005. “Topological Overview of Hybrid Electric and Fuel Cell Vehicular Power System Architectures and Configurations”, IEEE Transaction on Vehicular Technology v54, n3, pp. 763–770.

[7] Ma, Y., Chowdhury, M., Sadek, A., and Jeihani, M., 2009. “Real-Time Highway Traffic Condition Assessment Framework Using Vehicle–Infrastructure Integration With Artificial Intelligence”, IEEE Transactions on Intelligent Transportation Systems, v10, n4, pp. 615-627.

[8] Paikari, E., Tahmasseby, S., and Far, B., 2014. “A Simulation-based Benefit Analysis of Deploying Connected Vehicles using Dedicated Short Range Communication”, Proceedings of the 2014 IEEE Intelligent Vehicles Symposium, pp. 980-985.

[9] Bai, F., and Krishnan, H., 2006. “Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications”, 2006 IEEE Intelligent Transportation System Conference, pp. 355-362.

[10] Chen, X., Refai, H.H., and Ma, X., 2007. “A Quantitative Approach to Evaluate DSRC Highway Inter-Vehicle Safety Communication”, 2007 IEEE Global Telecommunications Conference, pp. 151-155.

[11] Kenney, J.B., 2011. “Dedicated Short-Range Communications (DSRC) Standards in the United States”, Proceedings of the IEEE, v99, n7, pp. 1162-1182.

[12] Dopart, K., 2014. Connected Vehicle Safety Pilot Program. U.S Department of Transport, http://www.its.dot.gov/factsheets/safety_pilot_factsheet.htm. Accessed Sept 5, 2014.

[13] Sepulcre, M., Gozalvez, J., and Hernandez, J., 2013. “Cooperative Vehicle-to-Vehicle Active Safety Testing Under Challenging Conditions”, Transportation Research Part C: Emerging Technologies, v26, pp. 233-255.

[14] Zhang et al., 2010. “Role of Terrain Preview in Energy Management of Hybrid Electric Vehicles”, IEEE Transactions on Vehicular Technology, v59, n3, March.

[15] Onori, S., Serrao, L. and Rizzoni, G., 2010. “Adaptive Equivalent Consumption Minimization Strategy for HEVs”, Proceedings of the ASME 2010 Dynamic Systems and Control Conference, v1, Sept, pp. 499-505.

[16] Musardo, C., Rizzoni, G. and Staccia, B., 2005. “A-ECMS: An Adaptive Algorithm for HEV Energy Management”, Proceedings of the 44th IEEE CDC-ECC, Dec, pp. 1816- 1823.

[17] Liu, J. and Peng, H., 2008. “Modeling and Control of a Power-Split Hybrid Vehicle”, IEEE Transactions on Control Sys.Tech., v16, n6, Nov.

[18] Morari, M., and Lee, J.H., 1999. “Model Predictive Control: Past, Present & Future”, Computers & Chemical Engineering, v23, n4–5, pp. 667-682.

[19] Nevistic´, V. and Morari, M., 1995. “Constrained Control of Feedback Linearizable Systems”, Proceedings of the 1995 European Control Conference, Rome, Italy, pp. 1726–1731.

[20] Borhan et al., 2009. “Predictive Energy Management of a Power Split Hybrid Electric Vehicle”, Proceedings of the 2009 American Control Conference, Hyatt Regency Riverfront, St. Louis, MO, June.

[21] Zhang, C., and Vahidi, A., 2010. “Real-Time Optimal Control of Plug-In Hybrid Vehicles with Trip Preview”, Proceedings of the 2010 American Control Conference, Baltimore, MD, pp. 6917-6922.

[22] Serrao, L., Onori, S. and Rizzoni, G., 2009. “ECMS as a Realization of Pontryagin’s Minimum Principle for HEV Control”, Proceedings of the 2009 American Control Conference, June, pp. 3964-3969.

[23] Kim, N., Cha, S. and Peng, H., 2011. “Optimal Control of HEVs based on Pontryagin’s Min. Principle”, IEEE Trans. on Control Sys. Technology, Sept, v19, n5, pp. 1279-1287.

[24] He et al., 2012. “Merging Mobility and Energy Vision with HEV and Vehicle Infrastructure Integration”, Energy Policy, v41, Feb, pp. 599-609.

[25] He et al., 2012. “An Energy Optimization Strategy for Power-Split Drivetrain PHEVs”, Transportation Research Part C: Emerging Technologies, v22, June, pp. 29-41.

[26] Tate, E., and Boyd, S., 2000. “Finding Ultimate Limits of Performance for Hybrid Electric Vehicles”, SAE Technical Paper 2000-01-3099.

[27] Steinmauer, G., and del Re, L., 2001. “Optimal Control of Dual Power Sources”, Proceedings of the 2001 IEEE International Conference on Control Applications, 2001, Mexico City, pp. 422-427.

[28] Hahn, S., Waschl, H., Steinmaurer, G., and del Re, L., 2015. “Extension of a Linear Optimal Control Strategy for HEV”, 2015 European Control Conference, Linz, pp. 154-159.

[29] Wu, G., Boriboonsomsin, K., and Barth, M.J., 2014. “Development and Evaluation of an Intelligent Energy-Management Strategy for Plug-in Hybrid Electric Vehicles”, IEEE Transactions on Intelligent Transportation Systems, v15, n3, pp. 1091-1100.

[30] He et al., 2012. “Forward Power-Train Energy Management Modeling for Assessing Benefits of Integrating Predictive Traffic Data into PHEVs”, Transp. Research Part D: Transport & Environment, v17, pp. 201-207.

[31] Manzie, C., Watson, H. and Halgamuge, S., 2007. “Fuel Economy Improvements for Urban Driving: Hybrid vs. Intelligent Vehicles”, Transp. Research Part C: Emerging Technologies, v15, n1, Feb, pp. 1-16.

[32] Ranjan, N and Li, Y., 2011. “Trip Based Stochastic Prediction of Battery State of Charge for Electric Vehicles”, SAE Tech. Paper, 2011-01-1363.

[33] Filipi, Z., Fathy, H., Hagena, J., Knafl, A., Ahlawat, R., Liu, J., Jung, D., Assanis, D., Peng, H., and Stein, J., 2006. “Engine-in-the-Loop Testing for Evaluating

Hybrid Propulsion Concepts and Transient Emissions - HMMWV Case Study", SAE Technical Paper 2006-01-0443.

[34] Hagena, J., Filipi, Z., and Assanis, D., 2006. "Transient Diesel Emissions: Analysis of Engine Operation During a Tip-In", SAE Technical Paper 2006-01-1151.

[35] Patterson, M., Kong, S., Hampson, G., and Reitz, R., 1994. "Modeling the Effects of Fuel Injection Characteristics on Diesel Engine Soot and NOX Emissions", SAE Technical Paper 940523.

[36] Hong, S., Assanis, D., and Wooldridge, M., 2002. "Multi-Dimensional Modeling of NO and Soot Emissions with Detailed Chemistry and Mixing in a Direct Injection Natural Gas Engine", SAE Technical Paper 2002-01-1112.

[37] Liu, B., and Frey, H.C., 2015. "Variability in Light-Duty Gasoline Vehicle Emission Factors from Trip-Based Real-World Measurements", Environmental Sciences & Technology, v49, n20, pp.12525-12534.

[38] Zhao, X., Wu, Y., Rong, J., and Zhang, Y., 2015. "Development of a Driving Simulator Based Eco-Driving Support System", Transportation Research Part C: Emerging Technologies, v58, pp. 631-641.

[39] Hou, Y., Zhao, Y., Hulme, K.F., Huang, S., Yang, Y., Sadek, A.W., and Qiao, C., 2014. "An Integrated Traffic-Driving Simulation Framework: Design, Implementation, and Validation", Transportation Research Part C: Emerging Technologies, v45, pp. 138-153.

[40] He, X., Liu, H.X., and Liu, X., 2015. "Optimal Vehicle Speed Trajectory on a Signalized Arterial with Consideration of Queue", Transportation Research Part C: Emerging Technologies, v61, pp. 106-120.

[41] Bhavsar, P., Chowdhury, M., He, Y., and Rahman, M., 2014. "A Network Wide Simulation Strategy of Alternative Fuel Vehicles", Transportation Research Part C: Emerging Technologies, v40, pp. 201-214.

[42] Schroth, C., Dötzer, F., Kosch, T., Ostermaier, B., and Strassberger, M., 2005. "Simulating the Traffic Effects of Vehicle-to-Vehicle Messaging Systems", Proceedings of the 5th International Conference on ITS Telecommunications.

[43] Ma, Y., Chowdhury, M., Sadek, A., and Jeihani, M., 2012. "Integrated Traffic and Communication Performance Evaluation of an Intelligent Vehicle Infrastructure

Integration (VII) System for Online Travel-Time Prediction", IEEE Trans. Intel. Transport. System, v13, n3, pp. 1369–1382.

[44] Bullock, D., and Catarella, A., 1998 "A Real-Time Simulation Environment for Evaluating Traffic Signal Systems", Transportation Research Record: Journal of the Transportation Research Board, v1634, pp. 130-135.

[45] Ma, W., 2008. "A Real-Time Performance Measurement System for Arterial Traffic Signals". Diss. University of Minnesota, Twin Cities.

[46] Wipke, K.B., Cuddy, M.R., and Burch, S.D., 1999. "ADVISOR 2.1: A User-Friendly Advanced Powertrain Simulation Using a Combined Backward/Forward Approach", IEEE Transactions on Vehicular Technology, v48, n6, pp. 1751-1761.

[47] Senger, R.D., Merkle, M.A., and Nelson, D.J., 1998. "Validation of ADVISOR as a Simulation Tool for a Series Hybrid Electric Vehicle", Technology for Electric and Hybrid Vehicles, Proceedings of the 1998 SAE International Congress, Detroit, MI, pp. 95-115.

[48] Manjunatha P., Vortisch P., and Mathew T., 2013. "Methodology for the Calibration of VISSIM in Mixed Traffic", Proceedings of the Transportation Research Board 92nd Annual Meeting, No. 13-3677.

[49] Mensing, F., Bideaux, E., Trigui, R., Ribet, J., and Jeanneret, B., 2014. "Eco-Driving: An Economic or Ecologic Driving Style?", Transportation Research Part C: Emerging Technologies, v38, pp. 110-121.

[50] Duoba, M., Ng, H., and Larsen, R., 2000. "In-Situ Mapping and Analysis of the Toyota Prius HEV Engine", SAE Technical Paper 2000-01-3096.

[51] Hu, H., Zou, Z., and Yang, H., 2009. "On-board Measurements of City Buses with Hybrid Electric Powertrain, Conventional Diesel and LPG Engines", SAE Technical Paper 2009-01-2719.

[52] Takada, Y., Ueki, S., and Saito, A., 2004. "Study on Fuel Economy and NOX Emissions of Medium Duty Hybrid Truck in Real Traffic Conditions", SAE Technical Paper 2004-01-1086.

[53] Norbeck, J.M., Miller, J.W., Welch, W.A., Smith, M., Johnson, K., and Pankratz, D., 2001. "Final Report: Develop On-Road System for Emissions Measurement from Heavy-Duty Trucks", South Coast Air Quality Management District Contract 20906.

[54] Daham, B., Andrews, G.E., Li, H., Ballesteros, R., Bell, M.C., Tate, J., and Ropkins, K., 2005. “Application of a Portable FTIR for Measuring On-road Emissions”, SAE Technical Paper 2005-01-0676.

[55] Li, H., Ropkins, K., Andrews, G.E, Daham, B., Bell, M., Tate, J., and Hawley, G., 2006. “Evaluation of a FTIR Emission Measurement System for Legislated Emissions Using a SI Car”, SAE Technical Paper 2006-01-3368.

[56] Hall, R.W., and Tsao, H.-S.J., 1997. Automated Highway System Deployment: A Preliminary Assessment of Uncertainties. In Petros A. Ioannou (Ed.), Automated Highway Systems (pp. 325-334). New York, NY: Springer US.

[57] Thorpe, C., Jochem, T., and Pomerleau, D., 1998. “Automated Highways and the Free Agent Demonstration”, Robotics Research, 8th International Symposium, pp. 246-254.

[58] Mohd Zulkefli, M.A., Zheng, J., Sun, Z. and Liu, H.X., 2013. “Hybrid Powertrain Optimization with Real-Time Traffic Information”, Proceedings of the 2013 ASME Dynamic Systems and Control Conference, Palo Alto, CA, DSCC2013-3919, Oct.

[59] Mohd Zulkefli, M.A., Zheng, J., Sun, Z. and Liu, H.X., 2014. “Hybrid Powertrain Optimization with Trajectory Prediction based on Inter-Vehicle-Communication and Vehicle-Infrastructure-Integration”, Transportation Research Part C: Emerging Technologies, v45, pp. 41-63.

[60] Mohd Zulkefli, M.A., and Sun, Z., 2016. “Real-Time Powertrain Optimization Strategy for Connected Hybrid Electrical Vehicle”, Proceedings of the 2016 ASME Dynamic Systems and Control Conference, Minneapolis, MN, DSCC2016-9727, Oct.

[61] Mohd Zulkefli, M.A., Mukherjee, P., Sun, Z., Zheng, J., Liu, H.X. and Huang, P., 2017. “Hardware-in-the-loop Testbed for Evaluating Connected Vehicle Applications”, Transportation Research Part C: Emerging Technologies, v78, pp. 50–62.

[62] Mohd Zulkefli, A., Mukherjee, P., Shao, Y. and Sun, Z., 2016. “Evaluating Connected Vehicles and Their Applications”, ASME Dynamic Systems and Control Magazine, v4, n4, pp.12-17, Dec.

[63] Liu, J., Peng, H., Filipi, Z., 2005. “Modeling and Control Analysis of the Toyota Hybrid System”, Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronic, July, pp. 134–139.

[64] Ben-Asher, J.Z., 2010. “Optimal Control Theory With Aerospace Applications”, AIAA Education Series, American Institute of Aeronautics, Print.

[65] Besnier, F. and Rousseau, A., 2002. “Comparison of Toyota US and Japan Prius”, Transportation Tech. R&D Center, Argonne National Lab, IL, May.

[66] Duoba, M., Lohse-Busch, H., Carlson, R., Bohn, T. et al., 2007. “Analysis of Power-Split HEV Control Strategies Using Data from Several Vehicles”, SAE Technical Paper 2007-01-0291.

[67] Wang, W. 2002. “Revision on the Model of Toyota Prius in ADVISOR 3.1”, SAE Technical Paper 2002-01-0993.

[68] Liu, J., 2007. “Modeling, Configuration and Control Optimization of Power-split Hybrid Vehicles”. Diss. University of Michigan, Ann Arbor.Web.24 Oct 2012.

[69] NREL. 2003.“ADVISOR 2003”.Computer Software. Big Ladder Soft., Win. 7.

[70] Wiedemann, R., 1974. “Simulation des Strassenverkehrsflusses”. Institute for Traffic Engineering, University of Karlsruhe, Band, Karlsruhe, Germany.

[71] Murphy, Y.L., Park, J., Chen, Z., Kuang, M.L. and Masrur, M.A., 2012. “Intelligent Hybrid Power Control – Part I : Machine Learning of Optimal Vehicle Power”, IEEE Transactions on Vehicular Technology, v61, n8, Oct, pp. 3519-3530.

[72] TOYOTA Technical Training: Toyota Hybrid System - Course 071: Section 3 High-Voltage Battery. Torrance, CA: Toyota Motor Sales, 2006. Print.

[73] Bradley, S.P., Hax, A.C., and Magnanti, T.L. Applied Mathematical Programming. Addison-Wesley, 1977. Print.

[74] Wang, Y., Sun, Z., and Stelson, K.A., 2011. “Modeling, Control, and Experimental Validation of a Transient Hydrostatic Dynamometer”, IEEE Transactions on Control Systems Technology, v19, n6, pp. 1578-1586.

[75] Wang, Y., and Sun, Z., 2010. “A Hydrostatic Dynamometer Based Hybrid Powertrain Research Platform”, Proceedings of the 2010 International Symposium on Flexible Automation.

[76] Wang, Y., Song, X., and Sun, Z., 2011. “Hybrid Powertrain Control with a Rapid Prototyping Research Platform”, Proceedings of the American Control Conference 2011, pp. 997-1002.

[77] Wang, Y., and Sun, Z., 2015. “Dynamic Analysis and Multivariable Transient Control of the Power-Split Hybrid Powertrain”, IEEE/ASME Transactions on Mechatronics, v20, n6, pp 3085-3097.

[78] Box, D., 1998. Essential COM. Reading, MA: Addison-Wesley.

[79] Otoshi, T., Oshita, Y., Murata, M. et al., 2013. “Traffic Prediction for Dynamic Traffic Engineering Considering Traffic Variation”, 2013 IEEE Global Communications Conference (GLOBECOM), pp. 1570-1576, Dec.

[80] Wang, Y., 2014. “Design, Control and Energy Optimization of a Rapid-Prototyping Hybrid Powertrain Research Platform”. Diss. University of Minnesota.

[81] Hu, J., Shao, Y., Sun, Z., Wang, M., Bared, J., and Huang, P., 2016. “Integrated Optimal Eco-Driving on Rolling Terrain for Hybrid Electric Vehicle with Vehicle-Infrastructure Communication”, Transportation Research Part C: Emerging Technologies, v68, pp. 228-244.

[82] Hu, J., Shao, Y., Sun, Z., and Bared, J., 2017. “Integrated Vehicle and Powertrain Optimization for Passenger Vehicles with Vehicle-Infrastructure Communication”, Transp. Research Part C: Emerging Tech., v79, pp. 85-102.

[83] iMonitor™ / iMeasure™, SMART Signal Technologies Inc.©, September 2014. Web. May 2013.<http://dotapp7.dot.state.mn.us/smartsignal/>.

[84] Mukherjee, P., 2016. “Investigation of Cooperative Adaptive Cruise Control with Experimental Validation”, MS Thesis, University of Minnesota.

Appendix 1: Overview of Pontryagin's Minimum Principle

Discussions below can be found in [64]. Consider an unconstrained optimization with open final time. The augmented cost function is given by

$$I = \phi(x_f, t_f, v) + \int_{t_0}^{t_f} L(x,u,t)dt \tag{1a}$$

$L(x,u,t)$ is the cost function and $\phi(x_f, t_f, v)$ is the augmented terminal cost given by

$$\phi(x_f, t_f, v) = \phi(x_f, t_f) + v^T \psi(x_f, t_f) \tag{2a}$$

where $\phi(x_f, t_f)$ = soft terminal constraint

$v^T \psi(x_f, t_f)$ = hard terminal constraint

Where v^T is the Lagrangian multiplier that drives the terminal condition $\psi(x_f, t_f)$ to zero.In this case, v^T is constant because its responsible to drive only the final state to zero. The states dynamics are given by

$$\dot{x} = f(x,u,t) \quad \text{with initial conditions} \quad x(t_0) = x_0 \tag{3a}$$

The Hamiltonianis given by

$$H = L(x,u,t) + \lambda^T f(x,u,t) \tag{4a}$$

where λ^T = co-states

The necessary conditions are derived here as bases to the implementations in powertrain optimization in the next section. The augmented cost function in Eq. (1a) can thus be re-written as

$$I = \phi(x_f, t_f, v) + \int_{t_0}^{t_f} \{L(x,u,t) + \lambda^T [f(x,u,t) - \dot{x}]\} dt$$

$$= \phi(x_f, t_f, v) + \int_{t_0}^{t_f} (H - \lambda^T \dot{x}) dt \tag{5a}$$

We define the candidate signals as :

$u^*(t)$ = optimal input $x^*(t)$ = optimal states t_f^* = optimal final time

Time-fixed input variation and its corresponding states variation due to input are given as

$$u(t) = u^*(t) + \delta u(t) \qquad x(t) = x^*(t) + \delta x(t)$$

while final time variations is given by : $t_f = t_f^* + dt_f$

Introducing 1st order of variation in the augmented cost function in Eq. (5a)

$$\begin{aligned} dI &= \phi_{x_f} dx_f + \phi_{t_f} dt_f + \phi_v dv + \int_{t_0}^{t_f} (H_x \delta x + H_u \delta u + H_\lambda \delta\lambda - \lambda^T \delta\dot{x} - \delta\lambda^T \dot{x}) dt \\ &\quad + (H - \lambda^T \dot{x})|_{t_f} dt_f \\ &= \phi_{x_f} dx_f + \phi_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} (H_x \delta x + H_u \delta u + \dot{x}^T \delta\lambda - \lambda^T \delta\dot{x} - \delta\lambda^T \dot{x}) dt \\ &\quad + (H - \lambda^T \dot{x})|_{t_f} dt_f \\ &= \phi_{x_f} dx_f + \phi_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} (H_x \delta x + H_u \delta u) dt \\ &\quad - \textstyle\int_{t_0}^{t_f} (\lambda^T \delta\dot{x}) dt + (H - \lambda^T \dot{x})|_{t_f} dt_f \end{aligned} \tag{6a}$$

Using integration by parts, the second integral term in Eq. (6a) can be rewritten as :

$$\begin{aligned} -\int_{t_0}^{t_f} (\lambda^T \delta\dot{x}) dt &= -\lambda^T \delta x|_{t_0}^{t_f} + \int_{t_0}^{t_f} \left(\dot{\lambda}^T \delta x\right) dt \\ &= -\lambda^T(t_f)\delta x(t_f) + \lambda^T(t_0)\delta x(t_0) + \int_{t_0}^{t_f} \left(\dot{\lambda}^T \delta x\right) dt \\ &= -\lambda^T(t_f)\delta x(t_f) + \textstyle\int_{t_0}^{t_f} \left(\dot{\lambda}^T \delta x\right) dt \end{aligned} \tag{7a}$$

Since the initial states are normally known, the variations $\delta x(t_0)$ of the initial states are assumed to be zero because. Replacing Eq. (7a) into Eq. (6a) gives

$$\begin{aligned} &= \phi_{x_f} dx_f + \phi_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} \left[\left(H_x + \dot{\lambda}^T\right)\delta x + H_u \delta u\right] dt - \lambda^T(t_f)\delta x(t_f) \\ &+ (H - \lambda^T \dot{x})|_{t_f} dt_f \end{aligned} \tag{8a}$$

Using linear approximation (Fig. A1.1), $\delta x(t_f)$ is approximated as

$$\delta x(t_f) = dx_f - \dot{x}_f dt_f \tag{9a}$$

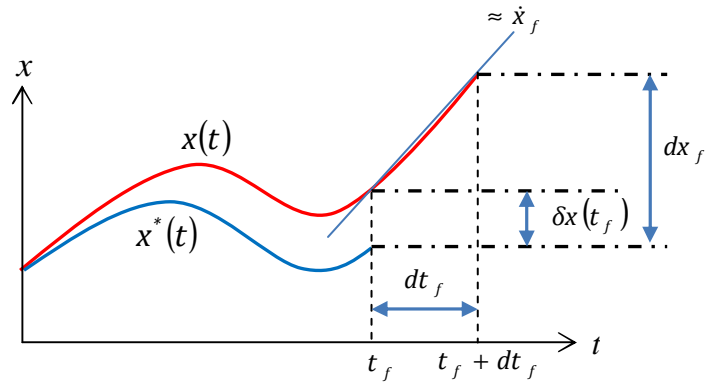


Figure A1.1. Linear Approximation of $\delta x(t_f)$

Replacing Eq. (9a) into Eq. (8a)

$$dI = \phi_{x_f} dx_f + \phi_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} \left[(H_x + \dot{\lambda}^T)\delta x + H_u \delta u\right] dt - \lambda^T(t_f)(dx_f - \dot{x}_f dt_f) + (H - \lambda^T \dot{x})|_{t_f} dt_f$$

$$= \left[\phi_{x_f} - \lambda^T(t_f)\right] dx_f + \phi_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} \left[(H_x + \dot{\lambda}^T)\delta x + H_u \delta u\right] dt + \lambda^T(t_f)\dot{x}_f dt_f + (H - \lambda^T \dot{x})|_{t_f} dt_f$$

$$= \left[\phi_{x_f} - \lambda^T(t_f)\right] dx_f + (\phi_t + H)|_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} \left[(H_x + \dot{\lambda}^T)\delta x + H_u \delta u\right] dt + \left[\lambda^T(t_f)\dot{x}_f - (\lambda^T \dot{x})|_{t_f}\right] dt_f$$

$$= \left[\phi_{x_f} - \lambda^T(t_f)\right] dx_f + (\phi_t + H)|_{t_f} dt_f + \psi^T dv + \int_{t_0}^{t_f} \left[(H_x + \dot{\lambda}^T)\delta x + H_u \delta u\right] dt \quad (10a)$$

From Eq. (10a), in order to ensure $dI = 0$, the following equations have to be met :

1) Adjoint Equations : $\dot{\lambda} = -H_x^T$ and $\phi_{x_f} = \lambda^T(t_f)$ (11a)

2) Controls Equation : $H_u = 0$ (12a)

3) System Equations : $\dot{x} = f(x, u, t)$ and $x(t_0) = x_0$ (13a)

4) Terminal Conditions: $\psi(x_f, t_f) = 0$ (14a)

5) Transversality Cond: $(\phi_t + H)|_{t_f} = 0$ (15a)

Appendix 2: Rule-Based HEV Powertrain Optimization

Rule-Based Engine Operating Points

The first controller combines the power request from the wheels and the virtual power request by the electrical machines and finds the minimum engine operating points for that combined power request [65-67]. Assuming the vehicle velocity and acceleration are known, the torque request from the vehicle can be found from Eq. (2.1) of Chapter 2.3 and the power request is calculated from

$$P_{wheel} = T_v \omega_v \tag{1b}$$

The virtual power request by the electrical machines, P_{SOC}, can be found from the following equations, based on ADVISOR 2003 Simulink Model [69].

$$P_{SOC} = \left(SOC_{target} - SOC\right) K_{fit} V_{batt} \eta_{gm}^{-1} \tag{2b}$$

Where (SOC_{target} – SOC) term is used to maintain the SOC around SOC_{target}, η_{gm} is the efficiency of the electrical machines (assumed constant), K_{fit} is the fitting constant (to fit with experimental results) and V_{batt} is given by

$$V_{batt} = V_{oc} - I_{batt} R_{batt} \tag{3b}$$

$$I_{batt} = \frac{V_{oc} - \sqrt{V_{oc}^2 - 4 R_{batt} P_{batt}}}{2 R_{batt}} \tag{4b}$$

The combined power request

$$P_{req} = P_{wheel} + P_{SOC} \tag{5b}$$

Assuming the combined power request has to be met by the engine, the engine speed can be iterated across the engine map and the corresponding engine torques can be calculated. The line formed from these points is called the Power Isomer line.

$$T_e = \frac{P_e}{\omega_e} = \frac{P_{req}}{\omega_e} \tag{6b}$$

The fuel consumption for all engine operating point candidates along a Power Isomer line are interpolated and the one with the minimum $\dot{m}_{fuel}(T_e, \omega_e)$ is selected. The process is repeated to find the minimum engine operating point at every time step. In Fig. A2.1, each dashed grey line represents a Power Isomer line where the engine power is constant. The background contour map represents the engine mass fuel rate with values displayed between 0.2 g/s to 2.5 g/s. The engine operating points are bounded above by the maximum torque line. The engine operating points with minimum fuel rate for every Power Isomer line are shown as red crosses.

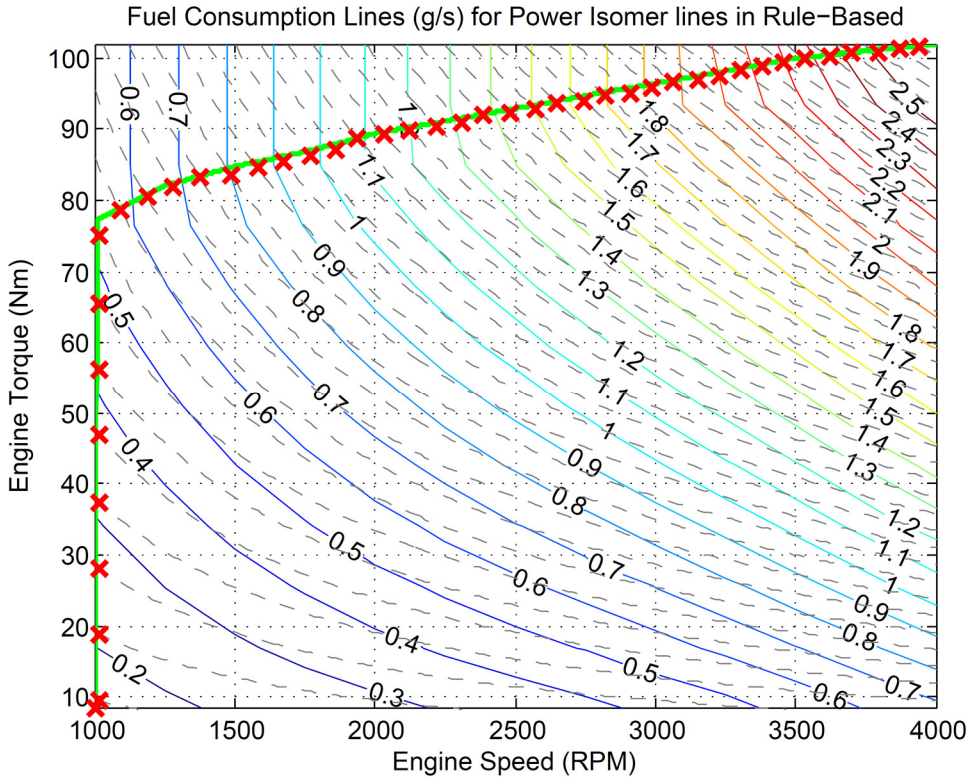


Figure A2.1. Power Isomer Lines on Toyota Prius Engine Map for Rule-Based method

In this work, the powertrain model does not include other sub components such as engine cooling. Therefore, the sub-components conditions involved in the secondary controller such as coolant temperature [67] are not implemented. Only the rules involving the SOC and power requests, P_{req} are used [63,67,68] as the following

1. All-Electric Range : Use only electrical energy if $SOC \geq 0.55$ and $P_{req} \leq 10$kW.
2. Overcharge Avoidance : Turn off battery charging if $SOC = 0.8$.

3. Undercharge Avoidance : If $SOC \leq 51\%$, recharge until $SOC = 55\%$.

The Rule-Based constant, K_{fit} in Eq. 2b is tuned such that the result with UDDS cycle will be closest with the simulations done in [63].

Rule-Based Engine Transient Dynamics Limit

In Chapter 3, where experimental work is done on an actual 4.5L John Deere diesel engine [61,74], the engine transient dynamics have to be considered in Rule-Based method. Shown in Figure A2.2. are the engine operating points for Rule-Based method on the John Deere engine. Each dashed grey line represents a Power Isomer line where the engine power is constant. The background contour map represents the engine mass fuel rate with values displayed between 0.3 g/s to 2.5 g/s The engine operating points are bounded above by the maximum torque line and the engine operating points are selected based on the minimum fuel rate along each Power Isomer line.

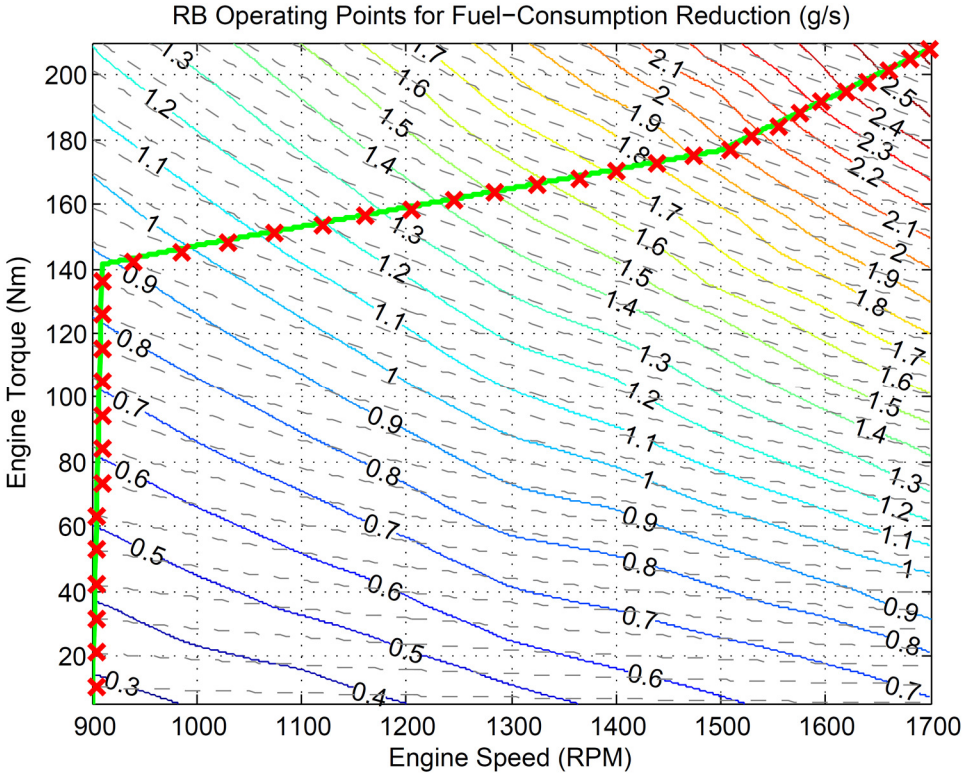


Figure A2.2. Power Isomer Lines on John Deere Engine Map for Rule-Based method

As can be seen, the engine operating points coincide with the maximum torque line, which are three regions of piecewise linear functions with constant slopes.

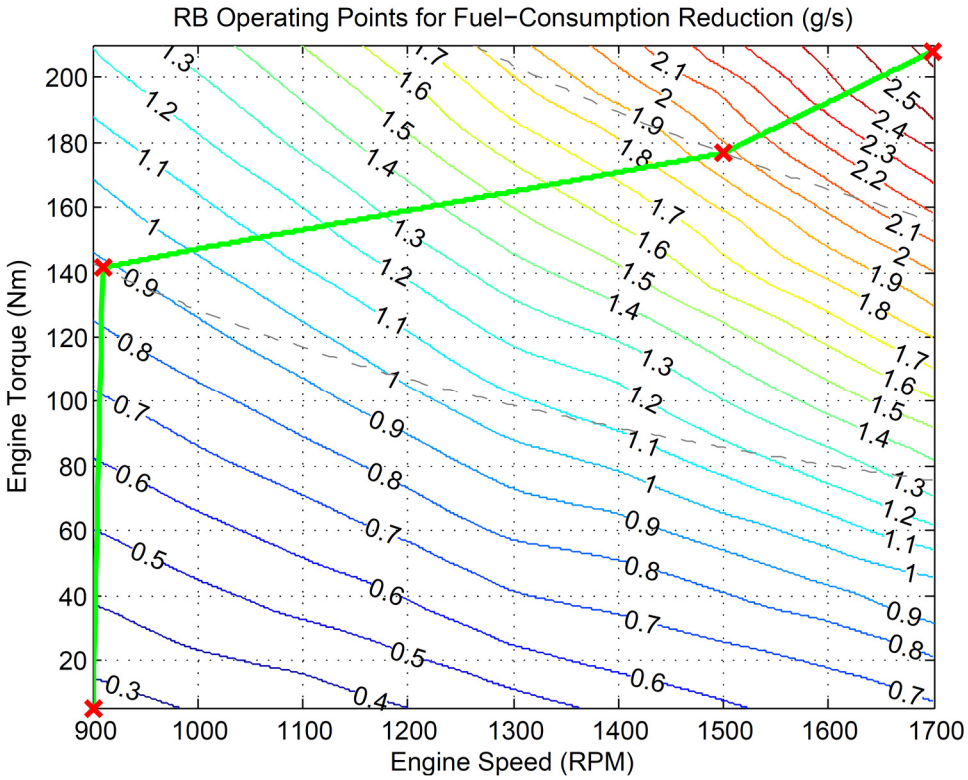


Figure A2.3. Three Sections of Piecewise-Linear Regions for Rule-Based Engine Operating Points

Fig. A2.3 shows the three Power Isomer regions, separated by two grey dashed lines that correspond to Power Isomer lines with values of 13.46 kWatts (bottom line) and 27.75 kWatts (top line):

1. Region 1: Between (0, 13.46) kWatts, bottom engine operating region.
2. Region 2: Between (13.46, 27.75) kWatts, middle engine operating region.
3. Region 3: Between (27.75, 40) kWatts, top engine operating region.

The change of engine speed with respect to the change of engine power (or Power Isomer) can be calculated for the three regions. The same can also be done for the engine torque. By knowing these slopes, the change in power request P_{req} between two time steps can be constrained based on the maximum allowable change in engine speed and engine torque between two time steps. Due to the constrained engine

dynamics, the difference between power demanded by the vehicle and actual power provided by the engine will be compensated by the electrical machines and battery.

Rule-Based Battery Charge Sustaining

For fair MPG comparisons, all optimization methods have to ensure battery charge sustaining. In Rule-Based method, SOC sustenance at the end of a driving cycle cannot be guaranteed as the powertrain operating points depend only on the current loading conditions. Therefore, a simple method of adjusting the requested power from the engine or P_{req}, based on the battery SOC level is used.

In the last 50s of the driving cycle, if the battery SOC is lower than 60% (initial battery charge), P_{req} calculated in Eq. (5b) will be increased by the maximum allowable change in P_{req}

$$P_{req} = P_{req} + max\,(\Delta P_{req}) \tag{7b}$$

Note that *max*(ΔP_{req}) depends on the regions of P_{req} shown in Fig. A2.3. The maximum allowable change in P_{req} is used to force the engine to operate more than the actual requested power in order to charge battery back to 60% as soon as possible. As the battery charge gets closer to 60%, ΔP_{req} is reduced proportionally with respect to the difference of actual battery SOC and 60%.

The same procedure is also used to reduce P_{req} if the battery SOC is higher than 60% in the last 50s of the driving cycle. By reducing P_{req}, the engine will operate less than the actual requested power, therefore draining the battery to meet the actual power demand.

Appendix 3: External Connected Vehicle Controller Architecture

Shown in Figure A3.1 is the integration of the Connected Vehicle (CV) Controller in the HiLS. In this architecture, VISSIM-COM operation is separated into two threads to efficiently handle data transfer. The working procedures are explained below and experimental results are reported as part of the work done in [62,84].

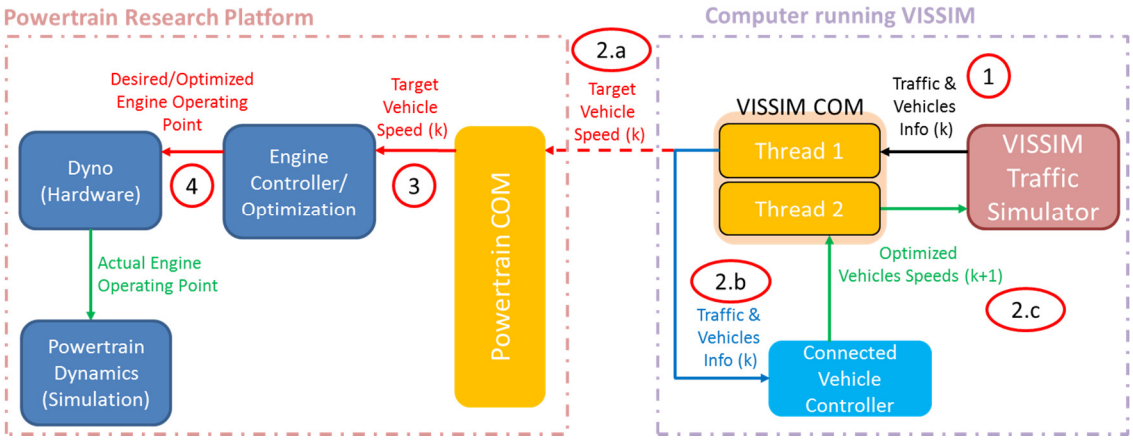


Figure A3.1. Connected Vehicle Controller Integration

1. VISSIM Traffic Data Extraction

First, VISSIM-COM Thread-1 starts VISSIM and executes one simulation time-step in VISSIM. A VISSIM-Dynamic Link Library (DLL) is programmed to run internally within VISSIM software that will extract traffic data and send them to a local virtual network port at every VISSIM simulation time step using UDP protocol. UDP is preferred over TCP for local network communication because UDP is faster and there is no reliability issue with local data transfer. DLL is used as an efficient large data extraction method to avoid time bottleneck when extracting traffic information as the traffic network gets bigger and more complex. Efficient data extraction is shown in Fig. 4.4, Chapter 4.5.1 which shows 89 vehicles data extraction in less than 0.06 ms. The data which now sits at the virtual network port are now available for VISSIM-COM extraction.

2a. VISSIM-COM Traffic Data Transfer to Powertrain COM

VISSIM-COM Thread-1 extracts all the traffic information from the virtual network port and passes only a specific vehicle speed, acceleration and road condition that is represented by the powertrain research platform to the Powertrain-COM across an internet network using socket programming with TCP protocol.

2b. VISSIM-COM Traffic Data Transfer to CV-Controller

VISSIM-COM Thread-1 extracts all the traffic information from the virtual network port and passes them using COM interface to the CV-Controller, which is built in MATLAB-Simulink. CV-Controller contains user-defined functions that can utilize the traffic information to calculate the desired/optimized vehicle speeds in order to control multiple vehicles in VISSIM traffic simulation.

2c. VISSIM-COM Traffic Data Transfer to VISSIM

The desired/optimized vehicle speeds calculated by the CV-Controller are sent back to VISSIM. In this process, VISSIM-COM Thread-2 writes the desired/optimized target vehicles speeds directly to the COM objects in VISSIM that represent the target vehicle speeds. In the next simulation time step, VISSIM will utilize these desired/optimized vehicle speeds as the target vehicles speeds.

3. Powertrain-COM Traffic Data Transfer

The Powertrain-COM receives the target vehicle data that was sent across an internet network by the VISSIM-COM Thread 1 on a virtual network port. Powertrain-COM then sorts the data and passes them to the Engine Controller.

4. Engine Optimization and Execution

The Engine Controller uses the target vehicle speed, acceleration and road condition to calculate the vehicle load and reference desired/optimized engine operating points to be tracked by the dynamometer and the engine. Engine fuel and emissions are then measured by laboratory measurement devices.