HEV FUEL OPTIMIZATION USING INTERVAL BACK PROPAGATION BASED DYNAMIC PROGRAMMING

A Thesis
Presented to
The Academic Faculty

by

Adithya Ramachandran

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W Woodruff School of Mechanical Engineering

Georgia Institute of Technology
May 2016

HEV FUEL OPTIMIZATION USING INTERVAL BACK PROPAGATION BASED DYNAMIC PROGRAMMING

Approved by:

Professor Nader Sadegh, Advisor
George W Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Professor David Taylor
Department of Electrical Engineering
Georgia Institute of Technology

Professor Michael Leamy
Department of Mechanical Engineering
Georgia Institute of Technology

Date Approved: 29 April 2016

To my Parents

Whose guidance, care, love, and support has proven extremely valuable to me

ACKNOWLEDGEMENTS

First and foremost I would like to thank my advisor, Dr. Nader Sadegh, for giving me the opportunity to carry out this research. I feel privileged to work alongside him for the last two years. He has been a guiding light, supporting me at all times and giving me the right advice. His words of encouragement thought the duration of the research spurred me onto this work. In addition to helping with my research, the discussions I had with Dr. Sadegh have helped me develop a deep interest control theory and has motivated me to delve further in this field.

I would also like to thank Dr. David Taylor and Dr. Michael Leamy, who took their valuable time to serve on my thesis committee and provided their valuable feedback.

Finally, I owe my thanks to Mrs. Johnson Glenda of the mechanical engineering department, who has helped guide me through the administrative process of completing my MS thesis degree in a timely manner.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

In this thesis:

- The primary powertrain components of a power split hybrid electric vehicle are modeled. In particular, the dynamic model of the energy storage element (i.e., traction battery) is exactly linearized through an input transformation method to take advantage of the proposed optimal control algorithm.
- A new dynamic programming approach called interval back propagation is introduced. This involves quantization of the energy storage states (i.e., states of charge) into a set of computed intervals.
- A closed form globally optimal solution is obtained for the optimal input under certain conditions.
- The procedure used for real time implementation of the algorithm is elucidated
- The fuel economy results are compared with those from standard rule based techniques to confirm improvement.

A lipschitz continuous and nondecreasing cost function is formulated in order to minimize the net amount of consumed fuel. The globally optimal solution is obtained using a dynamic programming routine that produces the optimal input based on the current state of charge and the future power demand. it is shown that the global optimal control solution can be expressed in closed form for a time invariant and convex incremental cost function utilizing the interval back propagation approach. The global optimality of both time varying and invariant solutions are rigorously proved. The optimal closed form solution is further shown to be applicable to the

time varying case provided that the time variations of the incremental cost function are sufficiently small. The real time implementation of this algorithm in Simulink is discussed and the fuel economy results obtained are compared to the existing rule based methods.

CHAPTER I

INTRODUCTION

1.1 Motivation

Personal Transportation has become a defining feature in American lifestyle over the past few decades. As of 2015, the number of personal vehicles plying on US roads stands at 257.9 million [1]. In addition, the number of cars sold in the United States is a little over 7.7 million [2]. Over the last decade (2001-2010), vehicle sales have increased by 20% [1]. In 2015, the average American drove 13476 [3] miles. It is evident that since cars are so popular (and often a necessity) among the public, consumers are expecting a higher level of performance, reduced cost of maintenance, and improved fuel efficiency.

The average fuel economy of a 2014 model car made in the US stands at 34.2 mpg [4]. This improvement in fuel economy is due to 2 prominent reasons:

- Car manufacturers have figured out methods to reduce the effect of disturbances and improve engine performance characteristics
- Consumers are shifting towards purchase of hybrid and electric vehicles

Given the current sociopolitical scenario, it is likely that the second reason is a more compelling for increase in average fuel economy. It is known that electric vehicles provide spectacular fuel economy, due to the absence of an ICE (Internal Combustion Engine). However, one must come to a realization that the specific energy provided by electrochemical reactions in the battery is far lower than that of petroleum. Therefore, they have inadequate range compared to ICE powered vehicles. In addition, the current infrastructure does not permit sufficiently quick charging time. Hybrid

Electric Vehicles (HEV's) have the niche advantage, since they can sustain a range equivalent to that of an ICE powered vehicle, provide higher fuel economy, and do not need to be charged for long periods of time. The Data below shows that the number of hybrid vehicles driven has increased to 458,994 [5] as of 2013. The average fuel economy provided by a hybrid vehicle is about 47.41 mpg [6] according to 2015 data, which is significantly higher than the typical ICE powered vehicle. A brief description of HEV configurations and their operation has been described below.

1.2 HEV Configurations

Hybrid vehicles use an Internal combustion engine combined with an electromechanical system to operate the vehicle powertrain. The electromechanical system consists of an electric motor and may include a generator depending on the HEV architecture. An electric motor (EM) complements the engine by distributing its load when instantaneous power demand is very high. This leads to engine downsizing while maintaining the desired level of performance, thereby resulting in more efficient operation. In addition, the motor tends to act like a generator system when the vehicle is braking (corresponding to negative power demand) by regenerating electricity from heat developed due to friction forces. This is known as regenerative braking. There are 3 main types of Hybrid vehicle architectures, namely Series, Parallel, and Power split Hybrid Electric vehicles. A brief description of each is provided below, and a more detailed explanation of the power split HEV is provided in the next chapter.

1.2.1 Series HEV

A series HEV is one in which the power produced by the engine and motor are in series. The power flow path is from the engine which powers a generator that in turn charges the battery (if charge is low) which powers the motor, and eventually provides torque to the wheels. There is no mechanical linkage between the powertrain unit and wheels in a series HEV, hence the engine can run at its most efficient operating point.

However, the power produced by the engine and size of electromechanical components has to be fairly large, which increases the overall weight. In addition, the large number of electromechanical conversions lead to inefficiencies. The IC_{work}/EM_{work} is about 1. The series HEV has superior Idling stop, efficiency, energy recovery. However, it lacks in providing sufficient output and acceleration [7]. This HEV architecture is used in Fisker Karma [8]. A simple illustration of the series HEV is shown below [9]:

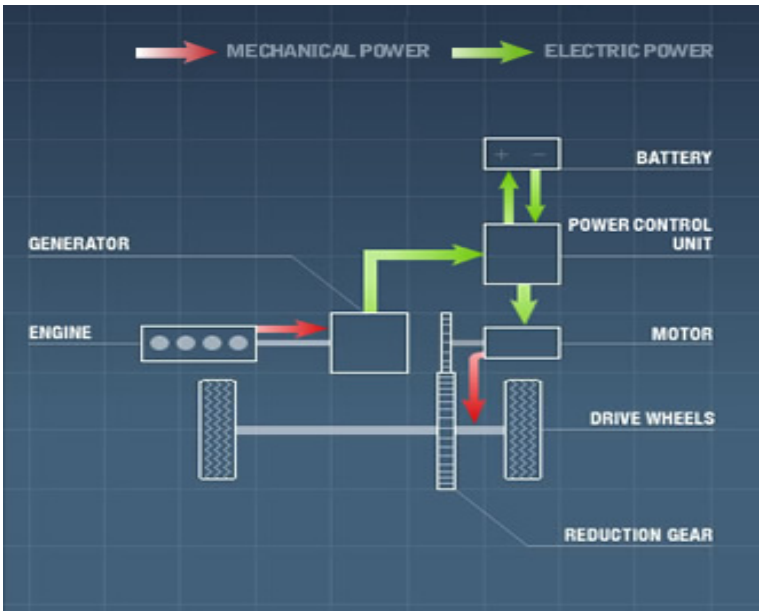


Figure 1: Series HEV

1.2.2 Parallel HEV

A parallel HEV is one in which the power produced by the engine and motor are in parallel. The power flow path is from the engine which is directly linked to the wheels by a transmission system, and the motor which uses the battery to power the wheels. There is no generator in a parallel HEV, However, the motor doubles up as a generator through regenerative braking to recharge the battery. The motor essentially complements engine operation during high power demand situations to provide the desired power output. The IC_{work}/EM_{work} is much greater than 1. The parallel

HEV has superior Idling stop, acceleration, energy recovery. However, it lacks in providing sufficient output and slightly lacks in efficiency. This HEV architecture is most commonly used in the Honda Insight, Civic Hybrid, CR-Z [10], and Hyundai Sonata Hybrid. One should note that the EM's used by Hyundai are much larger than those for Honda, and can individually drive the wheels. A simple illustration of the Parallel HEV is shown below [9]:

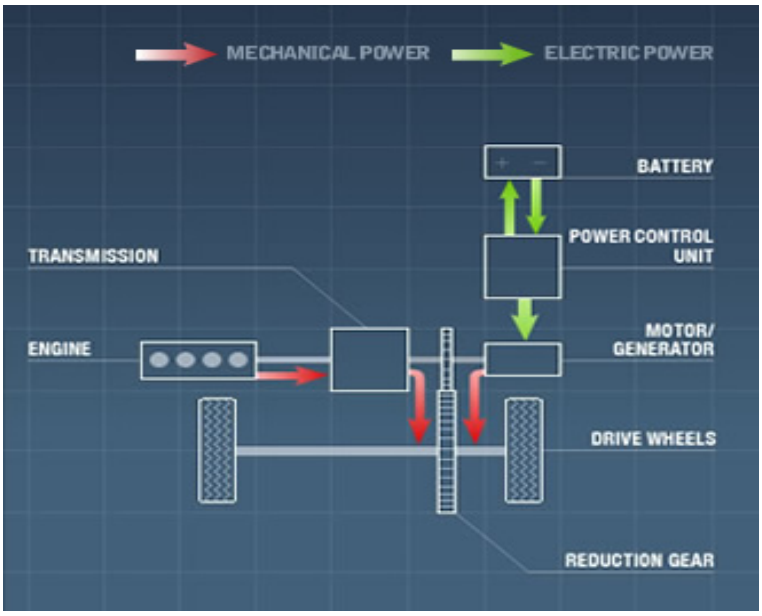


Figure 2: Parallel HEV

1.2.3 Power Split HEV

In a power split HEV, the Power demanded is met by providing the necessary torque requested at the wheels by using the motor alone, or a combination of the engine and motor. Henceforth, one benefits from the advantages presented for the series and parallel HEV architectures. A power split HEV has a ring and sun gear system within its powertrain which distributes power between the engine, motor, and generator systems. However, since a power split model has two degrees of freedom for control [11], its performance is highly dependent on the control algorithm which

monitors the operation of thermal and electrical power sources. A simple illustration of the power split HEV is shown below [9]:

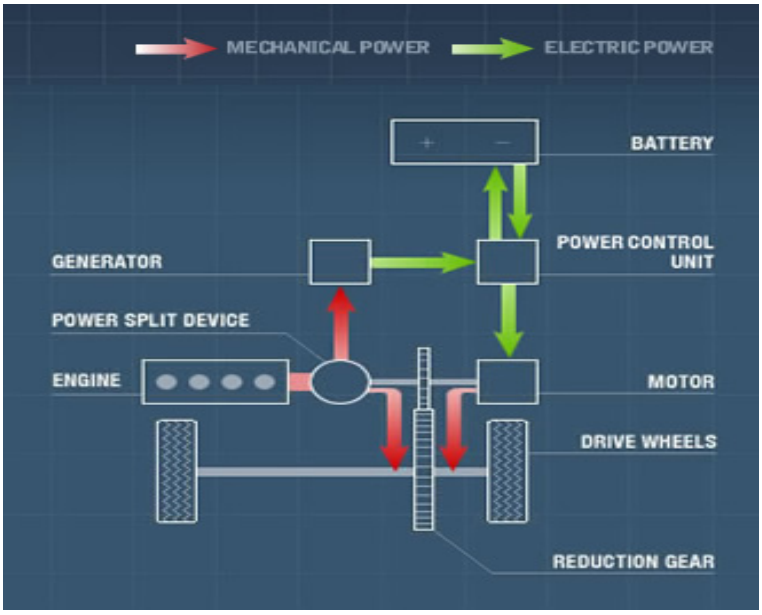


Figure 3: Power Split HEV

1.3 Control methods

As mentioned in the previous section, a key parameter in any Power Split Hybrid vehicle is to decide the proportion of engine power to electromechanical power based upon the driving condition (typically dependent on power demand) that will minimize fuel consumption. This will define the engine's operating condition, from which the fuel economy can be easily determined. To identify the proportion, a control algorithm is necessary. There are two major classification of such optimization algorithm's, namely Rule Based (RB) and Optimization based (OB). A brief overview of existing optimization algorithm's belonging to both categories has been presented below:

1.3.1 Rule Based Control

Rule based control systems are those which are implemented instantaneously, and do not take future driving conditions into consideration. They are mostly based off heuristics, intuition, or prior experience. Rule based systems are very suitable for real time operations. They can be categorized into deterministic rule based control and fuzzy logic rule based control.

Deterministic rule based control is typically based off human experience, and implemented using look up tables. For HEV's, rule based control can be subdivided into on/off control and Power follower control. In on/off control, state of the engine is determined by the Battery SOC, which lies between a minimum and maximum value. Clearly, this control strategy is not capable of handling power demanded under all operating conditions. Hence, it is only useful for a series HEV following a regular commute pattern. Power follower control is used to control the Electric motor in situations where power demand is high. While this helps decrease load on the engine, it does not optimize efficient operation of the drive train, or take into account emission improvements.

Fuzzy logic rule based control is similar to deterministic, but it is designed to take into account inaccuracies in measurements. Fuzzy logic control attempts to mimic the human behaviour process [12, Ch. 1, Pg. 9]. Hence, Fuzzy logic Control tends to be more robust. Fuzzy logic control is subdivided into 3 subcategories, namely conventional Fuzzy control, Fuzzy Adaptive control, and Fuzzy Model Predictive control.

Conventional Fuzzy control is based off a set of heuristic control rules, and these rules are evaluated using fuzzy sets and fuzzy logic as described by Mamdani and Assilian [13, Ch. 1 Pg. 2]. Conventional Fuzzy control algorithms are readily accepted for many engineering applications due to their simplicity and ease of implementation.

In HEV's, conventional fuzzy logic has been used for powertrain control [14]. Specifically, it has been used to determine engine power demand, given the battery state of charge and input power demand.This has helped achieve better engine and battery efficiency while extending the battery life. However, this method assumes a desirable state of charge, to achieve higher efficiency, which may not always be the case in an actual HEV. Conventional Fuzzy logic control has also been implemented in designing a torque control strategy for a parallel HEV [15]. This has led to improvements in fuel economy and maintains the Battery SOC within the specified range more effectively. However, this method fails to maximize efficiency of components when the battery SOC drops too low.

In contrast, Fuzzy Adaptive control takes advantage of partially known systems. It involves approximating the non linear functions in a nonlinear system using fuzzy logic, describing the unknown parameters, and solving for the parameters using well known adaptive control techniques [13, Ch. 1 Pg. 7]. Fuzzy adaptive control has been used for intelligent energy management in hybrid electric vehicles [16] Essentially, it takes into account the driving conditions, drivers style of operating the vehicle, and operating mode to determine the power split strategy that improves fuel economy and reduces emissions. However, this method does not incorporate drive line efficiencies.

Fuzzy Model Predictive Control is a methodology in which a non linear system is composed of several quasi linear systems which are regulated by fuzzy logic [17].It has been used for developing an effective torque split control strategy while incorporating the transient characteristics in engine operation [18]. However, it is heavily influenced by the input initial conditions, due to the fact that the range of optimal inputs used for implementing Model Predictive Control are locally optimal.

Although Rule Based strategies are implementable in real time, they require extensive tuning which can result in sub optimal strategies. Hence, we shall shift our focus to finding effective optimal control strategies used for a power split HEV.

1.3.2 Optimization Based Control

Optimization based control strategies primarily involve finding the optimal input and trajectory which minimizes a cost function in the presence of constraints imposed on system dynamics and parameters. Hence, prior knowledge regarding system operation is necessary. Consequently, optimization algorithms can't be implemented directly in real time. However, the results produced by these algorithms can be used in real time systems to optimize performance. Commonly used optimization algorithms involve:

- Linear Programming
- Optimal Control Theory
- Sequential Quadratic Programming
- Dynamic Programming

A brief description of the above methods had been described below, followed by its application pertaining to HEV's.

A general Linear programming problem involves finding an optimum solution for the problem of minimizing a given linear objective function, subject to a system of linear constraints [19, Ch. 1]. In HEV's The fuel economy optimization is considered as a convex nonlinear optimization problem, which is finally approximated by linear programming method. Linear programming has been used for fuel efficiency optimization in series HEV's [20]. It has also been used for powertrain optimization in parallel HEV's [21] However, the drawback is that this algorithm is highly dependent on the initial and final states of charge, and reducing emissions has not been taken into consideration.

Optimal Control theory is a popular approach for solving most optimization problems. The advantage is that it usually leads to a closed form analytical expression. Pontriagyin's minimum principle is the most commonly used method used to solve

optimization problems with dynamic and parameter constraints. It says that given a set of differential equations described by continuous functions, and a set of initial conditions, one can find an optimal input within the admissible range of inputs and corresponding performance index that minimizes the cost function and Hamiltonian. For a HEV fuel minimization problem, the cost function is usually given by a BSFC or fuel consumption function, while constraints are imposed by vehicle dynamics, engine model, and battery model. However, this method only guarantees local optimality and is highly dependent on the nature of constraints. An illustration of how this approach is used for our HEV algorithm is shown later, and its shortcomings are discussed.

Sequential Quadratic Programming is a common approach used to solve constrained optimization problems. It approximates the cost function as a quadratic function, while the constraint functions are approximated by linear functions. In HEV's, it is primarily used for optimizing power distribution [22]. The primary aim is to maximize system efficiency while meeting the power requirements. However, this method only uses an approximation to the continuous variables considered in a HEV transmission. In addition, this method isn't guaranteed to converge to a globally optimal solution.

Dynamic programming is the most popular and most commonly used method for HEV fuel minimization problems. Dynamic programming is essentially divided into 3 subcategories, namely forward, backward, and backward-forward strategies. Dynamic programming always guarantees global optimality. The following section provides a brief overview of dynamic programming and its applications in HEV fuel optimization.

1.3.3 Dynamic Programming Basics & ECMS

Dynamic Programming is essentially based off Bellman's Principle of Optimality, which states that an optimal policy has the property that no matter what the previous decisions have been, the remaining decisions must constitute an optimal policy with regards to the states resulting from those decisions [23, Ch. 6, Pg. 260]. In other words, when one follows an optimal trajectory between two points, the previous decisions taken to reach an intermediary point do not matter, as long as states resulting from the intermediary point to the end point form an optimal trajectory. For discrete dynamic programming problems, Bellman's method can be directly applied by using the functional equation of dynamic programming [23, Ch. 6, Pg. 264], which is given by:

$$J_k^*(x_k) = \min_{u_k}(L_k(x_k, u_k) + J_{k+1}^*(x_{k+1})) \tag{1}$$

This allows us to optimize over one control vector at a given time. For continuous nonlinear systems, the Hamilton Jacobi Bellman (shown below) equation is typically used to represent the functional equation of dynamic programming:

$$-\frac{\partial J^*}{\partial t} = \min_{u(t)}(L(x, u, t) + \frac{\partial J^*}{\partial x} f(x, u, t)) \tag{2}$$

However, we must realize that this equation is very rarely solvable in analytical form [][Ch. 6, Pg. 278]syrmos3, hence it is of little validity for finding optimal control solutions to most non linear systems. For HEV fuel consumption minimization problems, we will concern ourselves with the discrete form presented above.

There are 3 ways to solve discrete dynamic programming problems. One method is forward dynamic programming, where the first step starts from the state's initial condition. Another is backward dynamic programming where the final condition is considered as the first step. The conditions required for solving an optimization problem with dynamic programming are listed in [24, Sec. 2.2.2]. Note that all of these criteria are met for the HEV fuel minimization problem. A newly developed

novel approach is called backward forward dynamic programming, which computes the cost function of each state with respect to the initial and final conditions. This is a more efficient method than the other two presented above, due to a reduction in the number of calculations [25].

Backward dynamic programming has been used extensively used to optimize Power management and fuel consumption in hybrid electric vehicles. In [26], the power split strategy between fuel and electrical sources along with constraints due to vehicle dynamics has been posed as a dynamical optimization problem with constraints, and it has been solved using backward dynamic programming methods to optimize fuel consumption. A significant reduction in energy consumed has been observed in the above case. Similarly, in [27] it is seen that fuel economy improved by 21% using a backward dynamic programming algorithm to optimize the power split operation.

Traditionally, backward dynamic programming routines cannot be implemented in real time as they are computationally prohibitive. One such method for dealing with this issue is Adaptive dynamic programming. Adaptive dynamic programming is an on-line tuning method, which controls the system while simultaneously learning its characteristics in real time [28]. Doing so, one sees that this control approach outperforms conventional rule based strategies by 12.3 % across the UDDS, HWFET, US06, and LA92 cycles.

Another commonly used method to solve the HEV fuel minimization problem with dynamic programming is the Equivalent Cost Minimization strategy (or ECMS). The ECMS strategy is used as it is an instantaneous minimization algorithm [29]. In the ECMS strategy, one modifies the cost function to incorporate operation of the system when subject to inequality constraints. The general form of our cost function C is:

$$C_k(x) = \sum_{i=1}^{n} s_i L_i(x_k, u_k) \tag{3}$$

Here, s_i represents the equivalence weighting factors, and $L_i(x_k, u_k)$ is the performance index. The aim here is to optimize the equivalence factors such that the fuel consumption can be minimized. Another way to visualize the ECMS technique is shown in [30], where the Cost function is divided into 2 parts, those arising from engine operation and those from constraints on electrical components. An equivalence factor is adjoined along with the cost due to constraints on the electrical system, and as mentioned earlier, the aim is to optimize this equivalence factor.

On implementing this strategy, it is seen that the fuel economy deviates by less than 1% from the optimal value. In addition, it has been observed that fuel consumption can be improved by 17.5% for the CEN cycle [31].

Optimal solutions HEV fuel minimization problem can be further enhanced by combining Adaptive dynamic programming and ECMS strategies into one supervisory control unit. This algorithm is called the A-ECMS. It attempts to minimize fuel consumption, while simultaneously ensuring that the battery charge is within specified bounds [32]. This method also shows an improvement in performance over the standard ECMS method across a variety of commonly used controllers [33]

However, a major drawback of the A-ECMS algorithm is that it significantly complicates the cost function based on the number and nature of the constraints, and makes the cost to go highly non convex. Its been proven in [34, Ch. 8] that for a convex function, the local and globally optimum solutions will coincide, and the A-ECMS method does not take advantage of this property.

The offline dynamic programming approach that is usually used for solving HEV problems is exhaustive search. In this method, the design approach is to exhaustively search all admissible states and minimize an instantaneous cost function based on engine power and battery power, at each time instant of the drive-cycle [35]. However, this can prove to be a slow and time consuming process for a complicated drive cycle if a large number of states are considered. In addition, this method cannot effectively

deal with excessive switching between states.

In this thesis, we shall show that the convexity property of our cost function can be used to derive a globally optimal solution by employing an interval back propagation approach along with the dynamic programming routine, while satisfying all the necessary inequality constraints. An additional advantage of this method is that it provides us with a closed form solution, which can be implemented in real time. Subsequently, we shall show that the fuel economy is significantly improved in comparison to standard rule based techniques.

1.4 Contribution

- The primary powertrain components of a power split hybrid electric vehicle are modeled. In particular, the dynamic model of the energy storage element (i.e., traction battery) is exactly linearized through an input transformation method to take advantage of the proposed optimal control algorithm.

- A new dynamic programming approach called interval back propagation is introduced. This involves quantization of the energy storage states (i.e., states of charge) into a set of computed intervals.

- A closed form globally optimal solution is obtained for the optimal input under certain conditions.

- The procedure used for real time implementation of the algorithm is elucidated

- The fuel economy results are compared with those from standard rule based techniques to confirm improvement.

CHAPTER II

POWER SPLIT SYSTEM MODELLING

From chapter 1, recall that the power split hybrid electric vehicle comprises of an engine, motor, and generator. The power demanded can be delivered in series mode with the motor alone, or parallel mode with the engine and motor operating in tandem. It is also important to realize that a power split hybrid electric vehicle has no transmission component, hence losses due to transmission are nonexistent. A ring and sun gear system helps redistribute the power between the engine, motor, and generator. The motor and generator in a power split hybrid electric vehicle receive power from the battery. A detailed description of how each power source is modeled will be discussed in this chapter, and the nature of their overall interaction will also be presented. Finally, the dynamics will be modeled into a linear system of the form:

$$x_{k+1} = x_k + u_k - p_k \tag{4}$$

We will briefly discuss on how the fuel consumption rate is calculated, and its relationship with the cost function. The above system will be the dynamical constraint for our optimization problem, which attempts to minimize the amount of consumed fuel as given by the cost function.

2.1 Engine Model

The internal combustion engine (ICE) is currently the most commonly used power plant in motor vehicles. An ICE uses a predetermined mixture of combustible fuel and air which flows into the engine during the intake stroke. Through chemical reactions, it releases energy when subject to heat and pressure. The heat is caused by a spark ignition, while high pressure is due to the compression stroke.

During the expansion stroke, the fuel air mixture supplies power to move the piston and operate our power split gear system linked to the engine through a sun gear and shaft. Realize that the lack of a transmission system in a Power split HEV means that power losses are greatly reduced. In conventional vehicles and parallel HEV's, losses tend to be exacerbated by a transmission system. Another advantage of ICE's in HEV's is that the engine can be downsized since the Electrical system provides a fraction of the power. Therefore, the engine's displacement can be reduced significantly, thereby resulting in less power loss while moving the piston.

During the exhaust stroke, the byproducts resulting from combustion are emitted from the engine. An emission detector and catalytic converter are used to minimize the ejection of harmful oxides into the environment. In addition, HEV engines use lesser fuel during each cycle in comparison to conventional vehicles, due to their reduced power requirement and smaller size, thereby making them more fuel efficient.

The engine model we used was based off the 1.5 L 2007 Toyota Prius. A cross section of this engine is shown in figure 4 [36]:

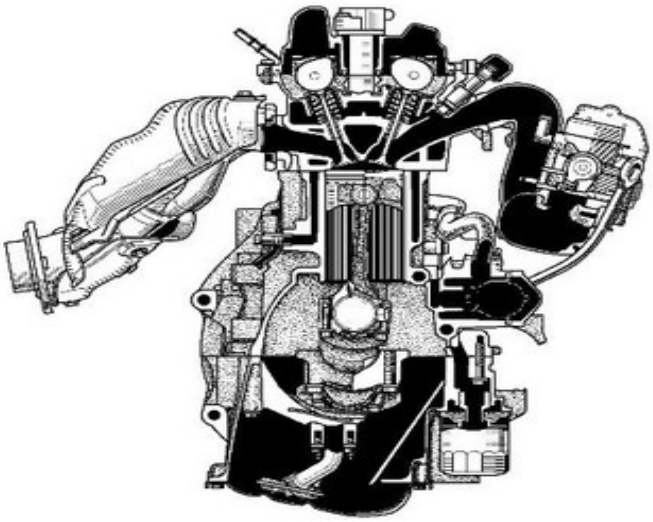


Figure 4: Toyota prius 2007 1.5 L Engine

This engine achieves high efficiency using the Atkinson cycle, one of the most

heat-efficient, high-expansion ratio cycles. Because the expansion ratio is increased by reducing the volume of the combustion chamber and the chamber is evacuated only after the explosion force has sufficiently fallen, this engine can extract all of the explosion energy. Consequently,trying to increase the expansion ratio also increases the compression ratio, resulting in unavoidable knocking and placing a limit on increases in the expansion ratio. To get around this problem, the timing for closing the intake valve is delayed, and in the initial stage of the compression stroke, part of the air that has entered the cylinder is returned to the intake manifold, in effect delaying the start of compression. In this way, the expansion ratio is increased without increasing the actual compression ratio. Since this method can increase the throttle valve opening, it can reduce the intake pipe negative pressure during partial load, thus reducing intake loss. VVT-i (Variable Valve Timing-intelligent) is used to carefully adjust the intake valve timing according to operating conditions, which guarantees maximum efficiency. [36].

The maximum engine power and engine speed as a function of engine speed was obtained from [37]. A piecewise cubic, and piecewise quadratic relationship was used to model the engine torque, subject to derivative constraints on maximum torque, and the torque at point of transition. Consequently, power would be represented by a piecewise quartic and piecewise cubic polynomial, subject to derivative constraints on maximum power, and power at the point of transition. In addition, we have constraints based off the data points obtained from torque Vs. speed data for this specific engine. The equations and constraints that constitute our engine model are as described below:

Polynomial Equations:

$$w = \frac{\omega - \omega_{mid}}{\omega_{max} - \omega_{mid}} \tag{5}$$

$$T = \begin{cases} a_0 + a_1 w + a_2 w^2 + a_3 w^3 & \omega_{min} \leq \ w \leq \ \omega_{mid} \\ T_{mid} + (T_{max} - T_{mid}) 4w(1-w) & \omega_{mid} \leq \ w \leq \ \omega_{max} \end{cases} \tag{6}$$

$$P = T\omega \tag{7}$$

Constraints:

$$T(\omega_i) = T_i \tag{8}$$

$$\frac{dT}{d\omega}(\omega_{mid}) = \frac{4T_{max} - T_{mid}}{\omega_{max} - \omega_{mid}} = a_1 + 2a_2\omega_{mid} + 3a_3\omega_{mid}^2 \tag{9}$$

$$\frac{dT}{d\omega}(\omega_{maxtrq}) = 0 \tag{10}$$

In equations (5-10), $\omega, \omega_{mid}, \omega_{maxtrq}, \omega_{max}$ are the engine speeds corresponding to the minimum value, transition value between our polynomial equations, maximum torque value, and maximum value as described by engine parameters. Meanwhile, w represents the normalized engine speed, T, T_{mid}, T_{max} represent the torque values, transition torque, and maximum torque values, i represents the number of data points, and P represents the engine power. The coefficients given by a_0, a_1, a_2, a_3 are calculated using curve fitting methods when subject to the above constraints.

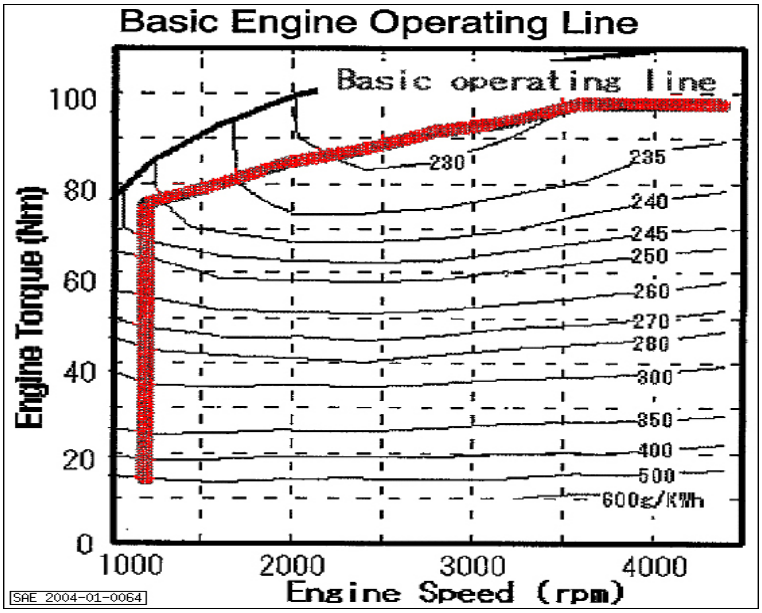


Figure 5: Toyota prius BSFC map

Table 1: Table of Engine Parameters

Parameter	Value
Maximum power	57 kW
T_{max}	115 Nm
T_{mid}	110 Nm
ω_{max}	5000 rpm
ω_{mid}	3000 rpm
ω_{min}	250 rpm
ω_{maxtrq}	4200 rpm
a_0	72.5
a_1	116.25
a_2	-180
a_3	101.25
U^*	20.9809 kW

Our fuel consumption map is a look up table which contains the mass flow rate of fuel $\dot{m}_f$ for any given engine speed and torque value. In simulation, linear interpolation is used to evaluate the fuel consumption rate for values not listed in the table. The fuel consumption map for our engine is as shown in figure 5 [38].

Knowing the Torque and engine speed, one can calculate the Brake specific fuel consumption (BSFC) using a fuel consumption map. Recall that BSFC is given by:

$$BSFC = \frac{\dot{m}_f}{T_e \omega_e} \tag{11}$$

For this engine, power level corresponding to the minimizing BSFC value is chosen as one of the optimal control inputs, denoted by U^*. This parameter will play a critical role in formulating the dynamic programming method, and interval back propagation algorithm. For our engine model, the parameters are given in table 1. The U^* value presented accounts for efficiency losses.

2.2 Battery Model

The battery used for powering the electrical system was a Lithium ion battery, as opposed to the conventional nickel metal hydride battery. The reason for making this choice was due to high energy density and less hysteresis losses. A lithium ion battery has 2.5 times the energy density of a Nickel Metal Hydride battery. Also, a nickel metal hydride battery has significant hysteresis losses, which is negligible in the case of a lithium ion battery. In addition, a lithium ion battery also has the following advantages compared to other existing batteries:

- Lighter than other rechargeable batteries for a given capacity
- Delivers a high open circuit voltage
- Low self discharge rate

A Lithium ion battery has 3 major components:

- **Anode:** The anode usually gives up electrons during the charging process, and therefore is oxidised in the electrochemical reactions. In lithium ion battery, the anode usually consists of carbon or graphite based compound of lithium, denoted by Li_xC_6
- **Cathode:** The Cathode usually accepts electrons during the charging process, and therefore is reduced in the electrochemical reaction. In a lithium ion battery, this is usually a transition metal oxide or phosphate, such as $LiCoO_2$
- **Electrolyte:** The electrolyte used is usually an inorganic non aqueous inorganic lithium salt solution

During the charging process, Li^+ ions are released from the cathode, move across the electrolyte, and deposit in between the graphite layers on the anode. This is an

electrochemical reduction reaction at the anode. Meanwhile, during the discharge process, Li^+ ions are released from the anode, move across the electrolyte, and deposit at the cathode. Subsequently, the cathode accepts these ions to reform $LiCoO_2$ material that was lost during the charging process. This is an electrochemical reduction reaction at the cathode. The basic structure of a Lithium Ion Battery is depicted in figure 6 [39]:

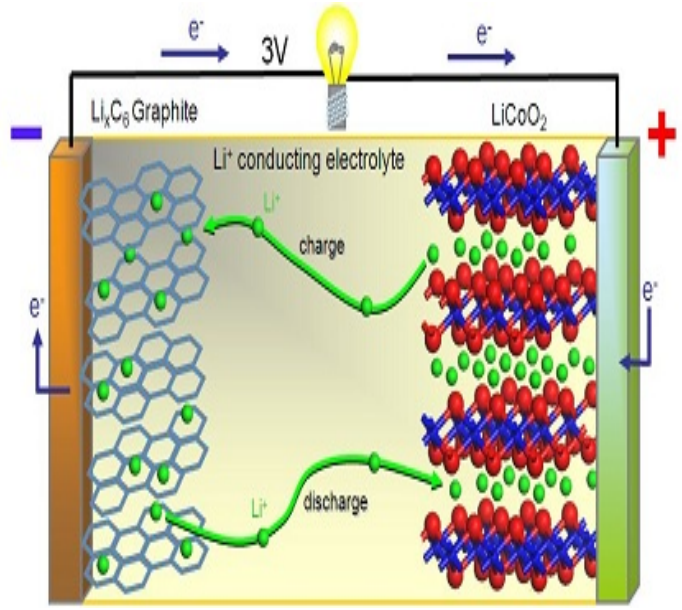


Figure 6: Lithium Ion Battery

Typically, the discharge capacity C_{dis} for any battery is a function of the discharge current I_{dis} and discharge time T_{dis}. It is governed by a relationship referred to as Peukert's law, which says that:

$$C_{dis} = I_{dis}^k T_{dis} \tag{12}$$

Here, k is the Peukert's constant which usually varies between 1 and 1.28 for lead acid batteries. However, as mentioned in [40] one must be note that this discharge relationship using Peukert's law is only valid for a limited current range, and a constant working temperature. For our operating purposes, the temperature will remain fairly constant, but the current will vary quite a bit. Therefore, we shall use the discharge

equation shown below [41]:

$$f_1(it, i^*, I_b) = E_0 - \frac{KQ_b i^*}{Q_b - it} - \frac{KQ_b it}{Q_b - it} + Ae^{-Bit} \tag{13}$$

Similarly, for charging we have:

$$f_2(it, i^*, I_b) = E_0 - \frac{KQ_b i^*}{0.1Q_b + it} - \frac{KQ_b it}{Q_b - it} + Ae^{-Bit} \tag{14}$$

Here, it, i^* represent the Extracted capacity and low frequency current dynamic also called as the filtered current, I_b represents the battery current, E_0 Represents the nominal voltage, Q_b represents the battery capacity, K is the polarization constant, A and B are the exponential voltage and capacity respectively. From equations (13) and (14), realize that the term $\frac{KQi^*}{0.1Q+it}$ represents the charging dynamic, $\frac{KQi^*}{Q-it}$ represents the voltage discharge dynamic $\frac{KQit}{Q-it}$ is the no load voltage, and Ae^{-Bit} is the exponential voltage. The presence of a filtered current helps achieve a sufficiently slow voltage dynamic for a current step response. However, it is important to note that the above charge and discharge relationship only holds true if the current is limited within reasonable bounds for the HEV electrical system. In addition, note that the battery current can be represented in terms of the battery capacity as:

$$\dot{Q}_b = I_b \tag{15}$$

The above relationship is a consequence of net power flow through the battery. Note that open circuit voltage V_{oc} is a function of battery capacity from the charge and discharge relationships shown in equations (**??**-**??**). In addition, the polarization resistance also varies as a function of battery capacity. Here, $R_+(Q_b), R_-(Q_b)$ shall denote the polarization resistance during the charge and discharge process. From equations (13-14), we observe that for our battery:

$$R_+(Q_b) = \frac{KQ_b}{0.1Q_b + it} \tag{16}$$

$$R_-(Q_b) = \frac{KQ_b}{Q_b - it} \tag{17}$$

Hence, the battery power P_b can now be represented as

$$P_b = V_{oc}(Q_b)I_b + R_{\pm}(Q_b)I_b^2 \tag{18}$$

Realize that P_b and I_b are positive when battery gets charged (gains energy), and negative during discharge (loses energy). Let us denote internal energy of our battery as:

$$S_b = U(Q_b) = \int_0^{Q_b} V_{oc}(Q)dQ \tag{19}$$

Then, from equations (14),(18), and (19) we have:

$$V_{oc}(Q_b)I_b = \frac{dU(Q_b)}{dQ_b}\frac{dQ_b}{dt} = \frac{dU(Q_b)}{dt} = \dot{S}_b \tag{20}$$

$$R_{\pm}(Q_b)I_b^2 = \frac{R_{\pm}(Q_b)}{V_{oc}(Q_b)^2}\dot{S}_b \tag{21}$$

Now, defining $\epsilon_{\pm}(S_b) = \frac{R_{\pm}(Q_b)}{V_{oc}(Q_b)^2}$ we can express the power demand as:

$$P_b = \dot{S}_b + \epsilon_{\pm}(S_b)\dot{S}_b^2 \tag{22}$$

For a lithium ion battery, notice that the voltage as a function of SOC remains linear from 20 to 80 % state of charge as shown in the figure below [42].

This is the range within which our interval back propagation algorithm will be applied to optimize the fuel economy. Within this range, the open circuit voltage $V_{oc}(Q_b)$ is assumed constant. Therefore, from equation (19) we can say that $U(Q_b) \approx V_{oc}Q_b$, and so $Q_b \approx \frac{S_b}{V_{oc}}$. Also, notice that the nonlinear function $\epsilon_{\pm}(S_b)$ in equation (22) remains constant over our selected operating range. Therefore, we shall approximate it as $\epsilon_b \in [0, 1]$ for the remainder of this chapter. In addition, a physical power limit of 25 kW for the discharge process has been imposed on the battery. It is important to remember that during operation, the amount of power that the motor can deliver or generator can accept is subject to the battery power limit, even if either electronic component has a higher maximum power rating. The operational constraints for our battery are as stated in table 2.

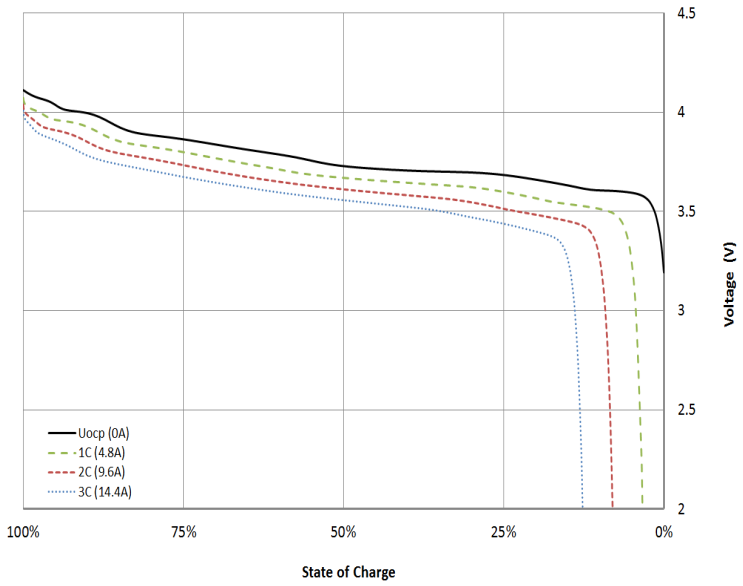


Figure 7: Li-Ion battery Discharge Curve

Table 2: Table of Battery Parameters

Parameter	Value
E_0	220 V
ϵ_b	$7.25 * 10^{-4}$
R_b	0.0347 Ω
Q_b	8.1 Ah
$I_{b,dis}^{max}$	130 A
$P_{b,dis}^{max}$	25 kW

2.3 Motor and Generator Dynamics

For our power split HEV, an AC synchronous motor and generator are used. An AC synchronous motor or generator is essentially a high efficiency brush less DC motor or generator. The performance advantages in using a brushless DC motor (or generator) over a brushed DC motor are:

- More accurate position control due to electronic commutation using hall effect sensors
- Ability to deliver higher torques at a given speed

- Higher power output and speed range
- Higher efficiency due to less voltage drop
- Superior Thermal characteristics due to better heat dissipation

[43]

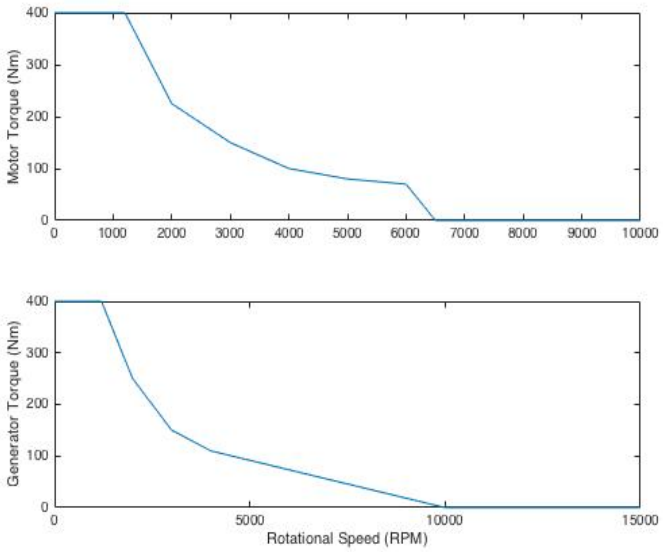


Figure 8: Torque speed curves for motor and generator

For the Motors, permanent magnets are arranged in a V formation, along with a rotor made of stacked electromagnetic plates to help improve the torque and power output [36]. Meanwhile, rotor strength enhancements to the generator enable it to rotate at high speeds of up to 10000 rpm. This allows it to supply sufficiently high power to the motor, which in turn meets the torque demanded at the wheels. Hence, the operation of the HEV is optimized for low and medium speed applications, as the engine does not have to supply power to satisfy the power demanded, unless state of charge is significantly low.

The final motor torque delivered is dependent on the torque requested at the wheels and torque limits imposed by the motor speed controller. We have limited the

maximum and minimum torque output of the motor between 400 and -300 Nm. The motor speed is directly related to the vehicle speed as it is connected to the axle.

The maximum power for the motor is 50.26 kW and that of the generator is 52.35 kW. The motor and generator losses are characterized by the single measurement efficiency model is given by $\eta = \frac{P}{P+k_eT^2}$ where P is the mechanical power and T is the shaft torque. This equation is very similar to the one presented in [44] where all power losses have been lumped into k_eT^2. The efficiency constant k_e is calculated such that η is equal to the nominal efficiency at a certain output speed and torque. The mechanical power P is treated as the positive output power for a motor, and negative input power for the generator. Also, realize that for a generator, η is actually inverse of the true efficiency. The nominal efficiencies for the motor and generator is 90%,resulting in a k_e value of 0.1164. However, it is important to realize that efficiency of these components are a decreasing function of torque delivered. The torque Vs. speed curves for the motor and generator systems are shown in figure 8.

2.4 Interaction of Engine, Motor, and Generator through Power Split

A planetary gear is used as the primary power split device, and consists of a ring gear, sun gear, carrier gear, and pinion gears. The ring gear is connected to the generator and motor, while the carrier gear is connected to the engine. Let $\omega_g, \omega_m, \omega_e$, represent the generator, motor, and engine speeds respectively. Also, let r, s represent the ring gear and sun gear radius. Then, due to the mechanical connections we can say that:

$$\omega_e = \frac{r}{r+s}\omega_m + \frac{s}{r+s}\omega_g \tag{23}$$

Also, let the carrier and sun gear input torques be denoted by T_c and T_s, respectively, and the ring gear output torque by T_r . Ignoring the pinion gear inertia and lumping the ring, carrier, and sun gear inertias with the motor, engine, and generator inertia's, respectively, we may assume without loss of generality that the power split device has

zero inertia. Therefore, we have that

$$\frac{T_c}{r+s} = \frac{T_r}{r} = \frac{T_s}{s} \tag{24}$$

Realize that a power split gear system shown in figure 9 [45] has two paths for power flow. One is the mechanical path and the other is an electrical path. The mechanical path transfers engine power into the carrier gear, which is linked with the ring gear directly connected to the motor axle. The electrical path uses the remaining engine power to operate the generator, which usually charges the battery [46]. The dynamic equations have been described in [47] and are as summarized below.

$$\dot{\omega_e} I_e = T_e - T_c \tag{25}$$

$$\dot{\omega_g} I_g = T_s - T_g \tag{26}$$

$$\dot{\omega_m} I_m = T_m - \frac{T_d}{K} + T_r \tag{27}$$

$$T_d = T_f + mgf_r R_t + 0.5\rho A C_d v^2 R_t + m\dot{v} R_t + mgR_t sin(\theta) \tag{28}$$

Here, T_e, T_g, T_m represent engine, generator, and motor torques respectively, R_t is the tire radius, K is the final drive ratio, I_m, I_g, I_e are the inertia's of the motor combined with ring gear, generator combined with sun gear , and engine combined with carrier gear, m, g, f_r, ρ, A, C_d are the mass, rolling friction coefficient, density of air, Frontal Area, and drag coefficient. These parameters are described in table 3.

In addition, vehicle dynamics terms that constitute the torque demand T_d from the system above can be classified into 3 categories:

- **Aerodynamic resistance:** This resistive force is due to the flow of turbulent air across a vehicle surface. As the vehicle moves forward, it breaks up the flow of air and creates a region of low pressure air behind its body, called as a wake. The high pressure air around the wake then moves into the low pressure region, thereby exerting a force opposite to the direction of motion [48].

Table 3: Table of Power Split Parameters

Parameter	Value
I_m	0.0233 kg/m^2
I_g	0.0233 kg/m^2
I_e	0.18 kg/m^2
R_t	0.3 m
m	1250 kg
ρ	1.225 kg/m^3
A	2.16 m^2
C_d	0.26
$\frac{r}{s}$	2.6

Aerodynamic resistance is generally represented as a function of vehicle velocity and air density. From equation (28), the term $0.5\rho AC_d(\omega_r/K)^2R_t^2$ represents aerodynamic resistance. Note that the vehicle speed is given by $(\omega_r/K)R_t$. From this relationship, it is clear that the aerodynamic resistance will increase significantly if vehicle speed is high.

- **Rolling resistance:**Rolling friction mainly results from tire deformation, wheel slip, and surface compression which act against the vehicle's direction of motion. For our dynamical system, it has been represented as $mgf_rcos(\theta)$. However, since longitudinal motion is considered in this case, $\theta = 0$ and so the rolling resistance becomes mgf_r.

- **Grade Resistance:** The grade resistance is primarily due to gravity, and can be represented as $mgsin(\theta)$. Note that if we assume a flat road, $\theta = 0$ and consequently the Grade resistance is zero.

Considering vehicle dynamics along the longitudinal direction, governing equations for the power split are:

$$(I_e + (\frac{r+s}{s})^2 I_g)\dot{\omega}_e - (\frac{r(r+s)}{s^2})I_g)\dot{\omega}_m = T_e - \frac{r+s}{s}T_g \tag{29}$$

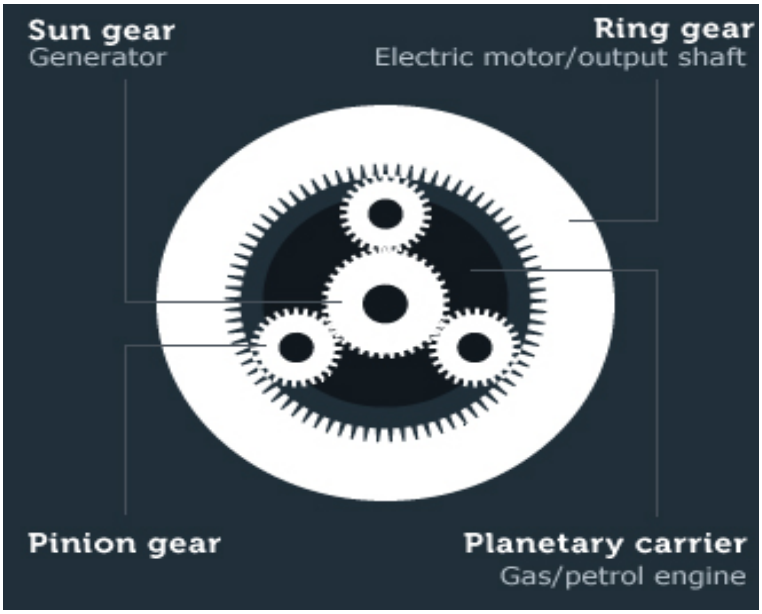


Figure 9: Toyota HEV Power Split System

$$(I_m + (\frac{r}{s})^2 I_g)\dot{\omega}_m - (\frac{r(r+s)}{s^2})I_g)\dot{\omega}_e = T_m - \frac{1}{K}T_d + \frac{r}{s}T_g \tag{30}$$

Realize the engine and generator form a coupled system of differential equations. In matrix form, this differential equation can be described as:

$$\begin{bmatrix} (I_e + (\frac{r+s}{s})^2 I_g) & -(\frac{r(r+s)}{s^2})I_g) \\ (I_m + (\frac{r}{s})^2 I_g) & -(\frac{r(r+s)}{s^2})I_g) \end{bmatrix} \begin{bmatrix} \dot{\omega}_e \\ \dot{\omega}_m \end{bmatrix} = \begin{bmatrix} T_e - \frac{r+s}{s}T_g \\ T_m - \frac{1}{K}T_d + \frac{r}{s}T_g \end{bmatrix} \tag{31}$$

Notice that we represent $\dot{\omega_m}$ in terms of $\dot{\omega}_e$ by elimination of $\dot{\omega}_g$ from the coupled engine-generator system present for power split HEV's. Also, total kinetic energy of the engine and generator system is denoted by:

$$E(\omega_g, \omega_e) = \frac{1}{2}(I_g\omega_g^2 + I_e\omega_e^2) \tag{32}$$

Realizing that rate of change of energy corresponds with power, we have that:

$$\dot{E}(\omega_g, \omega_e) = I_g\omega_g\dot{\omega}_g + I_e\omega_e\dot{\omega}_e \tag{33}$$

$$\dot{E} = P_e - P_g - P_r \tag{34}$$

In a power split HEV, the engine can be decoupled from the vehicular load, which proves to be especially useful in city cycles with frequent stop and go situations.

Essentially the engine is controlled by the motor and generator. When the vehicle is accelerating or maintaining a constant speed, the motor acts as the torque controller, while the generator is in speed control mode. When the vehicle is braking, the motor essentially acts as a generator and uses regenerative braking till it reaches the set torque limit, and if the torque limit's exceeded, the remaining braking force is provided by the mechanical braking system.

A divide and conquer approach has been suggested by Toyota [49] to decouple the multiple input control design. The system optimization specifies the required engine power and the engine optimization selects the optimal *steady–state* ($\dot{\omega}_e = 0$) engine speed and torque. The desired engine speed is then achieved by manipulating the generator speed ω_g. With the help of engine data, one can determine the desired engine torque. Thus, from the power split relationship shown earlier, one can obtain generator torques. This engine speed may or may not be achieved depending on the torque limits imposed on the generator. In such a case, the power surplus is then supplied by the motor [46]. In our case, we shall assume that the engine operates along predetermined curves such that the engine torque is given by $T_e = h_e(\omega_e)$, where $h_e(\omega_e)$ is a non negative continuous function. Defining $\rho = \frac{r}{r+s}$, the generator power and motor power according to the power split and power balance equation can be expressed as:

$$P_g = P_e - P_r - \dot{E} \tag{35}$$

$$P_m = P_d - P_r \tag{36}$$

where:

$$P_d = T_d\omega_m + I_m\omega_m\dot{\omega} \tag{37}$$

$$P_r = \rho(h_e(\omega_e)\omega_m) \tag{38}$$

Note that the power split incorporates the series and parallel power modes. Suppose $\rho = 0$, then $P_g = P_e - \dot{E}, P_m = P_d$, and the vehicle essentially operates in series mode.

On the other hand, if $\rho = 1$ then $P_g = P_e - h_e(P_e)\omega_m = 0, P_m = P_d - h_e(P_e)\omega_m = P_d$, and the vehicle essentially operates in parallel mode.

2.5 Modeling the Dynamical System and Cost Function

The control input is defined as the sum of net power flowing into the battery, and modified power demand $\tilde{P}_d$. In other words:

$$U = V_{oc}(Q_b)I_b + \tilde{P}_d \tag{39}$$

Realize from equation (19) that $\dot{S}_b = V_{oc}(Q_b)I_b$, therefore:

$$\dot{S}_b = U - \tilde{P}_d \tag{40}$$

Recall that due to battery limitations during the discharge process, the input U must be applied such that $|U - \tilde{P}_d| \leq P_b^{max}$ is satisfied. The power flow into the battery can also be represented as:

$$P_b = P_g - P_m - k_e T_g^2 - k_e T_m^2 \tag{41}$$

where $k_e T_g^2$ and $k_e T_m^2$ denote the generator and motor torque losses, respectively. Note that the power flow represented in the above equation is bidirectional, since either motor power or generator power can be positive or negative. Observe that the generator power that flows into the battery is reduced by $k_e T_g^2$ if $P_g > 0$, and the motor power that the battery/generator needs to supply is increased by $k_e T_m^2$ if $P_m > 0$

Using the power split relationships from equation 25 in P_e, P_m provided by equations (35-36), and substituting it in equation 41, one obtains:

$$P_b = P_e - P_d - \dot{E} - k_e(T_g^2 + T_m^2) \tag{42}$$

Also, note that from equation (22)

$$P_b = \dot{S}_b + \epsilon_b \dot{S}_b^2 = U - \tilde{P}_d + \epsilon_b(U - \tilde{P}_d) \tag{43}$$

Hence, from equations (42-43) along with the expression for $\dot{E}$ assuming a constant engine speed from equation (34) we have that:

$$P_e - P_d + \frac{r}{s} I_g \omega_g \dot{\omega}_m - k_e(T_g^2 + T_m^2) = U - \tilde{P}_d + \epsilon_b (U - \tilde{P}_d)^2 \tag{44}$$

where $T_g = \frac{s}{r+s} T_e + \frac{r}{s} I_g \dot{\omega}_m$, $T_m = T_d - \frac{r}{r+s} T_e$, and $T_d = P_d/\omega_m$ is the torque demand. Letting ω_e^* denote the engine speed that minimizes the BSFC, the modified power demand $\tilde{P}_d$ is chosen such that:

$$P_e = P_e^* := h_e(\omega_e^*)\omega_e^* \Leftrightarrow U = U^* = P_e^* - k_e h(\omega_e^*)^2 \tag{45}$$

Therefore, we have:

$$\epsilon_b (\tilde{P}_d - U^*)^2 - (\tilde{P}_d - U^*) - P_b^* = 0 \tag{46}$$

where:

$$P_b^* = P_e^* - P_d + \frac{r}{s} I_g \omega_g^* \dot{\omega}_m - k_e(T_g^{*2} + T_m^{*2}) \tag{47}$$

$$T_g^* = \frac{s}{r+s} h_e(\omega_e^*) + \frac{r}{s} I_g \dot{\omega}_m,\ T_m^* = T_d - \frac{r}{r+s} h_e(\omega_e^*) \tag{48}$$

$$\omega_g^* = \frac{r+s}{s} \omega_e^* - \frac{r}{s} \omega_m \tag{49}$$

Solving the above equation for $\tilde{P}_d$, we get:

$$\tilde{P}_d = U^* - \frac{2P_b^*}{1 + \sqrt{1 + 4\varepsilon_b P_b^*}} \tag{50}$$

Setting the engine power P_e to zero in (44) and letting

$$P_b^0 = -(P_d + (\frac{r}{s})^2 I_g \omega_m \dot{\omega}_m + k_e (\frac{r}{s} I_g \dot{\omega}_m)^2 + k_e T_d^2) \tag{51}$$

gives us:

$$\epsilon_b (U - \tilde{P}_d)^2 + (U - \tilde{P}_d) - P_b^0 = 0 \tag{52}$$

from which the control input U^0 corresponding to the engine off state is:

$$U^0 = \tilde{P}_d + \frac{2P_b^0}{1 + \sqrt{1 + 4\epsilon_b P_b^0}} \tag{53}$$

It can be easily seen that $U^0 < U^*$ since $P_b^* > P_b^0$ and the fact that $\frac{2P}{1+\sqrt{1+4\epsilon_b P}} = \frac{\sqrt{1+4\varepsilon_b P}-1}{2\varepsilon_b}$ is an increasing function of P for $\epsilon_b \neq 0$ and $P > \frac{-1}{4\varepsilon_b}$.

If we assume 100% efficiencies (i.e., $k_e = 0$), $\epsilon = 0$, and constant vehicle speed ($\dot{\omega}_m = 0$), then, we can say that $P_e = U$. More generally we can characterize P_e as a continuous function $\psi : \mathbb{R}^4 \to \mathbb{R}$ of ω_m, $\dot{\omega}_m$, $\tilde{P}_d$, and U according to (44) and (50):

$$P_e = \psi(\omega_m, \dot{\omega}_m, \tilde{P}_d, U) \tag{54}$$

Finally, the discrete state of charge equation is obtained by integrating the state of charge equation in terms of U over a sampling time period τ_s. Therefore, we obtain:

$$\delta S_b = \tau_s(U - \tilde{P}_d) \tag{55}$$

Where δS_b is the change in state of charge over the given sampling period. Now, we define the normalized state of charge, input (as normalized engine power), and normalized power demand as:

$$x = \frac{S_b - s_b^{max}}{S_b^{max}}, u = \frac{\tau_s U}{S_b^{max}}, p = \frac{\tau_s \tilde{P}_d}{S_b^{max}}, u^* = \frac{\tau_s U}{S_b^{max}} \tag{56}$$

we have the discrete state of charge of equation as:

$$x_{k+1} = x_k + u_k(x_k) - p_k \tag{57}$$

which has been used in the subsequent chapter for deriving a closed loop optimal solution. To derive the incremental cost function, let the function $\phi : \mathbb{R}^2 \to \mathbb{R}$ denote the fuel consumption rate in terms of engine speed and torque. Recall that fuel consumption rate can be represented in terms of the mass flow rate $\dot{m}_f$, which is provided by a look up table. Thus, the corresponding cost function given by:

$$L_k(u) = \tau_s \phi(\omega_{ek}, T_{ek}) = \tau_s \phi(\omega_{ek}, h(P_{ek})\omega_{mk}) \tag{58}$$

Also, recall that for the discrete system:

$$P_{ek} = \psi(\omega_{mk}, \dot{\omega}_{mk}, \frac{p_k S_b^{max}}{\tau_s}, \frac{u_k S_b^{max}}{\tau_s}) \tag{59}$$

We shall elucidate the properties of this cost function in the subsequent chapter, which are vital to deriving a closed form optimal solution.

CHAPTER III

OPTIMIZATION USING INTERVAL BACK-PROPAGATION

3.1 Problem Setup

From the previous chapter, it is seen that through an input transformation method, the Power split system of a hybrid electric vehicle vehicle can be modelled using the system expressed in equation (57), which says that:

$x_{k+1} = x_k + u_k(x_k) - p_k$

Here, $x_k, p_k,\ and\ u_k$ refer to the normalized state of charge, engine power, and power demand. The normalized state of charge is constrained between x_{min} and x_{max}, and normalized input lies between u_{min} and u_{max} such that $u_k^{min} \geq\ p_{min}$ & $u_k^{max} \leq\ p_{max}$. Hence, the goal is to find u_k over a time window $k = 0....N-1$ that minimizes the cost function.

$$F = \sum_{k}^{N-1} L_k(u_k(x_k)) \tag{60}$$

Subject to $x_{min} \leq\ x_k \leq\ x_{max},\ u_{min} \leq\ u_k \leq\ u_{max}$

Where L_k is the incremental cost function defined by the input. The following properties must hold true for the incremental cost function:

- The incremental Cost function is Lipschitz Continuous on k. i.e $|L_k(u_2) - L_k(u_1)| \leq\ l|u_2 - u_1|\ \forall [u_1, u_2] \in [u_k^{min}, u_k^{max}]$
- There exists a unique u_k^* that minimizes $L(u_k)/u_k$, and $L(u_k)/u_k$ is non-decreasing on $[u_k^*, u_k^{max}]$

The normalized incremental fuel cost $L(u)/u$ is equivalent to the Brake Specific Fuel Consumption (BSFC) which is minimized at u_k^*. Also, note that the optimal

value of $L(u_k^*)/u_k^*$ is time independent, hence it will be expressed in the cost to go function as $\bar{l}$.

To incorporate for these trends, it is easier if the cost function is modified to remove linear trends in L_k, and create a dead-zone for the non linear component between $[u_k^{min}, u_k^*]$. Hence, we redefine the cost function as:

$$\tilde{L}(u_k(x_k)) = L_k(u_k(x_k)) - \bar{l}u_k \tag{61}$$

From the State Equation (57) we have:

$$u_k(x_k) = x_{k+1} - x_k + p_k \tag{62}$$

Hence, the original Cost Function becomes:

$F = \sum_k^{N-1} \tilde{L}(u_k(x_k)) + \sum_k^{N-1} \bar{l}(x_{k+1} - x_k + p_k)$

The above expression simplifies to:

$$F = \sum_k^{N-1} \tilde{L}(u_k(x_k)) + \bar{l}(x_N - x_0 + \sum_k^{N-1} p_k) \tag{63}$$

Realize that other than the modified incremental cost function and final state, all other quantities are constant and do not affect the optimal control solution. The effect of x_N is to minimize the final state of charge. Recall that our aim is to optimize the amount of consumed fuel from any initial to final state over a moving time window, not to minimize the final state of charge since battery charge can be recoverable for future tasks. Hence, the dependence on x_N is not necessary for our cost function, thereby making our modified cost function:

$$F = \sum_k^{N-1} \tilde{L}(u_k(x_k)) + \bar{l}p_k \tag{64}$$

Hence, our control strategy will involve optimizing $\sum_k^{N-1} \tilde{L}(u_k(x_k))$. Additionally, we need to consider how our constraints will be imposed. The lower bound for the state of charge is a hard constraint, therefore $x \geq 0$. However, the upper bound, defined as $x \leq x_{max}$ is treated as a "soft constraint" and the extra charge is stored in

an auxiliary battery once the main battery reaches maximum charge. The auxiliary battery is only used in cases where the input has to be between u_0 and u_k^*. Denoting the Auxiliary battery charge by z_k and the engine state by e_k, one can use the following charge/discharge strategy to make sure the battery charge lies between 0 and z_{max}:

$$z_{k+1} = z_k + Aux_k(z_k, u_k, e_k) \tag{65}$$

$$e_{k+1} = \begin{cases} 1 & Aux_k(z_k, u_k, e_k) > 0 \ or \ u_k \geq \ u_k^* \\ 0 & otherwise \end{cases} \tag{66}$$

$$Aux_k(z_k, u_k, e_k) = \begin{cases} u_k^* - u_k & if \ u \in (0, u_k^*) \& \ z < u_k^* \ or \ (z \leq \ z_{max} - u_k^* \ \& \ e_k = 1) \\ -u_k & if \ u \in (0, u_k^*) \& \ z > z_{max} - u_k^* \ or \ (z \geq \ u_k^* \ \& \ e_k = 0) \\ 0 & otherwise \end{cases} \tag{67}$$

The above strategy ensures that zero incremental cost is incurred when the engine power input lies between u_0 and u_k^*. It also accommodates for any negative inputs by turning off the engine $i.e. \tilde{L_k}(u_k^*) = 0$ and making use of the mechanical braking action instead of regenerative braking. Hence, without loss of generality, we can define:

$$g_k(u_k(x_k)) = \begin{cases} \tilde{L_k}(u_k(x_k)) & u_k > u_k^* \\ 0 & u_k \leq \ u_k^* \end{cases} \tag{68}$$

Henceforth, the cost function becomes:

$$J = \sum_{k}^{N-1} g_k(u_k(x_k)) \tag{69}$$

Also, from the properties of L, it follows that the function $g_k(u_k(x_k))$ will be non-decreasing, non-negative, and Lipshitz continuous with Lipschitz constant l.

3.2 Pontriyagin's Minimization Principle and Its Shortcomings

A commonly used method from optimal control theory to solve optimization problems is Pontriyagins Minimization principle. Pertaining to the fuel minimization problem,

this method has been used alongside with ECMS to obtain a real time implementable optimal solution [50]. It states that Given a set of differential equations and accompanying initial conditions, find an optimal input within the admissible range of inputs and corresponding optimal trajectory that minimizes the Hamiltonian H and cost Function J. The Hamiltonian essentially consists of the performance index adjoined with Lagrange multipliers λ to constraints imposed by the system dynamics [51, Pg. 51-52]. In our case, the performance index is:

$$L = g_k(u_k(x_k)) \tag{70}$$

while the dynamic constraints are given by our state equation. Additionally, we impose a fixed final state of charge $x_N = s_f$. Therefore, our Hamiltonian is:

$$H_k = g_k(u_k(x_k)) + \lambda_k^T(x_k + u_k - p_k) \tag{71}$$

And our Final Cost is just $\Phi = \mu(x_N - s_f)$ where μ is the associated Lagrange multiplier.

We seek an input u_k which minimizes the Hamiltonian and Cost Function introduced in the Previous section. Thus, from Pontriyagin's Minimum Principle [52, Pg. 48-50], we have the following conditions:

Optimality Condition:

$$\min_{u_{min} \leq\ u_k \leq\ u_{max}} H \tag{72}$$

Co-state Equations:

$$\lambda_k = \frac{\partial H_k}{\partial x_k}^T \lambda_{k+1} \tag{73}$$

$$\lambda_N = \frac{\partial \Phi_k}{\partial x_k}^T (x_N) \tag{74}$$

Using the Above Conditions, we have determined that $\lambda_k = \lambda_N = \mu$. Assuming a k-invariant incremental cost $g_k = g$, it follows that the criteria to be satisfied is

$$\min_{u_{min} \leq\ u \leq\ u_{max}} (g(u) + \mu u)$$

The above equation leads to a constant value of u. Subject to the state equation constraint, the input satisfying our criteria is given by:

$$u_k^*(x_k) = \frac{1}{N-k}\sum_{j=k}^{N-1} p_j + \frac{s_f - x_k}{N-k} \tag{75}$$

provided that $u_k(x_k) \in [u_{min}, u_{max}]$. Recall that we also seek to minimize the cost function J. This is not yet possible to ascertain since the convexity properties of $g_k(u_k)$ have not been established. In addition, we need to ensure that the optimizing inputs for the Hamiltonian satisfy the simultaneous constraints on x_k in $[x_{min}, x_{max}]$ and u_k in $[u_{min}, u_{max}]$. Also, realize that $g_k(u_k)$ is time varying, which is not dealt with appropriately in this method. Furthermore, Pontriyagin's minimization principle only guarantees a locally optimal solution. Therefore, we resort to backward dynamic programming for finding a globally optimal solution. In the following sections, it will be shown that if state constraints are satisfied and $g_k(u_k)$ is convex, then the globally optimal solution is indeed very similar to the one stated above.

3.3 Dynamic Programming Method

The backward search dynamic programming algorithm starts from the final step. At each step, one computes the optimal input which minimizes the incremental cost and cost to go function for a specified state of charge.Subsequently, it updates the cost to go function for the steps to follow.

At the final step, one must find u_{N-1} which minimizes $g(u_{N-1}) = J_{N-1}(u)$ while keeping $x_N \geq 0$ and $u_{min} \leq u_{N-1} \leq u_{max}$. This is given by:

$$u_{N-1}^*(x) = \begin{cases} p_{N-1} - x & x < p_{N-1} - u^* \\ min(u^*, p_{N-1} - x + x_{max}) & x = p_{N-1} - u^* \\ min(u_0, p_{N-1} - x + x_{max}) & x > p_{N-1} - u^* \end{cases} \tag{76}$$

Subsequently, the cost to go function is given by:

$$J_{N-1}^*(x) = g(p_{N-1} - x) \tag{77}$$

Now, Let's define the following variables to simplify our notation:

$p_k^* = min(p_k, u_k^*),\ \tilde{p_k} = p_k - p_k^*,\ q_k = \sum_{l=k}^{N-1} \tilde{p_l},\ \tilde{q_k} = q_{k+1} - p_k,\ and\ \xi = q_k - x$

Let f_k be a function sequence in $(-\infty,\ q_k] \rightarrow \mathbb{R}^+$ satisfying:

$$f_k(\xi) = \inf_{u \in [\xi - \tilde{q_k},\ u_{max}^k] \cap [u_{min}^k,\ \xi - \tilde{q_k} + x_{max}]} (g_k(u) + f_{k+1}(\xi - u + p_k^*))\ \forall \xi \in (q_k - x_{max}, q_k) \quad (78)$$

where $f_{N-1}(\xi) = g_{N-1}(\xi + p_{N-1}^*)$. We want to show that $J_k^*(x) = f_k(\xi)$ through mathematical induction. First, we establish the case for k=N-1, which has been shown earlier. We now assume that the relationship holds for k, and then prove that the same relationship holds true for k+1.

Let us define $\mathcal{U}$ as follows:

$\mathcal{U}_k(x) = [p_k - x,\ u_{max}^k] \cap [u_{min}^k,\ p_k - x + x_{max}] = [\xi - \tilde{q_k},\ u_{max}^k] \cap [u_{min}^k,\ \xi - \tilde{q_k} + x_{max}]$

Therefore, for $x \in [0,\ x_{max}]$ we have:

$$J^*(x) = \inf_{u \in \mathcal{U}(x)} (g_k(u) + J_{k+1}^*(x + u - p_k)) \quad (79)$$

Recall that $J_{k+1}^*(x) = f_k(q_{k+1} - x)$. Therefore, we have

$J_k^* = \inf_{u \in \mathcal{U}(x)} (g_k(u) + f_{k+1}(q_{k+1} - x - u + p_k))$

Note that $q_{k+1} = \sum_{l=k+1}^{N-1} \tilde{p_l} = \sum_{l=k}^{N-1} \tilde{p_l} - p_k + p_k^*$, which gives us

$J_k^* = \inf_{u \in \mathcal{U}(x)} (g_k(u) + f_{k+1}(q_k - x - u + p_k^*))$

Substituting $\xi = q_k - x$, we obtain

$$J_k^* = \inf_{u \in \mathcal{U}(x)} (g_k(u) + f_{k+1}(\xi - u + p_k^*)) = f_k(\xi) \quad (80)$$

Therefore, we can say that the optimal cost to go function in equation (80) is entirely described by the properties of $f_k(\xi)$. Hence, we need to make sure that $f_k(\xi)$ satisfies the following desirable properties to ensure convergence of cost to a minimal value:

1. Each function $f_k(\xi)$ is non decreasing i.e. $f_k(\xi_1) \leq\ f_k(\xi_2),\ \forall \xi_1 \leq\ \xi_2 \leq\ q_k$

2. Each function $f_k(\xi)$ is Lipschitz Continuous with the same Lipschitz constant l

3.3.1 Proof of Non-decreasing property for $f(\xi)$

The proof for showing that $f_k(\xi)$ is non decreasing is carried out by mathematical induction. at k=N-1, we see that:

$f_{N-1}(\xi) = g_{N-1}(\xi + p^*_{N-1})$

In Addition, we know that since $p_{N-1} > u^*_{N-1}$:

$p^*_{N-1} = min(p_{N-1}, u^*_{N-1}) = u^*_{N-1}$

Therefore:

$$f_{N-1}(\xi) = g_{N-1}(\xi + u^*_{N-1}) \tag{81}$$

The above function is clearly non decreasing owing to the properties of g_k. Now, suppose that $f(\xi)_l, l \in [k+1, N+1]$ is non decreasing, we need to consider the following cases:

1. $\xi_2 \geq \xi_1 \geq \tilde{q_k} - x_{max} + u_{max}$

   From the induction hypothesis, we have:

   $$f_k(\xi_1) = \inf_{u \in [\xi_1 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]} (g_k(u) + f_{k+1}(\xi_1 - u + p^*_k)) \tag{82}$$

   Recall that since $\xi_2 - \tilde{q_k} \geq \xi_1 - \tilde{q_k}$:

   $f_k(\xi_1) \leq \inf_{u \in [\xi_2 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]} (g_k(u) + f_{k+1}(\xi_1 - u + p^*_k))$

   Thus, we can say with certainty that:

   $$f_k(\xi_1) \leq \inf_{u \in [\xi_2 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]} (g_k(u) + f_{k+1}(\xi_2 - u + p^*_k)) = f_k(\xi_2) \tag{83}$$

   Hence, we have proved that $f_k(\xi_1) \leq f_k(\xi_2)$

2. $\xi_1 \leq \xi_2 \leq \tilde{q_k} - x_{max} + u_{max}$

   We see that $f_k(\xi_2)$ can be represented as follows:

   $$f_k(\xi_2) = \inf_{u \in [u_{min},\ \xi_2 - \tilde{q_k} + x_{max}]} (g_k(u) + f_{k+1}(\xi_2 - u + p^*_k)) \tag{84}$$

Note that the intersection has been omitted since a hard constraint is imposed as $\xi_2 - \tilde{q_k} + x_{max} < u_{max}$. Now, lets consider $u_2^* \in [u_{min},\ \xi_2 - \tilde{q_k} + x_{max}]$, and $u_1 := \xi_1 - \tilde{q_k} + x_{max} \leq\ u_2^*$. We see that $f_k(\xi_2)$ is simplified to:

$$f_k(\xi_2) = g_k(u_2^*) - f_{k+1}(\xi_2 - u_2^* + p_k^*)) \tag{85}$$

Representing $f_k(\xi_1)$ as: $f_k(\xi_1) = \inf_{u \in [u_{min},\ \xi_1 - \tilde{q_k} + x_{max}]} (g_k(u) + f_{k+1}(\xi_1 - u + p_k^*))$
Note that $f_k(\xi_1)$ is bounded by u_1, hence it must satisfy:

$$f_k(\xi_1) \leq\ g_k(u_1) + f_{k+1}(\xi_1 - u_1 + p_k^*)) \tag{86}$$

Finally, from the inequality we can confirm that:

$$f_k(\xi_1) \leq\ g_k(u_2^*) + f_{k+1}(\xi_2 - u_2^* + p_k^*)) \leq\ f_k(\xi_2) \tag{87}$$

Proving that f_k is non decreasing owing to the non decreasing nature of g_k

3. $\xi_1 \leq\ \tilde{q_k} - x_{max} + u_{max}$ *and* $\xi_2 \geq\ \tilde{q_k} - x_{max} + u_{max}$

   Let $\eta_1 = \tilde{q_k} - x_{max} \leq\ \xi_2$, then from case 1 we have that $f(\eta_1) \leq\ f(\xi_2)$. Similarly, if $\xi_1 \leq\ \eta_1 = \tilde{q_k} - x_{max}$, then from case 2 we have $f(\xi_1) \leq\ f(\eta_1)$. Therefore, one can confirm that $f_k(\xi_1) \leq\ f_k(\xi_2)$.

3.3.2 Proof of Lipschitz Continuity for $f(\xi)$

The proof for showing that $f_k(\xi)$ is Lipschitz Continuous with Lipschitz constant l is also carried out by induction. At k=N-1, we know:

$$f_{N-1}(\xi) = g_{N-1}(\xi + u_{N-1}^*) \tag{88}$$

The above is Lipschitz continuous owing to the properties of $g_k(u_k)$. Now, suppose $f(\xi)_l, l \in [k+1, N+1]$ is Lipschitz continuous on $[-\infty, q_k]$, we would like to show that :

$$f(\xi_2) - f(\xi_1) \leq\ l(\xi_2 - \xi_1)\ \forall \xi_1, \xi_2 \in [-\infty, q_k]\ \xi_1 \leq\ \xi_2 \tag{89}$$

There are 3 cases arising from the possibility stated above for which lipschitz continuity must be established, which are identical to the criteria mentioned for satisfying the non decreasing property.

1. $\xi_2 \geq \ \xi_1 \geq \ \tilde{q_k} - x_{max} + u_{max}$

   Recall that $f_k(\xi_1)$ is represented as:

   $f_k(\xi_1) = \inf_{u \in [\xi_1 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]} (g_k(u) + f_{k+1}(\xi_1 - u + p^*_k))$

   Now, lets introduce $u^*_1 \in [\xi_1 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]$. Hence, $f_k(\xi_1)$ simplifies to:

   $$f_k(\xi_1) = g_k(u^*_1) + f_{k+1}(\xi_1 - u^*_1 + p^*_k) \tag{90}$$

   Also, note that $f_k(\xi_2)$ can be represented as:

   $f_k(\xi_2) = \inf_{u \in [\xi_2 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]} (g_k(u) + f_{k+1}(\xi_2 - u + p^*_k))$

   Suppose that $u^*_1 \in [\xi_2 - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ u^k_{max}]$, we can say that $f_k(\xi_1)$ is bounded by u^*_1, thereby it must satisfy

   $$f_k(\xi_2) \geq \ g_k(u^*_1) + f_{k+1}(\xi_2 - u^*_1 + p^*_k) \tag{91}$$

   Note that by the lipschitz condition, $f_{k+1}(\xi_2 - u^*_1 + p^*_k) \geq \ f_{k+1}(\xi_1 - u^*_1 + p^*_k) + l(\xi_2 - \xi_1)$. Hence, we get:

   $$f_k(\xi_2) \geq \ g_k(u^*_1) + f_{k+1}(\xi_1 - u^*_1 + p^*_k) + l(\xi_2 - \xi_1) = f_k(\xi_1) \tag{92}$$

   This proves that $f(\xi_2) - f(\xi_1) = l(\xi_2 - \xi_1)$

   If instead, $u^*_1 \in [\xi_1 - \tilde{q_k},\ \xi_2 - \tilde{q_k}]$, then let $u_2 = \xi_2 - \tilde{q_k} \geq \ u^*_1$. Then, we see that by lipschitz continuity of g_k

   $$g_k(u_2) = g_k(u^*_1) + l(u_2 - u^*_1) \tag{93}$$

   Also, note that since $\xi_1 - u^*_1 \leq \ \tilde{q_k} = \xi_2 - u_2$ we have the following by induction hypothesis:

$f_{k+1}(\xi_2 - u_2 + p_k^*) = f_{k+1}(\xi_1 - u_1^* + p_k^*) + l(\xi_2 - xi_1 - u_1^* + u_2)$

Also note that

$f_k(\xi_2) \leq\ g(u_2) + f_{k+1}(\xi_2 - u_2 + p_k^*)$

From convexity of g_k, we have that

$f_k(\xi_2) \leq\ g(u_1^*) + f_{k+1}(\xi_2 - u_2 + p_k^*) + l(u_2 - u_1^*) f_k(\xi_2)$
$\leq g(u_1^*) + f_{k+1}(\xi_1 - u_1 + p_k^*) + l(\xi_2 - \xi_1) f_k(\xi_2)$

$$f_k(\xi_2) \leq\ f_k(\xi_1) + l(\xi_2 - \xi_1) \tag{94}$$

Therefore, we have once again proved that $f(\xi_2) - f(\xi_1) = l(\xi_2 - \xi_1)$

2. $\xi_1 \leq\ \xi_2 \leq\ \tilde{q_k} - x_{max} + u_{max}$

   We see that $f_k(\xi_2)$ can be represented as follows:

$$f_k(\xi_2) = \inf_{u \in [u_{min},\ \xi_2 - \tilde{q_k} + x_{max}]} (g_k(u) + f_{k+1}(\xi_2 - u + p_k^*))$$
$$f_k(\xi_2) \leq \inf_{u \in [u_{min},\ \xi_1 - \tilde{q_k} + x_{max}]} (g_k(u) + f_{k+1}(\xi_2 - u + p_k^*))$$
$$f_k(\xi_2) \leq \inf_{u \in [u_{min},\ \xi_1 - \tilde{q_k} + x_{max}]} (g_k(u) + f_{k+1}(\xi_1 - u + p_k^*)) + l(\xi_2 - \xi_1)$$

$$f_k(\xi_2) \leq f_k(\xi_2) \leq\ f_k(\xi_1) + l(\xi_2 - \xi_1) \tag{95}$$

   Thereby completing the proof

3. $\xi_1 \leq\ \tilde{q_k} - x_{max} + u_{max}$ *and* $\xi_2 \geq\ \tilde{q_k} - x_{max} + u_{max}$

   Let $\eta_1 = \tilde{q_k} - x_{max} + u_{max} \leq\ \xi_2$, then from case 1 we have that $f(\xi_2) - f(\eta_1) = l(\xi_2 - \eta_1)$. Similarly, if $\xi_1 \leq\ \eta_1 = \tilde{q_k} - x_{max} + u_{max}$, then from case 2 we have $f(\eta_1) - f(\xi_1) = l(\eta_1 - \xi_1)$. Therefore, one can confirm that $f(\xi_2) - f(\xi_1) = l(\xi_2 - \xi_1)$.

3.4 Convexity of Cost Function

The solution to our optimal control problem can be expressed in closed form for all set of states within the defined constraints if an additional convexity assumption is

imposed for the incremental cost function L_k. Note that convexity of a function implies the following.

For some $\lambda \in [0,1]$, we have that $\forall [u,v] \in [a,b]$:

$$g(\lambda u + (1-\lambda)v) \leq \ \lambda g(u) + (1-\lambda)g(v) \tag{96}$$

The following Lemma indicates that a non decreasing and lipschitz continuous cost to go function $f_k(\xi)$ inherits the convexity properties of g_k.

Lemma 1: The incremental cost function g_k and non decreasing cost to go function $f_k(\xi)$ are convex on $(-\infty,\ u_k^{max})$ and $(-\infty,\ q_k)$ respectively if L_k is convex on $[u_k^*,\ u_k^{max}]$

Proof of Lemma 1:

If $u \geq \ u^*$ or $v \leq \ u^*$, then by virtue of convexity in L_k, we have convexity for g_k, since $g_k(u_k) = \tilde{L}(u_k(x_k))$ over the specified domain, and $\tilde{L}(u_k(x_k)) = L_k(u_k(x_k)) - \bar{l}u_k$. Otherwise, if $u \leq \ u^*$ or $v \geq \ u^*$ then $g_k(u_k) = 0$, which also satisfies the convexity assumption. Suppose that $u \leq \ u_k^* < v \leq \ 1$, then we know that $g_k(u) = 0$. In addition, due to the bounds on u and v:

$g_k(\lambda u + (1-\lambda)v) \leq \ g_k(\lambda u_k^* + (1-\lambda)v)$

$g_k(\lambda u + (1-\lambda)v) \leq \ \tilde{L_k}(\lambda u_k^* + (1-\lambda)v)$

Owing to the convexity of L_k and subsequently $\tilde{L}(u_k(x_k))$, we have that:

$g_k(\lambda u + (1-\lambda)v) \leq \ \lambda \tilde{L_k}(u_k^*) + (1-\lambda)\tilde{L_k}(v)$

Henceforth, we can establish that g_k is convex since the above expression simplifies to the convexity criteria shown below:

$$g_k(\lambda u + (1-\lambda)v) \leq \ \lambda g_k(u_k^*) + (1-\lambda)g_k(v) \tag{97}$$

From 3.3.1, it is seen that f_k is decreasing. Also, note that due to the convexity for g_k, we can establish that $f_{N-1}(\xi) = g_{N-1}(\xi + u_{N-1}^*)$ is convex for case k=N-1. For the remaining cases, we shall use mathematical induction to ascertain convexity. Suppose

that f_l $l \in [k, N-1]$ is convex on $[-\infty,\ g_k]$, for $k \geq\ 1$. We should now show that $f_{k-1}(\xi)$ is convex :

$$f_{k-1}(\xi) = \inf_{u \in [\xi - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ \xi - \tilde{q_k} + x_{max}]} (g_{k-1}(u) + f_k(\xi - u + p^*_{k-1})) \tag{98}$$

Let us consider that for a given $\xi_s \in (-\infty,\ q_k]$ we have $u^*_s \in [\xi - \tilde{q_k},\ u^k_{max}] \cap [u^k_{min},\ \xi - \tilde{q_k} + x_{max}]$ s=1,2 which is the optimal input for $g_k(u) - f_k(\xi_s - u + p^*_{k-1})$. Then, owing to the compact input search domain and lipschitz continuity of g_k and f_k (proof 3.3.2), we have that:

$$f_{k-1}(\xi_s) = g_{k-1}(u^*_s) + f_k(\xi_s - u^*_s + p^*_{k-1}) \tag{99}$$

Now, lets define $u_\lambda = \lambda u^*_1 + (1-\lambda)u^*_2$ and $\xi_\lambda = \lambda \xi^*_1 + (1-\lambda)\xi^*_2$ for some $\lambda \in [0, 1]$. Then, we have that $u^*_\lambda \in [\xi - \tilde{q_k},\ u^k_{max}] \cap [0,\ \xi - \tilde{q_k} + x_{max}]$. Also, observe that $f_{k-1}(\xi_\lambda)$ becomes:

$$f_{k-1}(\xi_\lambda) = g_{k-1}(u_\lambda) + f_k(\xi_\lambda - u_\lambda + p^*_{k-1}) \tag{100}$$

By the convexity g_k and f_k, one obtains that

$$f_{k-1}(\xi_\lambda) \leq\ \lambda g_{k-1}(u^*_1) + (1-\lambda) g_{k-1}(u^*_2) + \lambda f_k(\xi_1 - u^*_1 + p^*_{k-1})$$
$$+ (1-\lambda) f_k(\xi_2 - u^*_2 + p^*_{k-1})$$
$$f_{k-1}(\xi_\lambda) \leq\ \lambda(g_{k-1}(u^*_1) + f_k(\xi_1 - u^*_1 + p^*_{k-1})) + (1-\lambda)(g_{k-1}(u^*_2) + f_k(\xi_2 - u^*_2 + p^*_{k-1}))$$

$$f_{k-1}(\xi_\lambda) \leq\ \lambda f_k(\xi_1) + (1-\lambda) f_k(\xi_2) \tag{101}$$

Hence, we have proven that g_k is convex owing to convexity in L_k, and that f_k inherits the convexity properties of g_k. Now, since the convexity of g_k is confirmed, we can derive a closed form globally optimal solution using dynamic programming.

3.5 Globally optimal solution using Interval Back propagation

To formulate a closed form globally optimal solution, let us initially assume a time invariant cost to go function $g_k(u) = g(u)$. Let $\nu := max(0 \leq\ k \leq\ N : p_{N-1} > u^*$.

Then, $\alpha_k \in [0, x_{max}]$ is the state that can be transferred to $x_\nu = 0$ through $u_k = u^*$ using the backward recursion equation shown below starting at $\alpha_\nu = 0$.

$$\alpha_{k-1} = \alpha_k + u^* - p_{k-1} \tag{102}$$

Hence, at the initial step $k = \nu$, our optimal control can be described using the results from section 3.2. For $x \in [\alpha_{\nu-1}, x_{max}]$, the optimal control is $u_k^* = u_0$ or $u_k^* = u^*$ while the cost to go is zero. Meanwhile, for $x \in [0, \alpha_{\nu-1}]$ we have $u_k^* = p_{\nu-1} - x$.

Now, lets consider the case where $k < \nu$. As discussed before, the optimal control sequence is $u_l = p_l - x \; l \in [k, \nu - 1]$. Also, recall from 3.2 that since the cost-to-go may be nonzero, we need to satisfy the additional constraint $\sum_{l=k}^{\nu-1} g(u_l)$ must be minimized. Due to the convexity of g, we see that:

$\sum_{l=k}^{\nu-1} g(u_l) \geq \; (\nu - k) g(\frac{\sum_{l=k}^{\nu-1} u_l}{\nu-k}) = (\nu - k) g(\frac{\sum_{l=k}^{\nu-1} p_l - x}{\nu-k})$

Defining $\overline{p}_k^\nu = \frac{\sum_{l=k}^{\nu-1} p_l}{\nu-k}$, we have:

$$\sum_{l=k}^{\nu-1} g(u_l) \geq \; (\nu - k) g(\overline{p}_k^\nu - \frac{x}{\nu - k}) = J_k^*(x) \tag{103}$$

The equation above represents the cost to go. In addition, one can see that the optimal input sequence is a set of equal inputs, which are given by:

$$u_k = u_{k+1} = \cdots = u_{\nu-1} = \overline{p}_k^\nu - \frac{x}{\nu - k} \tag{104}$$

Knowing the nature of our optimal input sequence and optimal cost, we will now employ the backward recursion algorithm to obtain a more generalized form for the globally optimal solution

3.5.1 Algorithm

Let the lower bound for all trajectories resulting from the optimal input and cost-to -go functions derived in the previous section be denoted by $\alpha_k^{'} \in [0, x_{max}]$. Notice that this lower bound itself is an optimal trajectory when inside $[0, x_{max}]$, and reaches

zero prior to $k = \nu$. Henceforth, the backward recursion of $\alpha_k^{'}$ starting at $\alpha_{\nu-1}^{'} = 0$ is given by $\alpha_{k-1}^{'} = max(0, \beta_k^{'})$ where:

$\beta_k^{'} = \alpha_k^{'} - (\overline{p}_k^{\nu} - \frac{\alpha_k^{'}}{\nu-k}) + p_{k-1}$

Simplifying, we have:

$$\beta_k^{'} = \frac{\nu - k + 1}{\nu - k}\alpha_k^{'} - \overline{p}_k^{\nu} + p_{k-1} \tag{105}$$

The intervals $[0, \alpha_k^{'}][\alpha_k^{'}, \alpha_k]$ *and* $[\alpha_k, x_{max}]$ which partition $[0, x_{max}]$ are relevant provided that $0 \leq \alpha_k^{'} \leq \alpha_k \leq x_{max}$. Else, they are discarded. Also, note that new lower bounds are added at each step k.

Now lets consider the case where α_k exceeds x_{max}. A new upper bound, denoted by $\overline{\alpha}_k$ is generated at $k = \mu$. These optimal trajectories are distinct from those that reach the lower bound at $k = \nu$. Using backward recursion, one can represent $\overline{\alpha}_{k-1} = max(\overline{\beta}_k, x_{max})$ where:

$$\overline{\beta}_k = \frac{\nu - k + 1}{\nu - k}\overline{\alpha}_k - \overline{p}_k^{\nu} + p_{k-1} \quad \overline{\alpha}_{\mu} = 0 \tag{106}$$

Similar to the arguement presented for the lower bound, one can say that for $x \in (\overline{\alpha}_k, x_{max})$ the optimal control sequence and cost to go functions are:

$$u_k^* = \overline{p}_k^{\mu} - \frac{x - x_{max}}{\mu - k} \tag{107}$$

$$J_k^*(x) = (\mu - k)g(\overline{p}_k^{\mu} - \frac{x - x_{max}}{\mu - k}) + J_{\mu}^*(x) \tag{108}$$

The upper bounds for all trajectories resulting from the optimal input and cost to go functions presented above are denoted by $\overline{\alpha}_k^{'} \in [0, x_{max}]$. Notice that this upper bound itself is an optimal trajectory when inside $[0, x_{max}]$, and reaches x_{max} prior to $k = \mu$. Henceforth, the backward recursion of $\overline{\alpha}_k^{'}$ starting at $\overline{\alpha}_{\mu-1}^{'} = 0$ is given by $\overline{\alpha}_{k-1}^{'} = max(0, \overline{\beta}_k^{'})$ where:

$$\overline{\beta}_k^{'} = \frac{\mu - k + 1}{\mu - k}(\overline{\alpha}_k^{'} - x_{max}) - \overline{p}_k^{\mu} + p_{k-1} \quad \overline{\alpha}_{\mu-1}^{'} = x_{max} \tag{109}$$

Additional upper bounds are added at every instant where $k > \mu$. The backward recursion algorithm of our controller presented above can be generalized into the following steps.

1. Initialize the set of interval boundaries, their starting values, their time steps, and the value of their predecessors to $\mathcal{A}_N = [0, x_{max}],\ \mathcal{N}_N = [N, -N],\ \mathcal{M}_N = [\infty, -\infty]$

2. For $k = N, N-1, \ldots, 2$, Update $\mathcal{A}_N,\ \mathcal{N}_N,\ and\ \mathcal{M}_N$ as follows:

   (a) Let α_k^j and $m_k^j,\ j \in [1, j_k^{max}]$ be the j$^{\text{th}}$ members of $\mathcal{A}_N,\ \mathcal{M}_N$ respectively. Let $\nu = |m_k^j|,\ \&\ \sigma = \frac{1-sgn(m_k^j)}{2}$. Compute:

$$\beta_j^k = \frac{\nu - k + 1}{\nu - k}\alpha_k^j - \frac{\sigma}{\nu - k}x_{max} - \overline{p}_k^{\nu} + p_{k-1} \tag{110}$$

   (b) Let $j_k^{max} = |\mathcal{A}_k\ \mathcal{J}_k = [1, j_k^{max}]$ Defining $\mathcal{J}_k^+ = [j_k \in \mathcal{J}_k : \beta_k^{j_k} > 0],\ and\ \mathcal{J}_k^- = [j_k \in \mathcal{J}_k : \beta_k^{j_k} \leq\ 0]$ we evaluate:

$$j_k^+ = \begin{cases} max(\mathcal{J}_k^-) & \mathcal{J}_k \neq \varnothing \\ 0 & \mathcal{J}_k = \varnothing \end{cases} \qquad j_k^- = \begin{cases} min(\mathcal{J}_k^+) & \mathcal{J}_k \neq \varnothing \\ j_k^{max} + 1 & \mathcal{J}_k = \varnothing \end{cases} \tag{111}$$

   (c) Update $\mathcal{A}_N,\ \mathcal{N}_N,\ and\ \mathcal{M}_N$ as $\mathcal{A}_N = [0, \beta_k^{j_k^- + 1}, \ldots, \beta_k^{j_k^+ - 1}, x_{max}],\ \mathcal{N}_N = [k-1, n_k^{j_k^- + 1}, \ldots, n_k^{j_k^+ - 1}, 1-k],\ and\ \mathcal{M}_N = [m_{min}, m_k^{j_k^- + 1}, \ldots, m_k^{j_k^+ - 1}, m_{max}]$ where:

$$m_{min} = \begin{cases} m_k^{j_k^-} & n_k^{j_k^-} \geq\ 0 \\ n_k^{j_k^-} & n_k^{j_k^-} < 0 \end{cases} \qquad m_{max} = \begin{cases} m_k^{j_k^+} & n_k^{j_k^+} < 0 \\ n_k^{j_k^+} & n_k^{j_k^+} \geq\ 0 \end{cases} \tag{112}$$

   Where $n_k^0 = m_k^0 = n_k^1$ and $n_k^{j_k^{max}+1} = m_k^{j_k^{max}+1} = n_k^{j_k^{max}}$. An illustration of how m and n are quantified will be presented in the following section

3. The optimal control law $u_k^*(x)$ for $x \in [\alpha_k^j, \alpha_k^{j+1}]$ is given by

$$u_k^*(x) = \begin{cases} min(u_{opt}, x_{max} - x + p_k) & \nu = \infty \\ \overline{p}_k^{\nu} - \frac{x - \sigma x_{max}}{\nu - k} & \nu < \infty \end{cases} \tag{113}$$

Where

$$\nu = min(|m_k^j|, |m_k^{j+1}|)$$

$$u_{opt} = \begin{cases} u_0 & x > \alpha_k^j + u^* \\ u^* & otherwise \end{cases}$$

$$\sigma = \begin{cases} 0 & |m_k^j| \leq |m_k^{j+1}| \\ 1 & |m_k^j| > |m_k^{j+1}| \end{cases}$$

Now, suppose we are in the free space region which lies in between the top lower bound trajectory, and bottom upper bound trajectory. Then, we have $\mathcal{M}_N = [\infty, -\infty]$, and so $\nu = \infty$. Therefore, the optimal solution is one that switches between u_{opt} & u^* provided that $x - p_k < x_{max}$. Our switching criteria is identical to the one shown for u_{opt}. In a physical sense we saying that if the state of charge exceeds the top interval bound α_k^1 by a value greater than the optimal normalized engine power, then the engine must be switched off. Else, the engine must operate at optimal normalized engine power to adequately charge the battery.

3.5.2 Optimal input and Optimal Cost to go derivation

The following lemma states some key properties of the intervals spanned by $\mathcal{A}_k$ and the general form of the optimal input and optimal cost to go based off those properties

Lemma 2: Consider the sets $\mathcal{A}_N$, $\mathcal{N}_N$, *and* $\mathcal{M}_N$ as generated in the above algorithm. Let $\mu = min(|m_{k_0}^j|, |m_{k_0}^{j+1}|)$, and $\sigma_\mu = \frac{1 - sgn(m_{k_0}^j)}{2}$ for $k_0 \in [0N]$. Then:

1. There exists a $j_k \in [1, j_k^{max}]$ such that $|m_k^{j_k}| = |m_k^{j_k+1}| = \mu$ or $|m_k^{j_k - \sigma_\mu}| = |n_k^{j_k+1-\sigma_\mu}| = \mu$ for $k \in [k_0, \mu - 1]$ such that β_k^j introduced in the algorithm is

given by:

$$\beta_k^{j_k+i} = \frac{\mu-k+1}{\mu-k}\alpha_k^{j_k+i} - \frac{\sigma_\mu}{\mu-k}x_{max} - \overline{p}_k^{\mu} + p_{k-1}\ i = 0, 1 \qquad (114)$$

Proof:

The Lemma statement above will be proved using mathematical induction. Let us start with $k = \mu - 1$. In this case one sees that for $j_k = 1 + \sigma_k(j_k^{max} - 2)$, one obtains that $m_k^1 = n_k^2$ & $\mu = min(|m_{k_0}^1|, |m_{k_0}^2|)$ for $\sigma_k = 0$ and $_k^{j_k^{max}} =_k^{j_k^{max}}$ & $\mu = min(|m_{k_0}^{j_k^{max}}|, |m_{k_0}^{j_k^{max}+1}|)$ for $\sigma_k = 1$. Therefore, one can say that $|m_k^{j_k-\sigma_\mu}| = |n_k^{j_k+1-\sigma_\mu}| = \mu$. Subsequently, since $\alpha_k^{j_k+1-\sigma\mu} = 0$ we also have that: $\beta_k^{j_k+1-\sigma\mu} = p_{k-1} + \sigma_\mu x_{max}$

Suppose that $|m_k^{j_l-\sigma_\mu}| = \mu$ and $|n_k^{j_l+1-\sigma_\mu}| = \mu$ or $|m_k^{j_l+1-\sigma_\mu}| = \mu$ for $\mu - 1 \leq\ l \leq k+1$ and $1 \leq\ j_l \leq\ j_l^{max}$. we shall show that the assertion above is true for l=k. If $[\beta_{k+1}^{j_{k+1}}, \beta_{k+1}^{j_{k+1}+1}] \in [0, x_{max}]$ then we have that $\alpha_k^{j_k} = \beta_{k+1}^{j_{k+1}}$ & $\alpha_k^{j_k+1} = \beta_{k+1}^{j_{k+1}+1}$. Therefore, $m_k^{j_l+1-\sigma_\mu}\ n_k^{j_l+1-\sigma_\mu}\ n_k^{j_l-\sigma_\mu}\ m_k^{j_l-\sigma_\mu}$ are determined as explained in the algorithm step 2 c by equations (112-113).

Suppose that $\beta_{k+1}^{j_{k+1}} \leq\ 0$ & $\beta_{k+1}^{j_{k+1}+1} \in [0, x_{max}]$, then we say that $n_{k+1}^{j_{k+1}} = n_{k+1}^{j_{k+1}+1}$, $n_k^{j_k} = k$, $m_{k+1}^{j_{k+1}} = m_{k+1}^{j_{k+1}+1}$ and $m_k^{j_k} = \sigma_\mu n_{k+1}^{j_{k+1}} + (1-\sigma_\mu)m_{k+1}^{j_{k+1}}$. In other words, if the interval bound trajectory violates our lower bound criteria, then $n_{k+1}^{j_{k+1}}$ and $m_k^{j_k}$ are assigned to their previous values, unless the interval bound trajectory approaches from the upper bound, in which case $\sigma_\mu = 1$ and so $m_k^{j_k} = n_{k+1}^{j_{k+1}}$. Similarly, if $\beta_{k+1}^{j_{k+1}} \in [0, x_{max}]$ & $\beta_{k+1}^{j_{k+1}+1} \geq\ x_{max}$ then $n_k^{j_k} = n_{k+1}^{j_{k+1}}\ n_k^{j_k+1} = -k\ m_k^{j_k} = m_{k+1}^{j_{k+1}}$ while $m_{k+1}^{j_{k+1}} = \sigma_\mu m_{k+1}^{j_{k+1}+1} + (1-\sigma_\mu)n_{k+1}^{j_{k+1}+1}$. Hence, $m_{k+1}^{j_{k+1}} = n_{k+1}^{j_{k+1}+1}$ and n_k^j is assigned to its previous value if the interval bound trajectory violates our upper bound criteria, unless that interval bound trajectory is approaching from the lower bound. In such a case, $m_{k+1}^{j_{k+1}}$ is assigned to its previous value as $\sigma_\mu = 0$. A clearer illustration of this has been provided in figure 9:

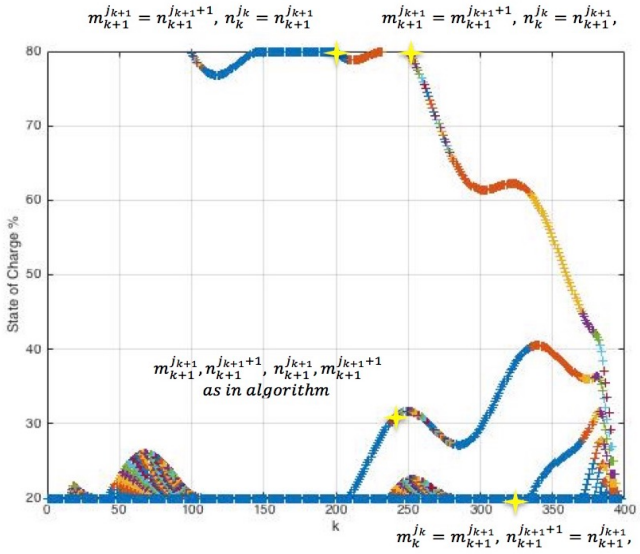


Figure 10: Illustration of m's and n's

Observe that for the above cases, our algorithm generates j_k which at most takes on a value of $j_{k+1}+1$ depending upon the constraints. Also, note that the two cases $[\beta_{k+1}^{j_{k+1}}, \beta_{k+1}^{j_{k+1}+1}] \leq 0$ & $[\beta_{k+1}^{j_{k+1}}, \beta_{k+1}^{j_{k+1}+1}] \geq x_{max}$ have not been taken into consideration, as they violate $\mu = min(|m_{k_0}^{j}|, |m_{k_0}^{j+1}|)$. Hence, we can say that $|m_k^{j_k+\sigma_\mu}| = \mu$, for $k = k_0 \dots \mu$. In addition, due to constancy in $m_k^{j_k+\sigma_\mu}$ arising from algorithm 1, we can say that the above lemma is proven for $i = \sigma_\mu$. According to Algorithm 1, one can also say there exists a $k1 \geq k_0$ for which $|n_k^{j_k+1-\sigma_\mu}| = \mu$, for $k = k_1 + 1 \dots \mu$, and provided that $k1 \neq k_0$ we also have that $|m_k^{j_k+1-\sigma_\mu}| = \mu$ for $k = k_0 \dots k_1$.

To establish the lemma for $i = 1 - \sigma\mu$, we consider $x_k = \alpha_k^{j_k+1-\sigma\mu}$ and define $\nu = |m_k^{j_k+1-\sigma_\mu}|$. At $k = \mu - 1$, we know that $u = \overline{p}_\mu^{nu}$ is the optimal input for $x_{\nu-1}$ based on our earlier assertions. Now, suppose that

$$x_{l-1} = \frac{\mu - l + 1}{\mu - l} x_l + \frac{\sigma_{nu} x_{max}}{\mu - l} + p_{l-1} - \overline{p}_l^{\mu} \; l \leq k + 1 \tag{115}$$

for some $k_1 \leq k \leq \mu$. One can see that in the case where $l = k + 1$ we have that:

$x_k = \frac{\mu-k}{\mu-k-1}x_{k+1} - \frac{\sigma_\nu x_{max}}{\mu--k-1} + p_k - \overline{p}^{\mu}_{k+1}$

The above equation can be represented in discrete differential equation form as:

$x_{k+1} = \frac{(\mu-k-1)(x_k-p_k+\overline{p}^{\mu}_{k+1})-\sigma_\nu x_{max}}{\mu-k}$

Simplifying, we have:

$x_{k+1} = x_k + \frac{\sigma_\nu x_{max}-x_k}{\mu-k} + \frac{\mu-x_k}{\mu-k}(\overline{p}^{\mu}_{k+1} - p_k)$

Now, noting that $\frac{\mu-x_k}{\mu-k}(\overline{p}^{\mu}_{k+1} - p_k) = \overline{p}^{\mu}_{k} - p_k$ one gets:

$$x_{k+1} = x_k + \overline{p}^{\mu}_{k} + \frac{\sigma_\nu x_{max} - x_k}{\mu - k} - p_k \tag{116}$$

Also, from our algorithm we have concluded that $m_k^{j_k+1-\sigma_\mu}$ and $n_k^{j_k+1-\sigma_\mu}$. Hence, it follows that $0 < \beta_k^{j_k+1-\sigma_\mu} < x_{max}$ and therefore $x_{k-1} = \alpha_{k-1}^{j_{k-1}+1-\sigma_\mu} = \beta_k^{j_k+1-\sigma_\mu}$ for $k = k_1 + 2, \ldots, \mu$. Therefore, from the algorithm we have:

$x_{k-1} = \frac{\nu-k+1}{\nu-k}\alpha_k^j - \frac{\sigma_\nu}{\nu-k}x_{max} - \overline{p}^{\nu}_{k} + p_{k-1}$

Hence, for x_{k+1}, one obtains the following using the above relationship:

$x_{k+1} = \frac{(\nu-k-1)(x_k-p_k+\overline{p}^{\nu}_{k+1})-\sigma_\nu x_{max}}{\nu-k}$

As shown in deriving equation (116), this will simplify to:

$$x_{k+1} = x_k + \overline{p}^{\nu}_{k} + \frac{\sigma_\nu x_{max} - x_k}{\nu - k} - p_k \tag{117}$$

Now, equating the relationships obtained in (116) and (117), one gets:

$$\overline{p}^{\nu}_{k} + \frac{\sigma_\nu x_{max} - x_k}{\nu - k} = \overline{p}^{\mu}_{k} + \frac{\sigma_\mu x_{max} - x_k}{\mu - k} \tag{118}$$

for $k = k_1 + 1, \ldots, \mu - 1$. Now, using the above relationship in $x_{k-1} = \beta_k^{j_k+1-\sigma_\nu}$ we get:

$x_{k-1} = \beta_k^{j_k+1-\sigma_\nu} = \frac{\nu-k+1}{\nu-k}x_k + \frac{\sigma_\nu x_{max}}{\nu-k} + p_{k-1} - \overline{p}^{\nu}_{k}$

$\beta_k^{j_k+1-\sigma_\nu} = x_k - (\overline{p}^{\nu}_{k} - \frac{\sigma_\nu x_{max}-x_k}{\nu-k}) + p_{k-1} = x_k - (\overline{p}^{\mu}_{k} - \frac{\sigma_\mu x_{max}-x_k}{\mu-k}) + p_{k-1}$

$$\beta_k^{j_k+1-\sigma_\mu} = \frac{\mu-k+1}{\mu-k}x_k - \frac{\sigma_\mu x_{max}}{\mu-k} + p_{k-1} - \overline{p}^{\mu}_{k} \tag{119}$$

for $k = k_1 + 1, \ldots, \mu - 1$ thereby completing the induction argument and proof for this lemma. If $k1 \neq k_0$ we also have that $\nu = |m_k^{j_k+1-\sigma_\mu}| = |m_k^{j_k+\sigma_\mu}| = \mu$ and thereby the above lemma will satisfy the criteria put forth in the algorithm for computing $\beta_k^{j_k+1-\sigma_\nu}$

2. The optimal input and corresponding cost to go for the dynamical system in equation (57) minimizing the cost function in equation (69) subject to constraints mentioned in 3.1 is given by:

$$u_k^*(x) = \overline{p}_k^\mu - \frac{x - \sigma_\mu x_{max}}{\mu - k} \tag{120}$$

$$J_k^*(x) = (\nu - k)g(\overline{p}_k^\mu - \frac{x - \sigma x_{max}}{\mu - k}) + J_\mu^*(\sigma_\mu x_{max}) \tag{121}$$

Proof: In the previous lemma, it has been proven that there exists a $j_k \in [1, j_k^{max}]$ such that $|m_k^{j_k}| = |m_k^{j_k+1}| = \mu$ or $|m_k^{j_k-\sigma_\mu}| = |n_k^{j_k+1-\sigma_\mu}| = \mu$ for $k \in [k_0, \mu - 1]$, and $\beta_k^{j_k+1-\sigma_\nu}$ can be computed from the algorithm provided that $k1 \neq k_0$.

In this lemma, we will prove the above statement using an induction argument. At $k = \nu - 1$, we know the optimal input using lemma hypothesis is given by $u_k^*(x) = p_k - x_k + \sigma_\mu x_{max}$, and therefore the corresponding optimal cost is $J_k^*(x) = g(p_k - x_k + \sigma_\mu x_{max}) + J_\mu^*(\sigma_\mu x_{max})$ for $x \in [\alpha_k^{j_k}, \alpha_k^{j_k+1}]$ where $j_k = 1 + \sigma_k(j_k^{max} - 2)$ as mentioned in the previous lemma.

Now, suppose that $u_l^*(x) = \overline{p}_l^\mu - \frac{x - \sigma_\mu x_{max}}{\mu - l}$ and $J_l^*(x) = (\nu - l)g(\overline{p}_l^\mu - \frac{x - \sigma x_{max}}{\mu - l}) + J_\mu^*(\sigma_\mu x_{max})$ for $l \in [\mu - 1, k + 1]$, $x \in [\alpha_k^{j_l}, \alpha_k^{j_l+1}]$, $j_l \in [1, j_l^{max}]$. We shall show that the assertion indeed holds true for l=k.

Owing to the convexity of g, we have:

$$J_k^*(x) = \inf_{u \in [p_k - x, u_k^{max}] \cap [u_k^{min}, p_k - x + x_{max}]} (g(u) - (\mu - k - 1)g(\overline{p}_{k+1}^\mu - \frac{x + u - p_k}{\mu - k - 1}))$$
$$= (\mu - k) \inf_{u \in [p_k - x, u_k^{max}] \cap [u_k^{min}, p_k - x + x_{max}]} (\frac{g(u)}{\mu - k} - \frac{\mu - k - 1}{\mu - k} g(\overline{p}_{k+1}^\mu - \frac{x + u - p_k}{\mu - k - 1}))$$

$$J_k^*(x) \geq \ (\mu-k)g(\frac{(\mu-k-1)\overline{p}_{k+1}^{\mu}+p_k-x_k}{\mu-k})$$

$$J_k^*(x) \geq \ (\mu-k)g(\frac{\sum_{i=k+1}^{\mu}p_i+p_k-x}{\mu-k}) = (\mu-k)g(\overline{p}_k^{\mu}-\frac{x}{\mu-k}) \tag{122}$$

provided that $x^+ \in [\alpha_{k+1}^{j_{k+1}}, \alpha_{k+1}^{j_{k+1}+1}]$ and $u \in [u_k^{min}, u_k^{max}]$ where:

$$x_k^+ = x_k + u(x) - p_k = \frac{\mu-k+1}{\mu-k}x_k + \frac{\sigma_\mu x_{max}}{\mu-k} - p_k + \overline{p}_k^{\mu} \tag{123}$$

Therefore, it is seen that the optimal input generating $J_k^*(x)$ is $u_k^*(x) = \overline{p}_k^{\mu} - \frac{x}{\mu-k}$. Furthermore, we can see that:

$u_{k+1}^*(x^+) = \overline{p}_{k+1}^{\nu} - \frac{x^+}{\nu-k-1}$

From the expression for x^+, the above equation simplifies to:

$$u_{k+1}^*(x^+) = \overline{p}_{k+1}^{\nu} - \frac{x^{\pm}\sigma_\nu x_{max}}{\nu-k-1} = u_k^*(x) \tag{124}$$

Thus, one can say that the input $u_k^*(x)$ is governed by a constant expression from $x \in [\alpha_k^{j_k}, \alpha_k^{j_k+1}]$ to $x_\nu = \sigma_{nu}x_{max}$. Therefore, if $\sigma_{nu} = 0$ then $u_k^*(x) = \overline{p}_k^{\mu} - \frac{x}{\mu-k}$, which is the condition to be satisfied for any lower bound trajectory. On the other hand $\sigma_{nu} = 1$ gives $u_k^*(x) = \overline{p}_k^{\mu} - \frac{x-x_{max}}{\mu-k}$, which is the condition to be satisfied for any upper bound trajectory. To show that $x^+ \in [\alpha_{k+1}^{j_{k+1}}, \alpha_{k+1}^{j_{k+1}+1}]$, we use the results obtained from our previous lemma, which states that:

$$\beta_k^{j_k} = \frac{\mu-k+1}{\mu-k}\alpha_k^{j_k} - \frac{\sigma_\mu x_{max}}{\mu-k} + p_{k-1} - \overline{p}_k^{\mu} \tag{125}$$

Realize for the upper bound that $\alpha_k^{j_k+1} = min(\beta_{k+1}^{j_{k+1}+1}, x_{max})$. So, according to the previous lemma, we can represent $\beta_{k+1}^{j_{k+1}+1}$ as:

$\alpha_{k+1}^{j_{k+1}+1} = \frac{\mu-k-1}{\mu-k}\beta_{k+1}^{j_{k+1}+1} + \frac{\sigma_\mu x_{max}}{\mu-k} - p_k + \overline{p}_k^{\mu}$

$$\alpha_{k+1}^{j_{k+1}+1} \geq \ \frac{\mu-k-1}{\mu-k}\alpha_k^{j_k+1} + \frac{\sigma_\mu x_{max}}{\mu-k} - p_k + \overline{p}_k^{\mu} \tag{126}$$

Now, the difference $\alpha_{k+1}^{j_{k+1}+1} - x^+$ can be represented as:

$$\alpha_{k+1}^{j_{k+1}+1} - x^+ \geq \ \frac{\mu-k-1}{\mu-k}(\alpha_k^{j_k+1} - x) \geq \ 0 \tag{127}$$

Similarly, for the lower bound, we know that $\alpha_k^{j_k} = min(0, \beta_{k+1}^{j_{k+1}})$, therefore:

$\alpha_{k+1}^{j_{k+1}} = \frac{\mu-k-1}{\mu-k}\beta_{k+1}^{j_{k+1}} + \frac{\sigma_\mu x_{max}}{\mu-k} - p_k + \overline{p}_k^\mu$

$$\alpha_{k+1}^{j_{k+1}+1} \leq \frac{\mu-k-1}{\mu-k}\alpha_k^{j_k} + \frac{\sigma_\mu x_{max}}{\mu-k} - p_k + \overline{p}_k^\mu \tag{128}$$

This implies that:

$$x^{\pm}\alpha_{k+1}^{j_{k+1}+1} \geq \frac{\mu-k-1}{\mu-k}(x - \alpha_k^{j_k}) \geq 0 \tag{129}$$

Thus confirming that $x^+ \in [\alpha_{k+1}^{j_{k+1}}, \alpha_{k+1}^{j_{k+1}+1}]$. We also need to ensure that $u_k^*(x) \in [p_{min}, p_{max}]$ & $u_k^{min} \leq u_k^*(x) \leq u_k^{max}$. To prove this, lets consider x_l for $k \leq l \leq \mu$ to be generated by the optimal control sequence $u_l^*(x)$. Therefore:

$$x_{l+1} = x_l + u_l^*(x) - p_l \tag{130}$$

As mentioned earlier, this control sequence has a constant expression, therefore $u_k^*(x) = u_{\mu-1}^{(}x_{\mu-1}) = p_{\mu-1} - x_{\mu-1} + \sigma_{nu}x_{max}$. Lets now consider the case where $\sigma_\nu = 0$, which is conducive with a lower bound trajectory, which means that $u_k^*(x) \leq p_{\mu-1} < 1$. Then we have:

$$u_k^*(x) = u_{\mu-1}^{(}x_{\mu-1}) = p_{\mu-1} - x_{\mu-1} \tag{131}$$

From the algorithm with $\alpha_\mu^2 = 0$, we have that

$$\beta_\mu^2 = p_{\mu-1} - \frac{\sigma_\nu}{\nu-\mu}x_{max} - \overline{p}_\mu^\nu \tag{132}$$

where $\nu = |m_\nu^1|$. For the lower bound, one realizes from our algorithm that $\alpha_{\mu-1}^2 = max(0, \beta_\mu^2)$. Therefore, one can claim that:

$p_{\mu-1} - x_{\mu-1} \geq p_{\mu-1} - \alpha_{\mu-1}^2 = p_{\mu-1} - \beta_\mu^2$

On simplifying further, this becomes

$$p_{\mu-1} - \beta_\mu^2 = \frac{\sigma_\nu}{\nu-\mu}x_{max} + \overline{p}_\mu^\nu \geq \overline{p}_\mu^\nu \tag{133}$$

Therefore, $u_k^*(x) \geq \ u_k^{min}$. Similarly $\sigma_\mu = x_{max}$ for the upper bound and so $u_k^(x) \geq \ p_{\mu-1} > u_k^{min}$. Also, from the algorithm with $\alpha_\mu^1 = x_{max}$, we have that

$$\beta_\mu^1 = \frac{\nu - \mu - 1}{\nu - \mu} x_{max} + p_{\mu-1} - \frac{\sigma_\nu}{\nu - \mu} x_{max} - \overline{p}_\mu^\nu \tag{134}$$

For the upper bound, one realizes from our algorithm that $\alpha_{\mu-1}^1 = min(\beta_\mu^1, x_{max})$. Therefore, one can claim that:

$p_{\mu-1} - x_{\mu-1} \geq \ p_{\mu-1} - \alpha_{\mu-1}^1 + x_{max} = p_{\mu-1} - \beta_\mu^1 + x_{max}$

On simplifying further, this becomes

$$p_{\mu-1} - \beta_\mu^1 = \frac{\sigma_\nu - 1}{\nu - \mu} x_{max} + \overline{p}_\mu^\nu \geq \ \overline{p}_\mu^\nu \tag{135}$$

thereby proving that $u_k^*(x) \in [u_k^{min}, u_k^{max}]$ minimizes $h(x,u) := g(u) + J_{k+1}^*(x + u - p_k)$ subject to $x^+ = x + u - p_k \in [\alpha_{k+1}^{j_{k+1}}, \alpha_{k+1}^{j_{k+1}+1}]$. Furthermore, $u_k^*(x)$ is the local minimum for unconstrained h(x,u) for a fixed $x \in [\alpha_k^{j_k}, \alpha_k^{j_k+1}]$. However, the convexity of g and $J_{k+1}^*(x)$ as described in lemma 2 imply that h(x,u) is a convex function. Therefore $u_k^*(x)$ is also the global minimum for h(x,u). Thus:

$$h(x, u_k^*(x)) = (\nu - k) g(\overline{p}_k^\mu - \frac{x}{\mu - k}) + J_\mu^*(\sigma_\mu x_{max}) \tag{136}$$

coincides with the cost to go function $J_k^*(x)$ on $[\alpha_k^{j_k}, \alpha_k^{j_k+1}]$ thereby completing the proof for our lemma.

3.5.3 Resulting Theorem

This Theorem results from the algorithm and lemma above. It asserts that the optimal solution obtained does indeed minimize the modified cost function globally, when subject to the dynamics of the system as defined by the state equation.

Theorem 1: Suppose L and g are convex on $[u^*, u^{max}]$, and the intervals $[\alpha_k^{j+k}, \alpha_k^{j_k+1}]$ partition the region $[0, x_{max}]$, then the optimal control law for $u_k^*(x)$ as introduced in the algorithm minimizes the modified cost function in equation (69) when subject to

the dynamics of the system as defined in equation (57). Moreover the Optimal Cost to go for $x \in [\alpha_k^{j+k}, \alpha_k^{j_k+1}]$ is described by:

$$J_k^*(x) = \begin{cases} 0 & \nu = \infty \\ (\nu - k)g(\overline{p}_k^{\nu} - \frac{\sigma_\nu x_{max}}{\nu - k}) + J_\nu^*(\sigma_\nu x_{max}) & \nu < \infty \end{cases} \tag{137}$$

Proof:

Let us first prove the theorem for $\nu = \infty$. This case only occurs before any of the interval bounds exceed the upper limit, such that $m_k^{j_k} = \infty$ and $\alpha_k^{j_k+1} = x_{max}$. The claim obviously holds true at $k = N$. Using an inductive argument, suppose our argument holds true for $N \leq \ l \leq \ k+1$, we must show that the above also holds true at k.

To this end, let $\alpha_k^j \leq \ x \leq \ x_{max}$. Suppose that $x_{max} - x + p_k < u_0$, we know that from the algorithm $u_k^*(x) = x_{max} - x + p_k$. Noting that $x^+ = x + u_k^*(x) - p_k$, and using the previous result we get $x^+ = x_{max}$. Now, suppose $x_{max} - x + p_k \geq \ u_0$, then from our algorithm $u_k^*(x) = u_0$ and therefore $x^+ = x + u_0 - p_k$. Since x_k^+ is bounded by x_{max}, the criteria $x + u_0 - p_k \leq \ x_{max}$ must be met.

Now, consider the switching criteria $x \geq \ \alpha_k^j + u^*$ and $x < \alpha_k^j + u^*$. In the first case, $u_0 = 0$, and therefore $x^+ = x - p_k \geq \ \alpha_k^j + u^* - p_k$ must be satisfied. In the other case, $u_0 = u^*$, and so $x^+ = x + u^* - p_k \geq \ \alpha_k^j + u^* - p_k$ must be satisfied. By induction hypothesis, note that $J_{k+1}^*(x^+) = 0$. Therefore, $J_k^*(x) = g(u_k^*) + J_{k+1}^*(x^+)$, and since $g(u) = 0 \ \forall u \leq \ u^*$, we can say that $J_k^*(x) = 0$

To complete the induction proof for $\nu < \infty$, suppose the results hold for $\nu = \nu_1, \ldots, \nu_i$. We need to show that the same argument holds true for $\mu = \nu - 1$. Recall that:

$$J_k^*(x) = \inf_{u \in [p_k - x, u_k^{max}] \cap [u_k^{min}, p_k - x + x_{max}]} (g(u) + J_\mu^*(x + u - p_k)) \tag{138}$$

Now, letting $k = \mu - 1$, and $x \in [\alpha_k^1, \alpha_k^2]$, we have

$$J_\mu^*(x) = (\nu - mu)g(\overline{p}_k^{\nu} - \frac{\sigma_\nu x_{max}}{\nu - \mu}) + J_\nu^*(\sigma_\nu x_{max}) \tag{139}$$

However, realize that the input $u_k^* = \overline{p}_k^{\nu} - \frac{\sigma_\nu x_{max}}{\nu - k}$ that minimizes $g(u) + J_\mu^*(x + u - p_k)$ is infeasible, since the bounds for x^+ are not met for $\sigma_\nu = 0$ or $\sigma_\nu = 1$.

$$x^+ = x + u_k^*(x) - p_k = \frac{\mu - k + 1}{\mu - k} x - \frac{\sigma_\nu}{\mu - k} x_{max} + \overline{p}_k^{\mu} - p_k \tag{140}$$

For $\sigma_\nu = 0$, we have that

$x^+ = \frac{\mu - k + 1}{\mu - k} x + \overline{p}_k^{\mu} - p_k$

Noting that $x \leq \alpha_k^2$ at $k = \mu - 1$, and along with the fact that $\alpha_\mu^2 = 0$, we have:

$$x^+ \leq \frac{\mu - k + 1}{\mu - k} \alpha_k^2 + \overline{p}_k^{\mu} - p_k = \frac{\mu - k + 1}{\mu - k} \alpha_\mu^2 + \overline{p}_k^{\mu} - p_k = 0 \tag{141}$$

Similarly, for $\sigma_\nu = 1$, we have that

$x^+ = x + u_k^*(x) - p_k = \frac{\mu - k + 1}{\mu - k} x - \frac{x_{max}}{\mu - k} + \overline{p}_k^{\mu} - p_k$

Noting that $x \geq \alpha_k^1$ at $k = \mu - 1$, and along with the fact that $\alpha_\mu^1 = x_{max}$, we have:

$$x^+ \geq \frac{\mu - k + 1}{\mu - k} \alpha_k^1 - \frac{x_{max}}{\mu - k} + \overline{p}_k^{\mu} - p_k = \frac{\mu - k + 1}{\mu - k} \alpha_\mu^1 + \overline{p}_k^{\mu} - p_k - \frac{x_{max}}{\mu - k} = x_{max} \tag{142}$$

Also, realize that the function $g(u) + J_\mu^*(x + u - p_k)$ is a convex function, since both $g(u)$ and $J_\mu^*(x) = f_{N-\mu}(q_{N-\mu} - x)$ are both convex from the lemma proven in 3.4. Consequently, $u_k^*(x) = p_k - x + \sigma_\nu x_{max}$ is the optimal solution resulting in $x^+ = \sigma_\nu x_{max}$ and minimizes the cost to go function $J_k^*(x) = g(p_k - x + \sigma_\nu x_{max}) + J_\mu^*(\sigma_\nu x_{max})$. Therefore, the hypothesis presented in the second part of lemma 2 for $u_k^*(x)$ is satisfied, and induction argument is complete thereby completing the proof.

3.6 Transitioning to a time varying cost

The interval back propagation routine we applied assumes a constant $g_k = g$. Therefore, it is necessary to show that the difference between optimal and resulting suboptimal solutions are sufficiently small, such that the solution derived from considering time invariant cost can be extended to the time varying case. The following theorem and proof will elucidate how this criteria is met.

Theorem 2: Let each incremental cost function $g_k : [u_k^{min},\ u_k^{max}] \rightarrow \mathbb{R}$ be convex. Now, suppose that $\sup_{u \in \mathcal{U}}(g_k(u) - g_{k'}(u)) \leq\ \epsilon\ \forall [k, k'] \in [0, N-1]$ *and* $\mathcal{U} \in [u_k^{min},\ u_k^{max}]$, then the resulting suboptimal cost-to-go for any given state of charge x is $J_k(x) = \sum_{l=k}^{N-1} g_l(\overline{u^*})$ is within ϵ of $J_k^*(x)$, where $\overline{u^*}$ and $J_k^*(x)$ are as specified in the algorithm.

Proof:

Let $g : \mathcal{U} \rightarrow \mathbb{R}^+$ be given by $g = \max_{0<k\leq\ N-1} g_k(u)$. It is seen that g is a convex function on $\mathcal{U}$. Now, let $u_k^*(x)$ be the optimal input that minimizes the control sequence $\sum_{l=1}^{N-1} g_l(u_l)$. Defining $\tilde{g_k}(u) = g(u) - g_k(u)\ \overline{u_k^*} \in [p_{min}, p_{max}]$ for $u_k^{min} \leq \overline{u_k^*} \leq\ u_k^{max}$ we have:

$$J_k(x) = \sum_{l=1}^{N-1} g_l(\overline{u_l^*}) \leq\ \sum_{l=1}^{N-1} g(\overline{u_l^*}) \leq\ \sum_{l=1}^{N-1} g(\overline{u_l^*}) \tag{143}$$

Due to the fact that $g_k(u) \leq\ \ g(u)$, and $u_k^*(x)$ minimizes the control sequence $\sum_{l=1}^{N-1} g_l(u_l)$. Using $\tilde{g_k}(u_k^*) = g(u_k^*) - g_k(u_k^*)$ we have that:

$$J_k(x) \leq\ \sum_{l=1}^{N-1} g_l(u_l^*) + \sum_{l=1}^{N-1} \tilde{g_k}(u_l^*) \leq\ \ J_k^*(x) + \sum_{l=1}^{N-1} \tilde{g_k}(u_l^*) \tag{144}$$

The proof now follows the form $\tilde{g_k}(u) = g(u) - g_k(u) \leq\ \epsilon\ \forall u \in \mathcal{U}\ \forall k \in [0, N-1]$

Realize that even though it is possible to obtain a closed form solution assuming a time varying $g_k(u)$, the above proof shows that the largest variation that occurs in $g_k(u)$ is bounded by ϵ. Therefore, in cases where ϵ is sufficiently small, one can use the back propagation results derived in section 3.5 assuming a constant g to obtain the optimal solution with less computational effort.

CHAPTER IV

REAL TIME CONTROL STRATEGY

4.1 Control Implementation

Net Power Demand is assumed as the primary input parameter. By Theorem 2 we may use the g_k–independent controller with Intervals, Cost to go, and optimal input functions are calculated using the Dynamic Programming algorithm discussed earlier without any significant increase in fuel economy as long as the variations in g_k induced by the vehicle speed and power demand are sufficiently small.

Knowing the power demand, input constraints, and state of charge constraints ahead of time, the Interval back propagation algorithm can be easily implemented on line. Using the interval data, one calculates the corresponding optimal inputs as described in the algorithm. The entire process of calculating the intervals and obtaining the optimal input for 800 steps takes around 15 microseconds on a 1.4 GHz Intel Core i5 Processor running MATLAB, which confirms that the algorithm is computationally effective for real time implementation. Recall that Interval bound calculation involves performing elementary operations on the power demand, state of charge, and known interval bounds from the previous step while satisfying a set of conditionals to ensure that the system constraints are met. Meanwhile, Optimal input is an elementary function of the power demand, state of charge, and time step which makes decisions based upon the interval data provided. Thus, neither operation is computationally intensive for the CPU to perform.

Recall that from section 2.5, the engine power can be expressed as a function $P_e = \psi(\omega_m, \dot{\omega}_m, P_d, U)$. Consequently, one can also calculate the engine speed. However, it is quite cumbersome to obtain an exact solution to w_e in the functional form described.

This is due to the circular dependence between w_e & T_e. Therefore, we have to make an approximation to obtain a viable solution for the optimal engine speed. One method of approximation involves using a look up table. Another method involves using an approximate equation to model the engine speed and engine power in terms of engine torque. We decided to proceed with the latter approach. From [46], it has been observed that the Optimal Engine torque Vs. Engine speed curve can be approximated by using a linear model. Therefore, we can model $\omega_e = h^{-1}(T_e)$, as:

$$\omega_e = h^{-1}(T_e) = \omega_0 + \omega_1 T_e \tag{145}$$

where ω_0 and ω_1 are constant coefficients. Consequently, the engine power is:

$$P_e = T_e w_e = \omega_0 T_e + \omega_1 T_e^2 \tag{146}$$

Therefore, the equation for input in terms of engine power from equation (44):

$$aT_e^2 + bT_e + c = 0 \tag{147}$$

where the coefficients a,b, and c are determined based on the efficiency functions, motor speed, and power demand. as shown below.

$$a = \omega_1 - k_e \frac{s^2 + r^2}{(r+s)^2} \tag{148}$$

$$b = \omega_0 - 2k_e \frac{r}{r+s}(I_g \dot{\omega_m} - T_d) \tag{149}$$

$$c = \frac{r}{s} I_g \omega_m \dot{\omega}_m - P_d + \tilde{P}_d - U - \epsilon_b (U - \tilde{P}_d)^2 - k_e (T_d^2 + \frac{r^2}{s^2} I_g^2 \dot{\omega_m}^2) \tag{150}$$

Solving for T_e from the above equation, we have:

$$T_e = \frac{-b \pm (b^2 - 4ac)^{1/2}}{2a} \tag{151}$$

In the above equation, we select the positive root for T_e since engine speed is always non negative. Consequently, one can calculate the corresponding engine torque based off our earlier assumption.

The optimal engine speed demand (in rpm), and optimal engine torque is used as a set point for the engine speed controller and generator controller which jointly control the throttle levels. A complete schematic of our controller is seen in figure 11. The MATLAB Function block takes in the current state of charge, Power Demanded, and time step to generate the interval data and subsequently compute the optimal input. This optimal input is sent to the MATLAB Function1 block. Here our optimal engine torque is calculated in terms of the motor speed, motor acceleration, power demand, and optimal input as shown in equations (145-151). Correspondingly, one obtains the optimal engine speed using equation (145). This information is sent to the engine speed, motor speed, and generator controllers, which are linked to the engine and electrical system. We will now discuss the design of our engine speed controller, generator controller, and motor speed controller in further detail.

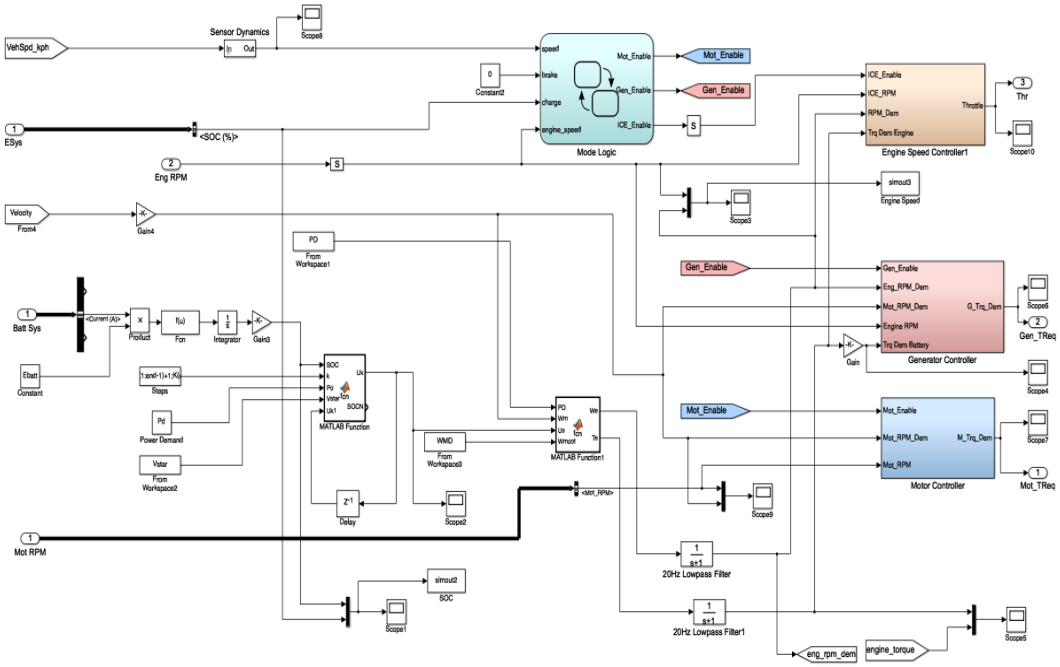


Figure 11: HEV Control Design using Simulink

4.1.1 Engine speed Controller

The Engine controller takes in the current Engine rpm, along with the optimal engine speed & torque, which depends on the optimal input calculated by our algorithm. Using this information, it calculates the required throttle levels. First off, it verifies whether the engine speed demanded is greater than the minimum engine speed required for turning on the engine. This physical constraint must be satisfied to make sure the engine does not operate under unfavorable conditions. Based on this, the controller decides whether the engine should be turned on or off.

A PI controller is then used to control the engine speed or engine torque based off information from the generator controller, which determines whether the engine will operate in speed control or torque control mode. Note that this is a consequence of the coupled system effect explained in section 2.5. For controlling the engine torque feedback loop, our optimal torque is normalized by the maximum torque for a given engine speed. Using engine speed or engine torque, one determines the desired throttle angle which is then communicated to the IC engine. A schematic of our engine speed controller is shown in figure 12

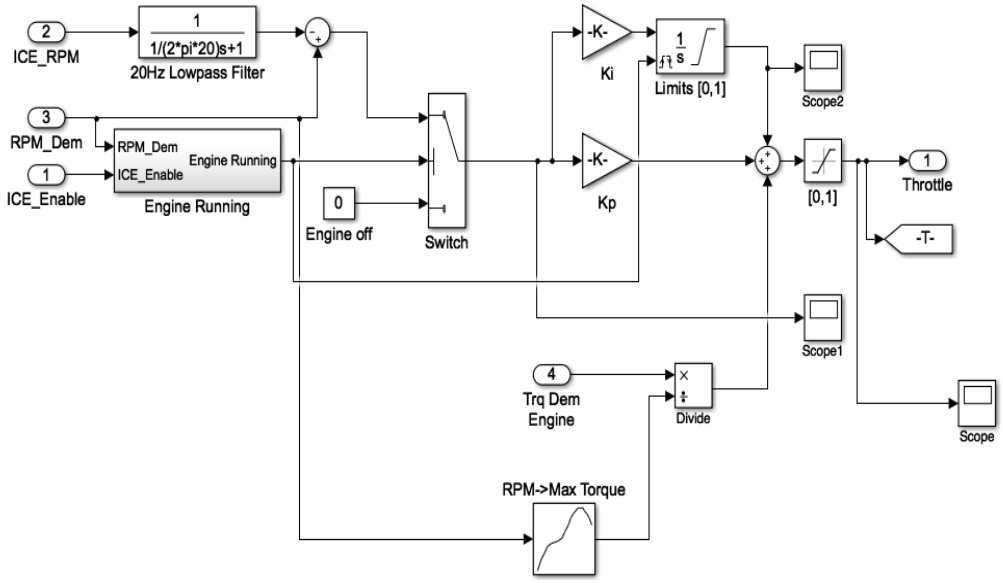


Figure 12: Engine Speed PI controller

4.1.2 Generator Controller

The generator Controller requires current engine rpm, engine speed demanded, and generator torque demanded as inputs. It then calculates the required generator torque for operating the engine. The generator controller follows a Proportional control architecture. As mentioned earlier, the engine speed demanded is checked against the physical constraints.

The torque Vs. Speed modes are based on the engine's operating condition. Under low rpm condition's it is used as a reference for the speed controller. This speed controller is used to calculate a reference value for the generator torque and does so based off the decision between speed versus torque control modes. It operates on a PI controller architecture. At higher rpm, engine rpm demand is used to calculate the reference generator torque using proportional control. This reference value is then compared with the generator torque demand. Subsequently, this generator torque demand is communicated to the electrical system. A schematic of our generator controller is shown in figure 13

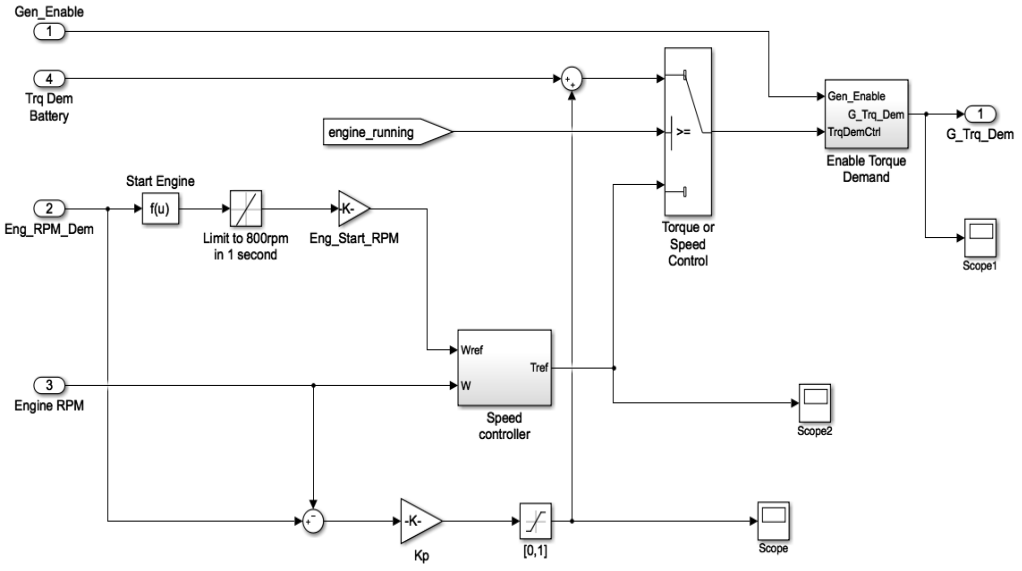


Figure 13: Generator Controller

4.1.3 Motor Speed Controller

In a power split system, the motor is distinct from the engine and the generator, as it is decoupled from both and linked directly to the external environment. This is apparent from the dynamical equations presented in section 2.5. The motor speed controller takes in the current motor rpm and motor speed demanded. The motor rpm demand is determined using linear interpolation based off the vehicle acceleration, since it is directly linked to the wheels. It then uses a PI speed control architecture to ensure that the desired rpm is met, and calculates the corresponding motor torque. This is then communicated to the electrical system. A schematic of our motor controller is shown in figure 14

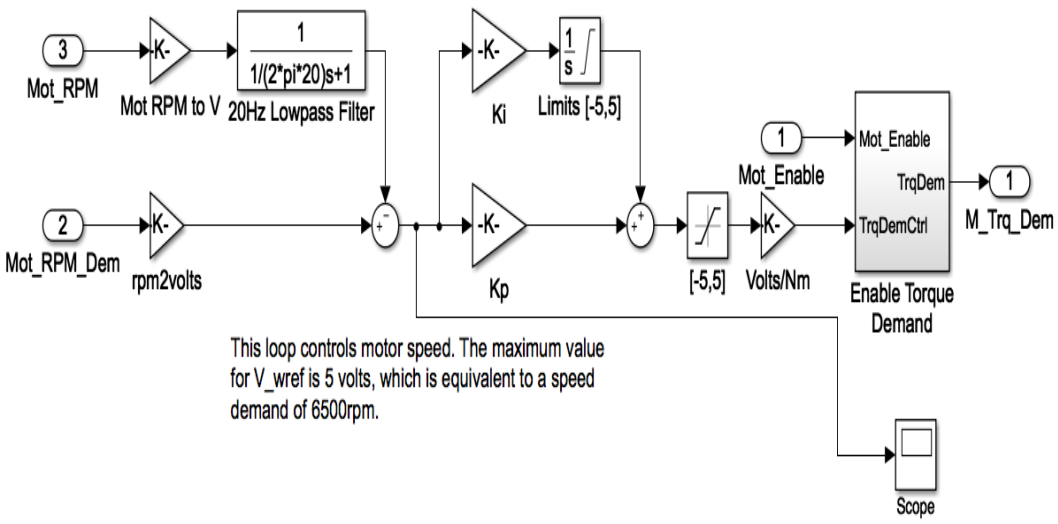


Figure 14: Motor PI Controller

4.2 Algorithm Testing and Results

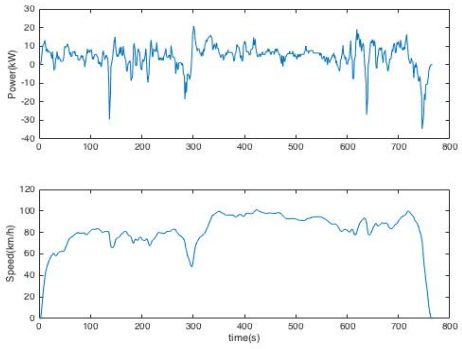
4.2.1 Drive Cycles and Algorithm Results

In order to test the DP algorithm proposed in this thesis the simulations are carried out on six different custom drive cycles namely Highway, City to Suburb, High Speed

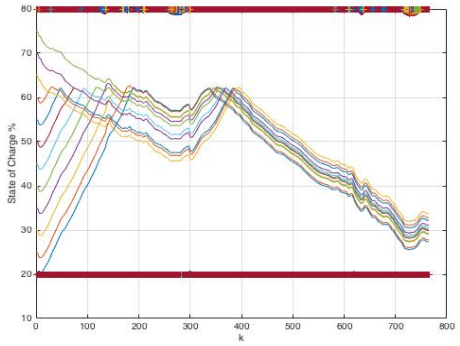
Test, City, Urban Dynamometer Driving Cycle (UDDS), and Japanese1015. We assume that there is no road slope across all cycles. A detailed description of each drive cycle and the corresponding results obtained on implementing the algorithm are shown below:

Highway This cycle was developed by the EPA to represent the operation of vehicles under typical freeway conditions in the United States. The length of this cycle is 765 seconds, average speed for this cycle is 52.06 mph and total distance travelled is 11.06 miles. The power demand and speed profiles, intervals along with optimal trajectories, incremental and cost to go functions, and optimal inputs are seen in figures 15 a to 15 d respectively. Notice that there are very few intervals due to low power demand. Consequently, the optimal input is greater than u^* for very few cases. From the optimal input profile, we expect the engine to turn on twice across the length of this cycle, which happens initially and at 300 seconds. In addition, notice that the optimal trajectories with different initial conditions get closer to each other over time.

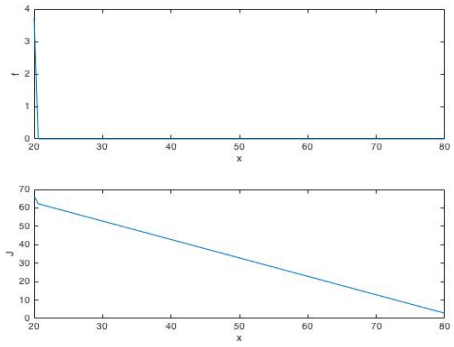
City to Suburb This drive cycle is a section based off the typical city to suburb commute pattern in an American city. It involves a mixture of arterial road and freeway driving, with highly variable speeds due to erratic variations in traffic conditions. The length of this cycle is 1000 seconds, average speed for this cycle is 41.75 mph and total distance travelled is 11.6 miles. The power demand and speed profiles, intervals along with optimal trajectories, incremental and cost to go functions, and optimal inputs are seen in figures 16 a to 16 d respectively. Notice that there are very few intervals due to low power demand. Consequently, the optimal input is greater than u^* for very few cases. From the optimal input profile, we expect the engine to turn on 8 times across the length of this cycle. The switch happens initially and periodically between 200 to 600 seconds where the power demand is increasing. In addition, notice that the optimal trajectories with different initial conditions converge beyond



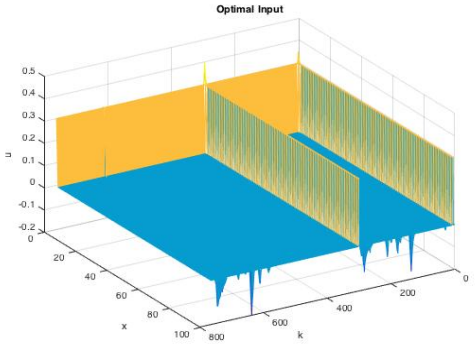
(a) Power Demand and Speed Profile



(b) Intervals and Optimal Trajectories

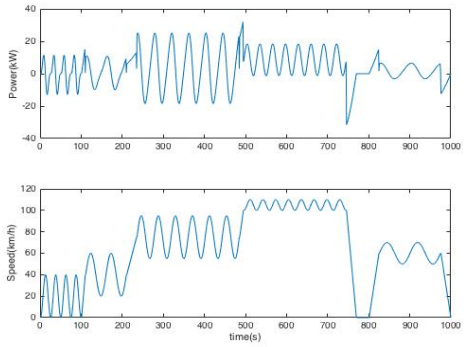


(c) Incremental and Cost to go Functions

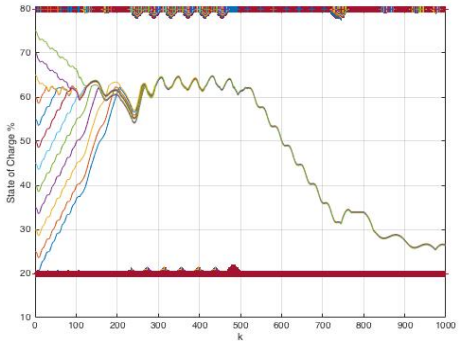


(d) Optimal Inputs

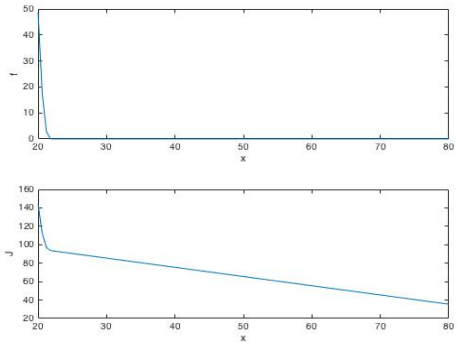
Figure 15: Highway Drive Cycle



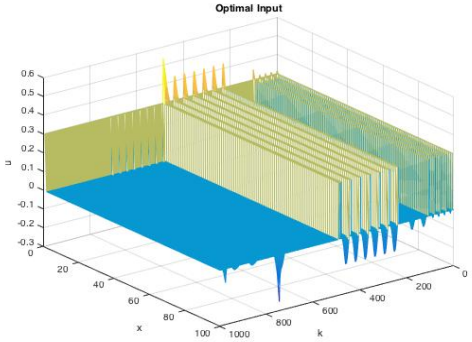
(a) Power Demand and Speed Profile



(b) Intervals and Optimal Trajectories



(c) Incremental and Cost to go Functions

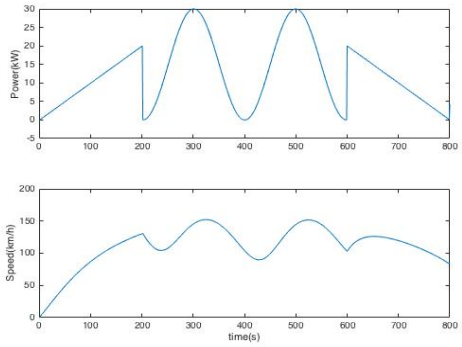


(d) Optimal Inputs

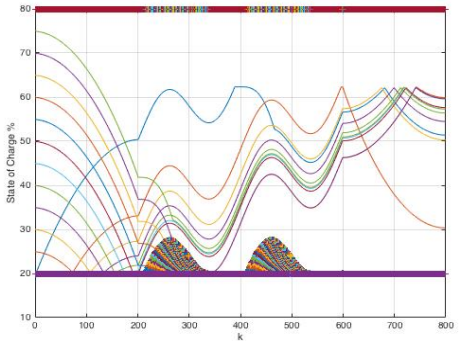
Figure 16: City to Suburb Cycle

300 seconds and the final state of charge is around 27 % regardless of the initial SOC.

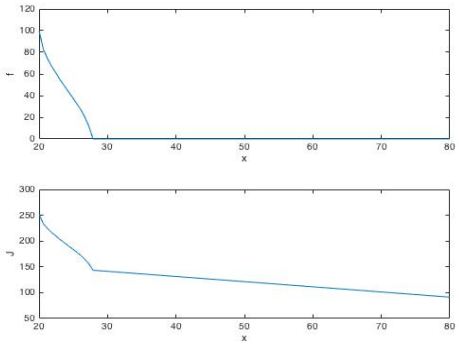
High Speed Test Cycle: This drive cycle is used to test the high speed performance of the vehicle. The length, average speed, and distance traveled during this cycle are 800 seconds, 70.92 mph, and 15.76 miles respectively. The power demand and speed profiles, intervals along with optimal trajectories, incremental and cost to go functions, and optimal inputs are seen in figures 17 a to 17 d respectively. Notice that there are a significant number of intervals due to high power demand. Consequently, the optimal input is greater than u^* at several instances. Notice that the engine is only switched on once, but the time for this switch depends on our initial state of charge. Also, for states of charge greater than 55 % note that the optimal input is greater than u^* to turn on the engine. This allows us to take advantage of the



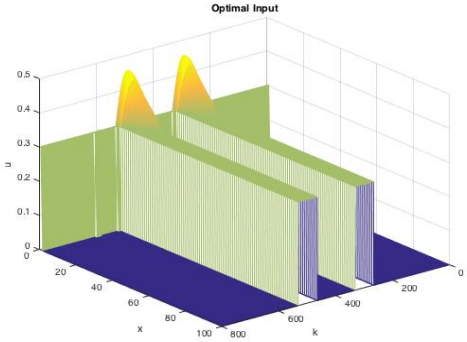
(a) Power Demand and Speed Profile



(b) Intervals and Optimal Trajectories



(c) Incremental and Cost to go Functions

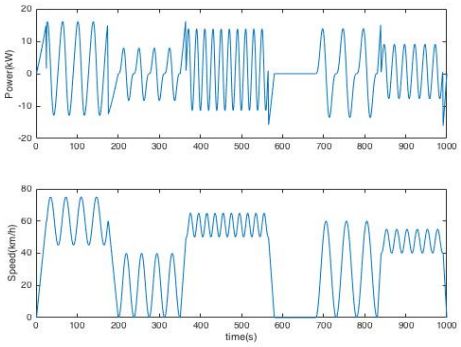


(d) Optimal Inputs

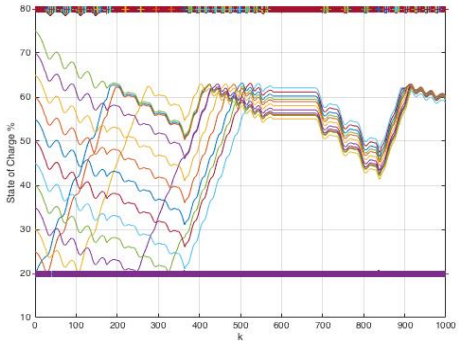
Figure 17: High speed Testing Drive Cycle

intervals for minimizing the control effort, and subsequently the fuel consumption.

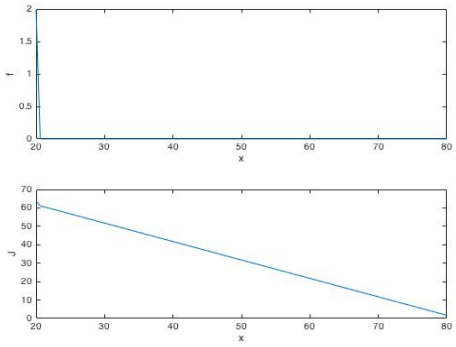
City: This drive cycle is representative of a large American metropolis with high pedestrian traffic and surface level public transit. This cycle involves a mixture of narrow streets, crowded arterial roads, and urban freeways. Variations in speed are highly erratic with frequent stop and go situations. The length of this cycle is 1000 seconds, average speed for this cycle is 24.22 mph and total distance travelled is 6.73 miles. The power demand and speed profiles, intervals along with optimal trajectories, incremental and cost to go functions, and optimal inputs are seen in figures 18 a to 18 d. respectively. Notice that there are very few intervals due to low power demand. Consequently, the optimal input is greater than u^* for very few cases. From the optimal input profile, we expect the engine to turn on twice across the length of this



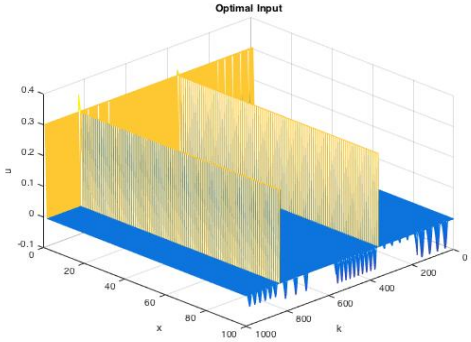
(a) Power Demand and Speed Profile



(b) Intervals and Optimal Trajectories



(c) Incremental and Cost to go Functions

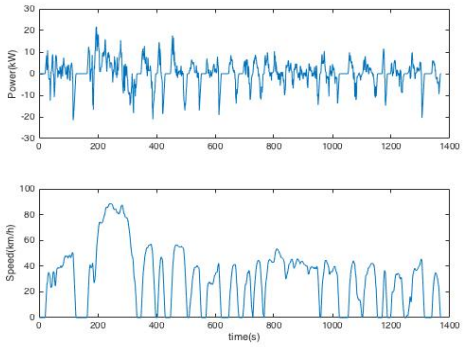


(d) Optimal Inputs

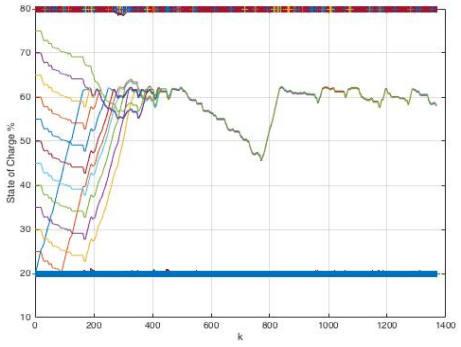
Figure 18: City Drive Cycle

cycle. The instant for our first switch depends on the initial state of charge. However, it is seen that the second switch occurs at 750 seconds for all optimal trajectories. In addition, notice that the optimal trajectories with different initial conditions start to converge towards the end.

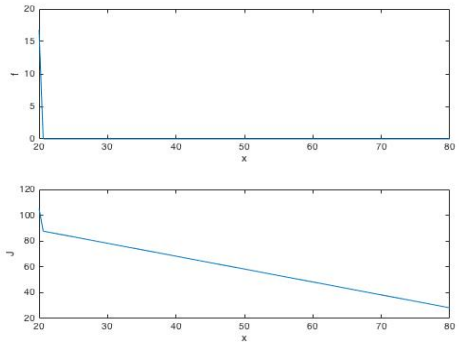
UDDS: This drive cycle was developed by the EPA and simulates erratic variations in speed, high acceleration rates, and frequent stop & go situations synonymous with city driving. The length of this cycle is 1370 seconds, average speed for this cycle is 19.56 mph and total distance travelled is 7.44 miles. The power demand and speed profiles, intervals along with optimal trajectories, incremental and cost to go functions, and optimal inputs are seen in figures 19 a to 19 d respectively. Notice that there are very few intervals due to low power demand. Consequently, the optimal



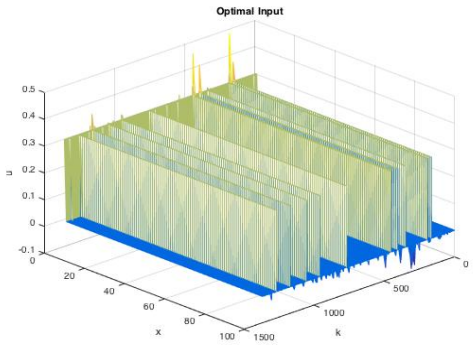
(a) Power Demand and Speed Profile



(b) Intervals and Optimal Trajectories



(c) Incremental and Cost to go Functions

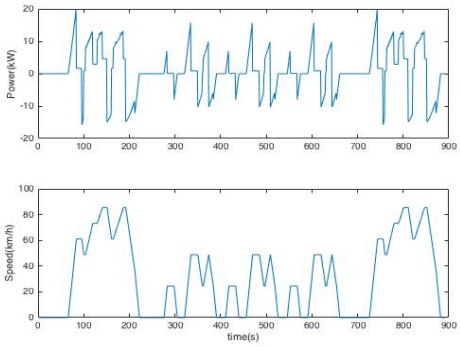


(d) Optimal input

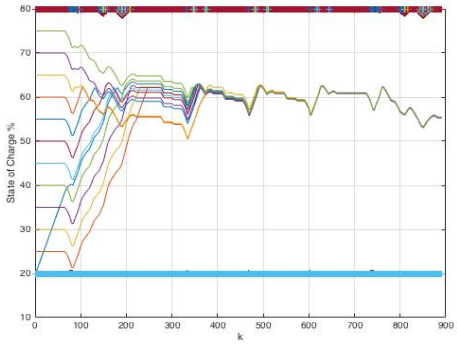
Figure 19: UDDS Drive Cycle

input is greater than u^* for very few cases. From the optimal input profile, we expect the engine to turn on 9 times across the length of this cycle. The switch happens at 190, 350, 420, 470, 780, 960, 1050, 1150, and 1260 seconds respectively. In addition, notice that the optimal trajectories with different initial conditions converge beyond 400 seconds and the final state of charge is around 60 % regardless of the initial SOC.

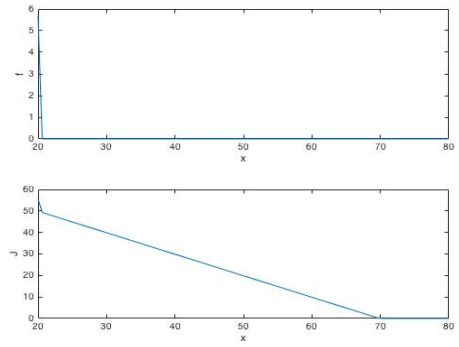
Japanese1015: This drive cycle was developed to test the fuel economy provided by cars designed in Japan. It simulates moderate variations in speed and acceleration rates with frequent stop & go situations synonymous with suburban driving conditions. The length of this cycle is 892 seconds, average speed for this cycle is 32.02 mph and total distance travelled is 7.935 miles. The power demand and speed profiles, intervals, incremental and cost to go functions, and optimal inputs are seen in



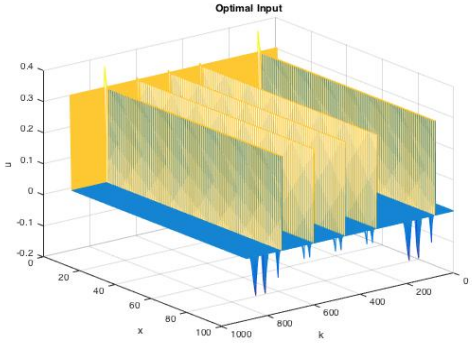
(a) Power Demand and Speed Profile



(b) Intervals and Optimal Trajectories



(c) Incremental and Cost to go Functions



(d) Optimal Input

Figure 20: Japanese 1015 Drive Cycle

figures 20 a to 20 d respectively. Notice that there are very few intervals due to low power demand. Consequently, the optimal input is greater than u^* for very few cases. From the optimal input profile, we expect the engine to turn on 5 times across the length of this cycle. The switch happens at 80, 340, 470, 600, and 740 seconds respectively. In addition, notice that the optimal trajectories with different initial conditions converge beyond 450 seconds and the final state of charge is around 55 % regardless of the initial SOC.

4.2.2 Fuel Economy

One can compute the fuel consumption rate (in gal/s) knowing optimal engine speed and corresponding minimized BSFC. The fuel economy (in mpg) is then obtained as:

$$MPG^* = 0.788V(t)/FC(\omega_e(U,t)) \tag{152}$$

Table 4: Fuel Economy Comparison

Drive Cycles	Rule Based Control mpg	Interval Back propagation mpg	Percent Change
Highway	60.35	102.7	70.17
City to Suburb	52.54	73.34	39.59
High Speed Test	50.27	57.65	14.7
City	48.94	68.6	40.17
UDDS	45.86	57.08	24.47
Japanese1015	49.72	62.8	26.31

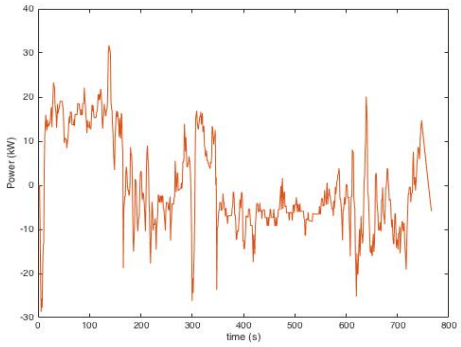
Since we want fuel economy over an entire cycle, its average value over the time window T_w is computed using:

$$\overline{MPG}^* = \frac{0.788}{T_w} \int_0^{T_w} \frac{V(t)}{FC(\omega_e(U,t))} dt \tag{153}$$

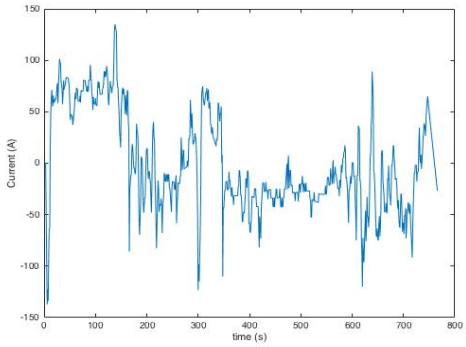
The Fuel Economy Performance for these cycles was tested using mode logic control and Interval back propagation based dynamic programming algorithm. Initial state of charge is 30% for highway and City to Suburb drive cycles, 55% for Japanese1015 cycle, and 60% for UDDS, City, and High Speed Test cycles. From the Table below, its seen that using the Interval Back propagation algorithm helps achieve significant improvements in fuel economy over all cycles. The average fuel economy is 70.36 mpg over all cycles, which is a 35.9% improvement over the fuel economy obtained using rule based mode logic control. The fuel economy results are presented in table 4.

4.2.3 Effect of Maximum Battery Discharge Rate

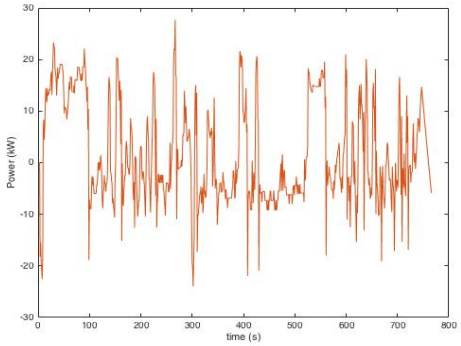
It is important to realize that we are using a battery with a significantly large discharge rate. We have imposed a discharge power limit of 25 kW, as this remains consistent with the power limits mentioned in [46]. Subsequently, fuel economy for the highway drive cycle is unusually high, since power demand is low outside the two acceleration periods since velocity is fairly constant. Therefore, our battery is capable of running the HEV in electric mode, while meeting the power and current



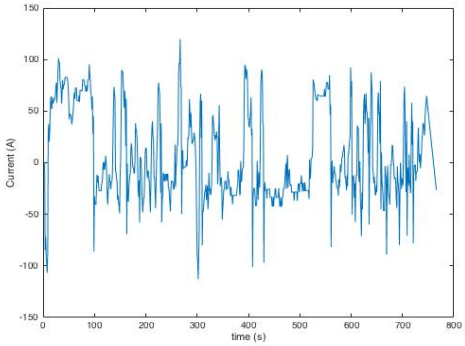
(a) Battery Power 25 kW discharge



(b) Battery Current 25 kW discharge



(c) Battery Power 10 kW discharge



(d) Battery Current 10 kW discharge

Figure 21: Battery power and Current for Highway Drive Cycle

limits as shown in Figure 21 a) and b) under steady state conditions (when the engine isn't switching between on and off states). The battery power momentarily spikes at instances where the engine switches between off to on, because it has to reach 2450 rpm within a short period of time, and this requires a significant amount of energy.

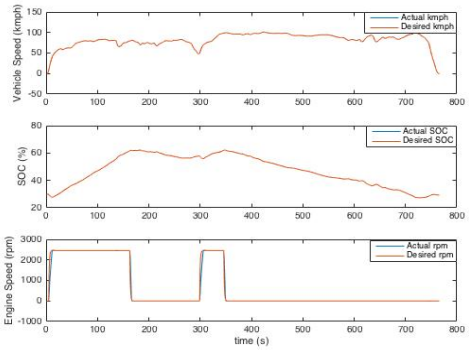
However, earlier HEV's may not have such powerful batteries. Therefore, we repeated our simulation for the highway drive cycle by imposing a lower discharge power limit of 10 kW. From Figure 21 c) and d), it is seen that this power limit is met under steady state conditions. Subsequently, the engine has to remain on for a longer period of time to satisfy the power requirements. Thereby, one sees a substantial decrease in fuel economy to 76.45 mpg.

For a HEV with less than 10 kW of battery bower, any optimization based algorithm would contribute very little to improvement in fuel economy. In essence, both energy sources are greatly unbalanced in terms of capacity as the electrical system is underpowered and the engine must run much of the time to satisfy the power requirements.

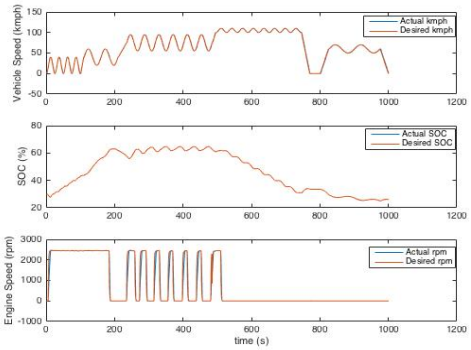
4.2.4 Control Accuracy & Implementation on Vehicles

Our controller ensures that the desired velocity is reached by taking advantage of the power split device to ensure that engine operation is optimized by operating at the optimum engine speed, which is decided based upon our optimal power level. The battery we use has a sufficiently high charge capacity (8.1 Ah). Therefore, while testing our controller with our drive cycle mentioned earlier, we limited the state of charge to a maximum of 62 % such that one can observe the charge and discharge cycles and corresponding operation of the engine in such cases. A comparison between the Actual Vs. Desired Vehicle speed (in kmph), State of Charge (SOC (%)), and Engine Speed (rpm) for each drive cycle are shown in figure 21:

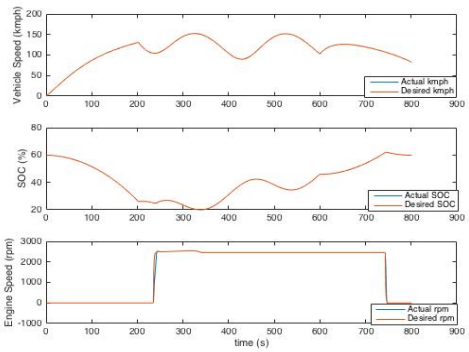
It is clearly seen that the optimal solution for these drive cycles is such that the engine is switched on during the charging period, and off otherwise. This is in accordance with what the algorithm predicts when one remains above all intervals. In addition, the engine is used to satisfy the power requirements. For the cycles shown above, it is seen that the engine effectively handles erratic changes in power demand. The engine speed during this period of operation is around 2450 rpm, which corresponds to the optimal engine power level U^* of 20.9809 kW. It also has the capability of operating at a Higher rpm by taking advantage of intervals as seen for part of the High Speed Cycle. The actual and desired values vary by an average of 0.0397 kmph for vehicle speed, 13.5286 rpm for engine speed, and $3.22 * 10^{-9}$ % for SOC over all cycles. Hence, our optimal controller maintains a sufficiently high level



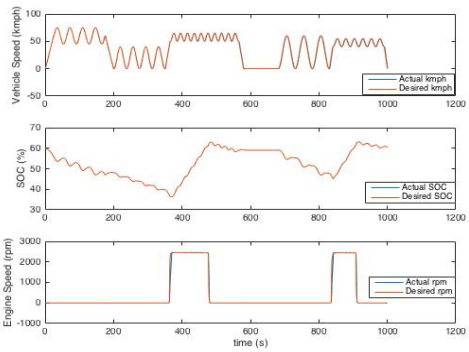
(a) Highway



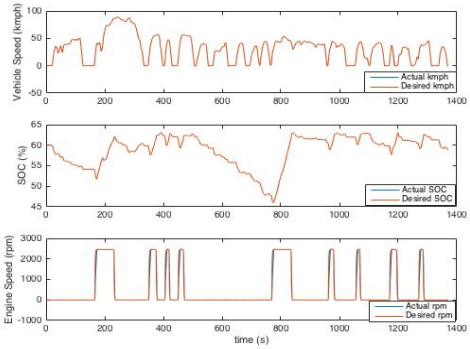
(b) City to Suburb



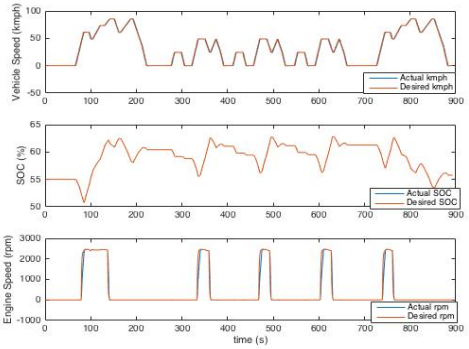
(c) High Speed Cycle



(d) City



(e) UDDS



(f) Japanese1015

Figure 22: Accuracy of Controller across drive cycles

of accuracy.

To implement this on an actual vehicle, it is important to realize that power demand data must be known before implementing the interval back propagation algorithm. There are two ways this information can be obtained.

- Using real time GPS data, one has information about the traffic conditions ahead and distance to destination. Therefore, we can estimate the velocity profile, and provided with vehicle data it is possible to compute the power demand for our route. The important issue to consider here is that we must continuously update the power demand calculation within a sufficiently small period to take into account the traffic variations over time.

- With the help of a learning algorithm embedded in the vehicle controller, one can predict the power demand along a particular route based off information obtained from previous sets of data while traversing the same route. One such example has been shown in [53]. Here, two distinct neural network algorithms are used to optimize HEV performance. One is used to obtain data with regards to the driving environment, while the other is used for optimizing the power split operation through Dynamic programming. In our case, the dynamic programming algorithm used to optimize the power split will use interval back propagation. One of the neural network systems will repetitively run our algorithm over a variety of drive cycles and learn the optimal solution corresponding to each cycle. Meanwhile, the other neural network will learn the necessary power demand data corresponding to each cycle.

CHAPTER V

CONCLUSION

This thesis has presented an Optimization based dynamic programming strategy to improve the fuel economy in Hybrid electric vehicles. The salient feature with this method is that it gives a closed form globally optimal solution for the optimal input, and can be implemented in real time.

The convexity assumption we make on the cost function in Lemma 1 is critical to ensuring that we can derive a closed form globally optimal solution using interval back propagation based dynamic programming. In addition, realize that with this algorithm, we are able to enforce state and input constraints at each and individual step without impacting the convex nature of our cost function. This is a key advantage of Interval back propagation over ECMS. In addition, note that we are able to limit our calculation of the optimal input within interval bounds specified by the states at each time step, rather than compute the optimal input for each state at each time step. Thereby, we save significant computation time in comparison to exhaustive search. Also realize that grouping the discrete states within specified interval bounds would help deal with excessive switching between states.

We Observed that while operating within the constraints imposed by state of charge and maximum engine power, the closed form globally optimal solution for the input is either zero, u^*, or linearly dependent on the state of charge and the average power demand. The simple nature of our optimal input greatly enhances its ability to be implemented in real time. Also, note that when one operates above the top lower bound interval, and the bottom upper bound interval, the optimal input essentially switches between 0 during the discharge cycle and u^* during the charging

cycle. In addition, notice from the results and properties of $f(\xi)$ that the modified cost function decreases with respect to state of charge, which means that at higher states of charge, the vehicle essentially operates in electric vehicle mode, while at lower states of charge, both the engine and motor power the vehicle.

We mentioned that the algorithm takes 15 microseconds to run online when timed using MATLAB for 800 steps of future power demand on a 1.4 GHz Intel Core i5 processor, which proves that it is indeed real time implementable. While implementing our algorithm in real time, we assume a linear relationship between the engine torque and engine speed. Consequently, this yields a quadratic relationship between the engine speed and engine power. We make use of equation (44) to relate the engine speed to the optimal input, and subsequently solve for the engine torque and speed.

We have explained how the interval back propagation algorithm is incorporated within real time optimal controller on the HEV, and briefly discuss the design of engine speed, motor speed, and generator controllers.

We have explained the 6 drive cycles used for testing the algorithm in detail, and observed that fuel economy improves significantly on using the interval back propagation algorithm. Over 6 distinct cycles, a 35.9 % improvement is seen when compared to rule based mode logic control methods under flat road conditions. The unusually high fuel economy for our highway cycle is due to high limits on the battery discharge rate. If this limit is lowered, its seen that fuel economy is reduced considerably since the engine must be switched on more often to meet the cycle power requirement. Our controller also proves to be sufficiently accurate as the speed following and state of charge management coincides with our expectations.

Future work on this topic will involve obtaining the necessary power demand data, as the interval back propagation algorithm needs prior information regarding the power demand. Two methods have been proposed to achieve this goal, namely parsing the GPS data and using neural network based learning algorithms.

REFERENCES

[1] "United states vehicle ownership data, automobile statistics and trends." https://hedgescompany.com/automotive-market-research-statistics/auto-mailing-lists-and-marketing, 2016.

[2] "Us car sales from 1951 to 2015." https://www.statista.com/statistics/199974/us-car-sales-since-1951/, 2015.

[3] "Average annual miles per driver by age group." https://www.fhwa.dot.gov/ohim/onh00/bar8.htm, February 20th 2015.

[4] "Average fuel efficiency of u.s. light duty vehicles." http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/html/table_04_23.html, Nov. 3, 2015.

[5] "Alternative fuel vehicle data." http://www.eia.gov/renewable/afv/supply.cfm?fs=a&sfueltype=gas, April 8, 2013.

[6] "Most efficient alternative-fuel vehicles 2016." https://www.cars.com/articles/most-efficient-alternative-fuel-vehicles-2016-1420683233397/, 2016.

[7] "Advanced vehicles fuel." www.fueleconomy.gov, March 2014.

[8] "Insight overview." http://resources.fiskerautomotive.com/static/downloads/enus/pdf/2012/karma/2012_Fisker_Karma_Vehicle_Specifications-v1.pdf, March 2014.

[9] "Technology file: Hybrid systems." http://www.toyota-global.com/innovation/environmental_technology/technology_file/#h303, 2016.

[10] "Specifications." http://automobiles.honda.com/civic-hybrid/specifications.aspx, March 2014.

[11] C.-C. Lin, H. Peng, J. W. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *Control Systems Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 839–849, 2003.

[12] J. H. Lilly, *Fuzzy control and identification*. John Wiley & Sons, 2011.

[13] G. Feng, *Analysis and Synthesis of Fuzzy Control Systems: a model-based approach*, vol. 37. CRC press, 2010.

[14] Z. Chen, X. Zhang, and C. C. Mi, "Slide mode and fuzzy logic based powertrain controller for the energy management and battery lifetime extension of series hybrid electric vehicles," *Journal of Asian Electric Vehicles*, vol. 8, no. 2, pp. 1425–1432, 2010.

[15] Z. Yi, L. Heping, and W. Huabin, "Torque control strategy for parallel hybrid electric vehicles using fuzzy logic," *Wseas Trans on Syst*, vol. 10, pp. 116–125, 2011.

[16] R. Langari and J.-S. Won, "Intelligent energy management agent for a parallel hybrid vehicle-part i: system architecture and design of the driving situation identification process," *Vehicular Technology, IEEE Transactions on*, vol. 54, no. 3, pp. 925–934, 2005.

[17] Y. Huang, H. H. Lou, J. Gong, and T. F. Edgar, "Fuzzy model predictive control," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 665–678, 2000.

[18] F. Yan, J. Wang, and K. Huang, "Hybrid electric vehicle model predictive control torque-split strategy incorporating engine transient characteristics," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 6, pp. 2458–2467, 2012.

[19] K. G. Murty, *Operations Research Methodologies*. Taylor and Francis Group, LLC, 2009.

[20] A. Panday and H. O. Bansal, "A review of optimal energy management strategies for hybrid electric vehicle," *International Journal of Vehicular Technology*, vol. 2014, 2014.

[21] A. Kleimaier and D. Schröder, "Optimization strategy for design and control of a hybrid vehicle," in *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on*, pp. 459–464, IEEE, 2000.

[22] Y. Zhu, Y. Chen, G. Tian, H. Wu, and Q. Chen, "A four-step method to design an energy management strategy for hybrid vehicles," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 1, pp. 156–161, IEEE, 2004.

[23] F. L. Lewis and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 1995.

[24] R. Shell, *Handbook of industrial automation*. CRC Press, 2000.

[25] S. Verdú and H. V. Poor, "Backward, forward and backward-forward dynamic programming models under commutativity conditions," in *Proc. the 23rd IEEE Conference on Decision and Control (CDC84), Las Vegas, NV*, pp. 1081–1086, 1984.

[26] L. V. Pérez, G. R. Bossio, D. Moitre, and G. O. García, "Optimization of power management in an hybrid electric vehicle using dynamic programming," *Mathematics and Computers in Simulation*, vol. 73, no. 1, pp. 244–254, 2006.

[27] D. Sinoquet, G. Rousseau, and Y. Milhau, "Design optimization and optimal control for hybrid vehicles," *Optimization and Engineering*, vol. 12, no. 1-2, pp. 199–213, 2011.

[28] W. Li, G. Xu, Z. Wang, and Y. Xu, "Dynamic energy management for hybrid electric vehicle based on adaptive dynamic programming," pp. 1–6, 2008.

[29] D. Pei and M. J. Leamy, "Dynamic programming-informed equivalent cost minimization control strategies for hybrid-electric vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 5, p. 051013, 2013.

[30] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 3, pp. 352–363, 2004.

[31] G. Paganelli, S. Delprat, T.-M. Guerra, J. Rimaux, and J.-J. Santin, "Equivalent consumption minimization strategy for parallel hybrid powertrains," in *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 4, pp. 2076–2081, IEEE, 2002.

[32] C. Musardo, G. Rizzoni, Y. Guezennec, and B. Staccia, "A-ecms: An adaptive algorithm for hybrid electric vehicle energy management," *European Journal of Control*, vol. 11, no. 4, pp. 509–524, 2005.

[33] S. Onori and L. Serrao, "On adaptive-ecms strategies for hybrid electric vehicles," in *Proceedings of the International Scientific Conference on Hybrid and Electric Vehicles, Malmaison, France*, pp. 6–7, 2011.

[34] A. Beck, *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*, vol. 19. SIAM, 2014.

[35] J. Arata, M. J. Leamy, J. Meisel, K. Cunefare, and D. Taylor, "Backward-looking simulation of the toyota prius and general motors two-mode power-split hev powertrains," *SAE International Journal of Engines*, vol. 4, no. 1, pp. 1281–1297, 2011.

[36] "Ths ii. section: High-voltage systems motor and generator." `http://www.evworld.com/library/toyotahs2.pdf`, May 2003.

[37] "Prius specifications." `http://www.priups.com/misc/prius-specs.htm`, March 2009.

[38] "Toyota 1nzfxe prius bsfc." `http://ecomodder.com/wiki/index.php/File:Toyota_1nzfxe_prius_bsfc.jpg`, June 2009.

[39] "How cells work." `http://www.jmbatterysystems.com/technology/cells/how-cells-work`, 2014.

[40] N. Omar, P. V. d. Bossche, T. Coosemans, and J. V. Mierlo, "Peukert revisited-critical appraisal and need for modification for lithium-ion batteries," *Energies*, vol. 6, no. 11, pp. 5625–5641, 2013.

[41] O. Tremblay and L.-A. Dessaint, "Experimental validation of a battery dynamic model for ev applications," *World Electric Vehicle Journal*, vol. 3, no. 1, pp. 1–10, 2009.

[42] "Discharge curve." `http://epg.eng.ox.ac.uk/sites/default/files/Adrien/blog060213/discharge_curve.PNG`, June 2013.

[43] "Brushless vs brushed motors." `http://www.dynetic.com/brushlessvsbrushed.htm`, July 2014.

[44] B. Lu, T. G. Habetler, and R. G. Harley, "A survey of efficiency-estimation methods for in-service induction motors," *Industry Applications, IEEE Transactions on*, vol. 42, no. 4, pp. 924–933, 2006.

[45] "Technology file: Hybrid vehicle." `http://www.toyota-global.com/innovation/environmental_technology/technology_file/#h303`, 2016.

[46] J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 6, pp. 1242–1251, 2008.

[47] J. Liu, H. Peng, and Z. Filipi, "Modeling and control analysis of toyota hybrid system," in *International Conference on Advanced Intelligent Mechatronics*, pp. 24–28.

[48] T. Curry, I. Liberman, L. Hoffman-Andrews, and D. Lowell, "Reducing aerodynamic drag and rolling resistance from heavy-duty trucks," tech. rep., M.J. Bradley Associates, October 2012.

[49] D. Hermance, "Toyota hybrid system," (Albany, NY), May 1999.

[50] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on pontryagin's minimum principle," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 1279–1287, 2011.

[51] J. A. Momoh, *Adaptive Stochastic Optimization Techniques with Applications*. CRC Press, 2015.

[52] A. E. B. Jr. and Y.-C. Ho, *Applied Optimal Control*. Taylor Francis Group, LLC, 1975.

[53] Y. L. Murphey, J. Park, Z. Chen, M. L. Kuang, M. A. Masrur, and A. M. Phillips, "Intelligent hybrid vehicle power controlpart i: Machine learning of optimal vehicle power," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 8, pp. 3519–3530, 2012.

[54] "About li-ion batteries." http://www.nexeon.co.uk/about-li-ion-batteries/, 2016.