# Intelligent Energy Management Strategy for Eco-driving in Connected and Autonomous Hybrid Electric Vehicles

# INTELLIGENT ENERGY MANAGEMENT STRATEGY FOR ECO-DRIVING IN CONNECTED AND AUTONOMOUS HYBRID ELECTRIC VEHICLES

BY

AASHIT RATHORE, B.Tech.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

| | |
|---|---|
| Master of Applied Science (2021) | McMaster University |
| (Mechanical Engineering) | Hamilton, Ontario, Canada |

| | |
|---|---|
| TITLE: | Intelligent Energy Management Strategy for Eco-driving in Connected and Autonomous Hybrid Electric Vehicles |
| AUTHOR: | Aashit Rathore<br>B.Tech., (Mechanical Engineering)<br>Indian Institute of Technology Guwahati, Guwahati, India |
| SUPERVISOR: | Dr. Ali Emadi |
| NUMBER OF PAGES: | xvii, 151 |

*This thesis is dedicated to my parents who gave me freedom of career, and to people who made me believe I can achieve my goals.*

# Abstract

Over the past two decades, with the rise of information technology infrastructure and exponential growth of computational capacity, the connected vehicle technology is getting closer to reality day by day. This paradigm shift allows exploiting the algorithms to their full capabilities in performance output by leveraging the environment information. This thesis focuses on developing an intelligent energy management strategy for eco-driving in Connected and Autonomous Hybrid Electric Vehicles (CA-HEV's), which can be implemented in real-time.

The strategy is divided into two layers, i.e. the upper level controller and the lower level controller. The upper level controller can be executed on the remote server. It is responsible for extracting the information from the driver about the trip and the vehicle information using the communication capabilities of the CA-HEV. The gathered information is then utilized by the dynamic programming (DP), which is implemented in a bi-layer fashion to reduce the computation burden on the server. The outer layer of the DP algorithm finds the optimal velocity trajectory, and the inner layer optimizes the power distribution in the powertrain to minimize fuel consumption alongside maintaining charge balance conditions. These global optimal results are evaluated for an ideal environment without any traffic information.

The lower level controller is responsible for real-time implementation on vehicles in

the real world environment and is based on a well-accredited reinforcement learning (RL) strategy, i.e., Q-learning. The RL-based controller optimally distributes the power in a CA-HEV and maintains charge balance conditions. Furthermore, the RL-based controller is also trained on the remote server based on global optimal results obtained from the DP algorithm. The optimal parameter information is then resent to the vehicle's embedded controller for real-time implementation.

Simulations are performed for Toyata Prius (2010) on MATLAB and Simulink, and road information is gathered from SUMO. Simulation results provide a comparative study between the global optimal and the RL-based controller. To validate the adaptiveness of the RL-based controller, it is also tested on two approximate real-world drivecycles, and its performance is compared against global optimal results evaluated using DP.

# Acknowledgements

This research was undertaken, in part, thanks to funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chair in Transportation Electrification and Smart Mobility.

I would like to sincerely thank my supervisor Dr. Ali Emadi, for granting me the opportunity to work with an esteemed group of people at McMaster Automotive Resource Centre (MARC). I am also grateful for his constant guidance, support and motivation during my graduate studies.

I would like to pay special thanks to Dr. Atriya Biswas, who immensely helped me throughout my studies and for being a wonderful friend. I would also like to thank the Stellantis team, who gave me a platform to work on exciting new technology.

I would like to acknowledge all my fellow colleagues at MARC, who provided me an excellent and friendly work environment. I would also like to thank Reihaneh Ostadian and Nathan Emery for their helpful insights. Furthermore, I would also like to thank Joel Roeleveld, Lucas Bruck, Yue Wang, Sumedh Dhale, Adam Lempert, Ankur Shiledar, Shobhit Gupta and Lekhani Ray for sharing their wonderful knowldege with me.

Finally, I would like to express my deep gratitude towards my family for believing in my dreams and never letting me down in my hardest times.

# Notation and Abbreviations

| | |
|---|---|
| ADAS | Advanced Driver Assistance System |
| BEV | Battery Electric Vehicle |
| BSFC | Brake Specific Fuel Consumption |
| CNN | Chaining Neural Netwrok |
| CA-HEV | Connected and Autonomous Hybrid Electric Vehicle |
| CAV | Connected and Autonomous Vehicle |
| DSRC | Dedicated Short Range Communication |
| DQN | Deep Q Network |
| DRL | Deep Reinforcement Learning |
| DP | Dynamic Programming |
| EM | Electric Motor |
| EMS | Energy Management Strategy |
| ECM | Equivalent Circuit Model |
| ECMS | Equivalent Consumption Minimization Strategy |
| FCEV | Fuel Cell Electric Vehicle |
| GM | General Motors |
| GIS | Geographic Imaging System |

| | |
|---|---|
| GPS | Global Positioning System |
| GHG | Greenhouse Gases |
| Hil | Hardware-in-the-Loop |
| HEV | Hybrid Electric Vehicle |
| IT | Information Technology |
| ITS | Intelligent Transportation System |
| ICE | Internal Combustion Engine |
| MDP | Markov Decision Process |
| MARC | McMaster Automotive Resource Centre |
| MPC | Model Predictive Control |
| MG | Motor Generator |
| OCV | Open Circuit Voltage |
| OEM | original Equipment Manufacturer |
| PGS | Planetary Gear Set |
| PHEV | Plug-in Hybrid Electric Vehicle |
| PMP | Pontragyin Minimum Principle |
| RL | Reinforcement Learning |
| RB | Rule Based |
| SHARCNET | Shared Hierarchical Academic Research Computing Network |
| SPaT | Signal and Phase Timing |
| SUMO | Simulation of Urban Mobility |
| SQP | Sequential Quadratic Programming |
| SOC | State-of-Charge |
| TPM | Transition Probability Matrix |

| | |
|---|---|
| V2I | Vehicle-to-Infrastructure |
| V2R | Vehicle-to-Roadside |
| V2S | Vehicle-to-Sensors |
| V2V | Vehicle-to-Vehicle |

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Since the advent of automobiles, engineers and researchers have been continually trying to simultaneously improve the efficiency and power output of the automotive system. Massive developments over the ages on automobiles and their inherent advantages have engendered the automotive industry into a giant economic front. The rising socio-economic status of the masses led to rapid and huge market demand for automobiles. This phenomenon induced the creation and development of several business competitors and a constant need to outperform others. Thus, resulting in a tremendous workforce and financial investment being put into the industries, creating a sophisticated and multi-stage development and manufacturing process.

This chapter focuses on the motivation and background for the research conducted. Furthermore, the thesis contribution, research scope, and thesis outlines are presented in this chapter.

## 1.1 Background and Motivation

Historically, the automotive sector was dominated by conventional vehicles, where the propelling energy came from the internal combustion engine (ICE). Due to the wide availability of fossil fuels and performance benefits, ICE became an integral part of automobiles. A high amount of research was carried out to improve its efficiency on several fronts. ICE works on the principle of burning fossil fuels in an enclosed chamber to produce energy which is then utilized to propel the vehicle. The biggest drawback of the modern ICE is its peak operating efficiency which tends to be around 30-36%. Additionally, combustion of the fuel produces side products such as CO2, NOx, etc., which constitutes greenhouse gases (GHG) responsible for environmental degradation. The tremendously growing number of automobiles in the environment started impacting the fossil fuel reserves and the carbon footprint in the atmosphere. According to the report [6], the transportation sector alone in 2010 was responsible for nearly 23% of total CO2 emissions related to the energy domain, which constituted roughly 6.7GtCO2.

However, in the past three decades, raising awareness of the depleting fossil fuel and environmental degradation has revolutionized the current automotive industries. Leading them towards a greener and sustainable future with stringent governmental policies being enforced over carbon emissions on next-generation vehicles. Emerging hybrid vehicle technologies are a step closer to realizing an idle green and sustainable transportation sector. They offer many advantages in terms of fuel economy, efficiency improvements, and greenhouse emission reductions. Several technologies such as hybrid electric vehicles (HEVs), battery electric vehicles (BEVs), fuel-cell electric vehicles (FCEVs) are getting attention, and a lot of research investment is being put into them [7].

Hybrid vehicles typically consist of two or more power sources coupled together or independently to reinforce each other during their operation to deliver optimum performance output. Recent advancements in electric battery packs and current technological limitations for utilization of other renewable energy sources have predominantly shifted the focus on electric vehicles (xEV).

The technological advancement in electric vehicles can be broadly classified into power electronics, battery systems, controls, and materials [8]. Electric vehicles mainly rely on electric motors for providing propelling torque. Depending upon the energy source, they can be classified into battery electric vehicles (BEVs), plug-in hybrid vehicles (PHEVs), fuel-cell electric vehicles (FCEVs), or just hybrid electric vehicles (HEVs). HEV's typically have an ICE to provide energy to the battery, which provides energy to electric motors to drive the vehicle. There are different configurations in which the ICE and the electric motors can be arranged in the hybrid powertrain resulting in three basic types of configurations: series, parallel, and series-parallel (power-split) [9].

Incorporating an additional energy source in HEV such as ICE supplements the electrical power output allowing the system to operate more closely towards higher efficiency points. However, this inclusion inherently increases systems complexity leading to more complicated solutions. One of the challenging tasks in an HEV is to develop a supervisory control algorithm to optimally manage and distribute the energy output of different power sources such that they operate within their physical constraints satisfying the driver demands and increasing overall system efficiency.

Many control strategies known as energy management strategies (EMS) are available in the literature to tackle this issue, like rule-based (RB), dynamic programming (DP), equivalent consumption minimization strategy (ECMS), model-predictive control (MPC), etc., each with its own set of challenges and limitations [10].

The most formidable challenge in devising a control strategy comes from the fact that the global optimization techniques like DP, pontraygin's minimum principle (PMP), etc., are acausal and thus require future information as a priori to evaluate results [11]. Hence, the incorporation of future information prediction algorithms plays a massive role in the energy management of HEV's to provide near optimum results [12, 13]. Defining an outlook of how a car in the future should be: efficient, predictive, and adaptive. The emergence of technologies such as autonomous and connected vehicles clear a path upon which the foundations of future automobile industries will be laid upon. Connected vehicle technologies such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) communications, along with the inclusion of different sensors, allows the transfer of massive data's in real-time, necessary to facilitate the future driving prediction [14, 15, 16]. They generate the capability in the system to monitor the real-time traffic information, its historical data, road conditions, the behavior of other vehicles, which are then critically analyzed to predict the required information.

Over the past two decades, with the rise of the IT infrastructure and exponential growth of computational capacity, the connected vehicle technology is getting closer to reality day by day. In Europe, for some parts of the local public transportation, communication between vehicles and traffic lights is practically implemented [17]. In the USA, a lot of academic and industrial investment is being put into experimenting with the cross-communication of traffic signal status for security purposes [18].

This paradigm transition allows the researcher to exploit the algorithms to their full capabilities in terms of performance output by leveraging the communicated data to develop more sophisticated algorithms. This motivates the incorporation of intelligent and predictive energy management strategies into the connected vehicles utilizing its advantages and capabilities to minimize energy consumption and reduce emissions.

## 1.2 Thesis Contribution and Research Scope

The main contribution of this thesis is to present a new approach to solving the problem of supervisory energy management strategies for connected and autonomous hybrid electric vehicles (CA-HEV's). Connected and autonomous vehicles offer capabilities to incorporate environmental awareness, future prediction, and cloud computing techniques to tackle energy management. A robust strategy can be developed utilizing the available resources, resulting in as close as possible optimal behavior.

In this thesis, a novel bi-layer eco-driving based energy management strategy is formulated, which can be utilized in real-time applications in CA-HEV's. The bi-layer structure is used to optimize the velocity profile and, ultimately, the vehicle's fuel consumption. The bi-layer algorithm helps reduce the computation time of the algorithm and enables it to be real-time. In the upper layer of the algorithm, a DP algorithm is used to provide the optimal velocity profile and optimal reference state-of-charge (SOC) profile. A systematic approach is taken to solve the problem using DP by decomposing it into two sub-problems, optimizing the velocity, and evaluating the optimal SOC reference profile. This layer can be solved in the cloud-servers facilitated by the CA-HEV's for fast computation as DP suffers from the curse of dimensionality. Furthermore, a learning-based algorithm based on reinforcement learning (RL) is applied to the lower level controller. According to the author's knowledge, it has not been explored yet for the CA-HEV's. The learning-based controller, which can be implemented on the vehicle in real-time, allows the model to be robust and react to uncertainties in the environment.

The methodology presented in this thesis is tested in the simulation environment, where the information gathered by the CA-HEV's is fed onto the algorithm, and the results are evaluated. This imitates the real-life scenario involving the CA-HEV's where they can gather the trip information, the surrounding environment information and communicate with the cloud-server for enhanced safety, mobility, and efficiency. At last, the online simulation results from the RL controller are compared against those obtained from offline DP to evaluate the effectiveness of the RL agent in an unknown environment.

To completely understand the algorithm's advantages and limitations, it is imperative to define the scope of the research. To begin with, this thesis is only based on simulation and modeling work; no hardware analysis or physical testing work is performed. The simulation is conducted for Toyota Prius 2010 vehicle mode, which is a full hybrid electric vehicle with a power-split architecture. However, the algorithm can be developed irrespective of the level of electrification of the vehicle and irrespective of the vehicle's architecture. The developed vehicle model is quasi-static in nature, and for some components, constant operating efficiencies are considered. It is considered that the components are able to follow the feasible requested commands always, and temperature, fatigue, and physical deformities are ignored.

The developed strategy is based on the eco-driving aspect of the CA-HEV's, hence an optimal velocity for a particular starting and final destination, is generated. The optimal velocity is generated using a DP algorithm, which suffers from the curse of dimensionality; hence the state-space grid is designed carefully to balance computation time and performance. For the current thesis work, only certain features of CA-HEV's are considered, i.e., it is assumed that CA-HEV can provide the speed limit for the entire trip. Road grade for the entire trip in this thesis is assumed to be a constant value. Furthermore, it is assumed that the CA-HEV has the required hardware technology that enables it to communicate with the cloud-server (V2V and V2I technology enablers). Also, due to the resources and time limitations, the modeling of vehicle-to-vehicle infrastructure and cloud communication has also been excluded from the scope of the simulations. The interaction of the CA-HEV with other CAV's, traffic roadside units, and pedestrians is ignored, and a single-vehicle scenario is only considered while generating the optimal velocity profile. At last, it is assumed that the results generated after training the RL agent will be communicated back to the vehicle's embedded controller, such that it tunes and optimizes the controller parameters to behave accordingly. However, this thesis does not present the embedded controller tuning. The simulation is only limited to results extraction from the RL agent and its comparison against the DP offline results.

## 1.3 Thesis Outline

The rest of the thesis is divided into six chapters. Chapter 2 discusses the fundamentals of hybrid electric vehicle powertrains and provides a brief introduction to CAV's. Chapter 3 delves deep into the energy management strategies and explains the standard methodologies for conventional and CA-HEV's. Chapter 4 explain the mathematical formulation of the representative vehicle model and defines the optimal control problem for this thesis. Chapter 5 represents the core of this thesis and explains the developed novel intelligent EMS for eco-driving of CA-HEV in great detail. Chapter 6 shows the results evaluated by using the algorithm and provides an in-depth discussion on them. At last, Chapter 7 gives the conclusion for the whole thesis and suggests some of the future work that can be done to improve the algorithm's performance.

# Chapter 2

# Fundamentals of Hybrid Electric Powertrains and Connected and Autonomous Vehicles

This chapter discusses the fundamentals of hybrid electric vehicle powertrains and their various architectures. Furthermore, an introduction to connected and autonomous vehicles (CAV's) in the intelligent transportation system (ITS) is presented.

## 2.1 Introduction

Since the advent of automobiles, people have always wondered about ways to improve the performance and increase the efficiency of vehicles. One of the most prominent aspects to tackle this challenge is to develop sophisticated powertrain technology. In the current era, industries have been focusing on developing modern electric vehicles with complex powertrains. However, the concept of electric vehicles is not new; in fact, it was first successfully introduced in the U.S around the 1890's [19]. During that period of time, the electric vehicle market was flourishing. However, later a drastic transition occurred when the internal combustion engine (ICE) vehicle offered higher performance and better range capabilities. Additionally, the low cost of fossil fuel with its high availability transitioned the market towards ICE vehicles.

The automotive industry under the umbrella of ICE's thrived for decades, and a significant amount of research was carried out. The large-scale production of cars such as Ford and Toyota popularized the vehicles and made it possible for ordinary people to utilize the technology. Automotive industries began making automobiles cheaper day by day and increasing their performance, efficiency, and desirability. The rate at which the technology develops is far slower than its popularity. This led to the biggest downside of the ICE-based vehicles, which came in the form of greenhouse gases (GHG's) and depleting fossil fuels.

Rising awareness of the drawbacks of combustion vehicles and increasing government regulations over GHG emissions redeveloped the interest of automotive industries towards a possible solution in the form of electric vehicles. By the end of the 20th century, industries slowly started working towards a more sustainable approach and started developing electric vehicles. In 1997 Toyota released the hybrid electric vehicle named Prius, which had both the ICE and the electric motors. More companies began to focus on electric vehicles as government legislation began promoting electric vehicles and stringing emission regulations. In 2010, Nissan launched a battery electric vehicle (BEV) named Leaf as a competitor to Toyota's Prius. Furthermore, in 2012 Chevrolet launched a plug-in hybrid electric vehicle named Volt, which became a massive hit in the U.S market. All these vehicles provided much higher efficiency and significantly reduced fuel consumption, which was a step towards a greener and sustainable automotive sector.

An evident difference between the vehicles launched by different brands was in their level of electrification. In electric vehicles, electrical energy is used to provide the propelling energy in combination with the chemical energy from the ICE. Depending upon the size of the electrical energy storage system, they can have various levels of electrification in them. Different level of electrification leads to different advantages and drawbacks depending upon their architecture. Conventional vehicles with ICE fall on one end of the spectrum while full-electric vehicles on the other. They can broadly be classified into five categories, i.e., micro hybrid, mild hybrid, full hybrid, plug-in hybrid, and battery electric vehicle as shown in Fig. 2.1.

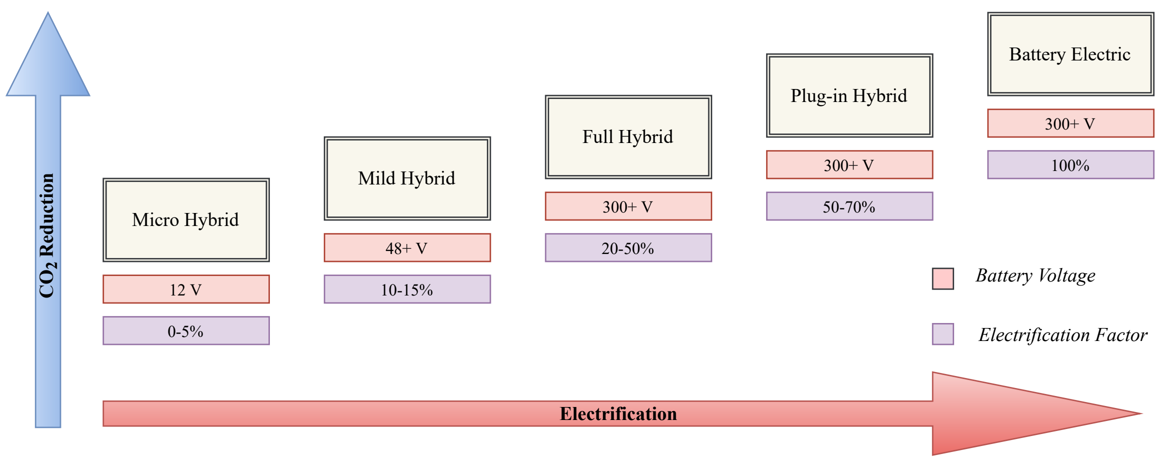


Figure 2.1: Electrification Level for various powertrains (adapted from [1][2])

In the past two decades, full hybrid electric vehicles have received a lot of research consideration and popularity due to their many advantages. Unlike mild hybrid electric vehicles, full hybrid vehicles incorporate a generator motor to turn the engine on and off. The electric motors can also assist ICE power and can also be used for regenerative braking. Furthermore, due to the current limitations on batteries, battery electric vehicles are still in the nascent development phase and require more research work. Plug-in-hybrid electric vehicles can also be considered a viable option; however, it still requires the charging infrastructure built-up before they can be commercialized effectively. Hence, much effort has been put up into increasing the efficiency of the full hybrid electric vehicles. In a full hybrid electric vehicle, the selection of powertrain topology plays a vital role and should be determined by size, weight, driving cycle routine, and performance requirements [20]. The following section discusses some fundamentals of different vehicle powertrain architectural topologies.

## 2.2 Hybrid Electric Vehicle Powertrain Architectures

HEV powertrain architectures are classified based on the arrangement of the electric motors and the ICE. Different architectures offer different advantages and disadvantages and are usually selected based on their application. They are divided into series, parallel, and series-parallel (more commonly known as power-split) architectures. The following sub-sections provide a brief introduction to different configurations. Moreover, for the application of this thesis, power-split architecture was selected due to its multiple advantages that shall be explained below.

### 2.2.1 Series Architecture

In a series HEV, the primary role of the ICE is to act as a supplemental energy-providing source when the battery needs it. In a series configuration, an electric motor (EM2) is the propelling device, which gets the power from the battery. The ICE is connected to the generator motor (EM1), which converts the mechanical power from the ICE to electrical power. There can be multiple ways of positioning the ICE, EM1 and EM2 as shown in Fig. 2.2. In a series architecture, the power flows in a single path, i.e. from chemical to mechanical in ICE, from mechanical to electrical through EM1, and then at last from electrical to mechanical from the battery to the vehicle by EM2.

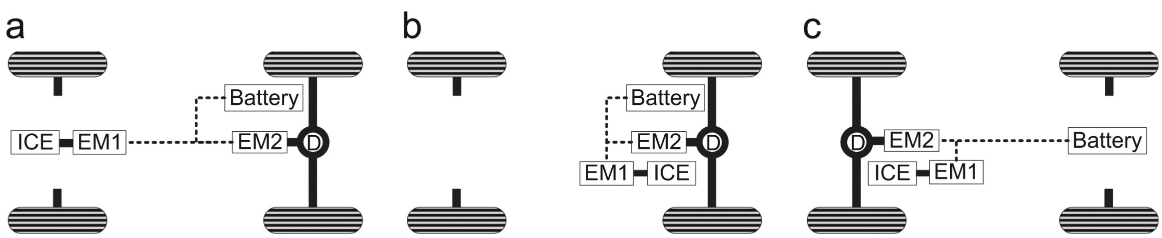


Figure 2.2: Series Configurations (adapted from [1])(Copyright permission granted for thesis/ dissertations from Elsevier, subjected to re-submission of permission if work is published.)

One of the main advantages of the series architecture is that the ICE is decoupled from the vehicle. Hence it can be operated at an optimal point at all, thereby reducing fuel consumption [21]. This type of configuration can be helpful in city drivings, although there are some significant drawbacks in this architecture. One of them is that the EM2 is the only propelling device and requires a large battery, requiring a large engine and generator pair. This adds a lot of weight and inefficiencies to the system.

### 2.2.2 Parallel Architecture

In a parallel HEV, the engine and the electric motor are connected so that at a given instant, any one of them or both can provide the propelling power. Both the ICE and EM2 are mechanically connected to the wheel, which enables them this capability. Here, there can be a variety of ways in which the ICE and the EM can be attached, giving rise to multiple configurations as shown in Fig. 2.3. In a parallel HEV, one of the critical tasks is to determine the power distribution of the ICE, and the EM, which determines the overall efficiency of the system [22].

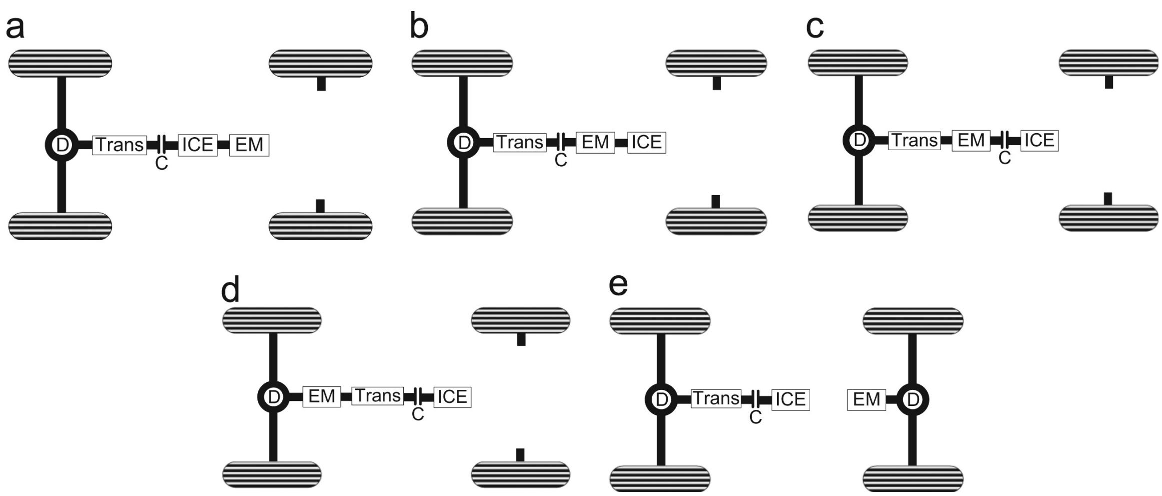


Figure 2.3: Parallel Configurations (adapted from [1])(Copyright permission granted for thesis/ dissertations from Elsevier, subjected to re-submission of permission if work is published.)

These architectures are very suitable for highway driving. As both the EM and the ICE contribute to supplying power to the vehicles, it is possible to downsize both of them and reduce the system's weight. Also, as the mechanical energy from ICE and electric energy from EM can directly be used to propel the vehicle, it increases the efficiency of the system [21]. However, the battery charging is done through the engine, and therefore might be limited, depending upon the exact arrangement of the device.

### 2.2.3 Series-Parallel Architecture

The series-parallel architecture, more commonly known as the power-split architecture, is one of the most widely used architecture in an HEV. It usually involves a planetary gear set to combine the benefits of the series and the parallel architecture. In this configuration, the arrangement of the ICE and the electric motors is such that its power can be divided into two-part: one through the generator to recharge the battery (series path) and the other one to propel the vehicle (parallel path) along with the motor. The power-split is achieved by attaching different components to the sun, ring and carrier of the planetary gears. The power is always flowing through both parts and can be managed to optimize the transfer. In literature, there are many ways in which a power-split architecture can be composed, and they can be classified as input-split, output-split, and compound split [23]. An example of input-split, output-split and compound split architectures is shown in Fig. 2.4(a), Fig. 2.4(b), and Fig. 2.4(c) respectively.

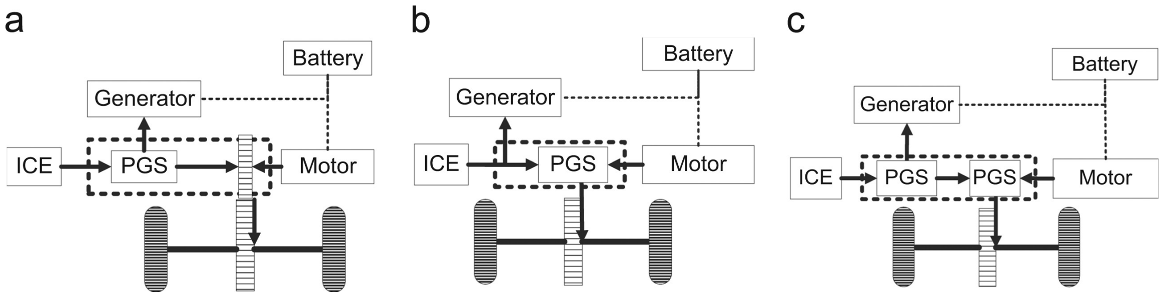


Figure 2.4: Different Series-Parallel Configurations (adapted from [1])(Copyright permission granted for thesis/ dissertations from Elsevier, subjected to re-submission of permission if work is published.)

The main advantage of the power-split architecture is that the engine, generator and motor speeds are decoupled, thus allowing an extra degree of freedom in the system. This extra freedom allows efficient transmission of power during all events by managing the amount of power flow with its direction. However, the power-split is more complicated and expensive than the rest two architectures, and the optimal power distribution problem is a challenging task. The most common arrangement of power-split comes from the Toyota Prius's Hybrid Synergy Drive, which incorporates an ICE, one generator, one motor and one planetary gear set. For the application of this thesis, the power-split architecture has been selected, whose modelling is described in detail in a later chapter.

## 2.3 Connected and Autonomous Vehicles

Connected and autonomous vehicles (CAV) are vehicles that are equipped with technologies that enable them to show autonomous capabilities along with connectivity to communicate with other vehicles, traffic, road, and the cloud servers [24]. The future of vehicles can be thought of in two classes: one where connected vehicles are fully autonomous and drive themselves, and the second one where they still possess manual control while having advanced communication capability to enhance mobility, safety, and efficiency [25]. CAV's are equipped with various communication devices, which enable them to communicate with the external environment. These communications can be divided into two categories, namely vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications [25]. V2V communications are made up of wireless data transmissions where vehicles communicate with each other, regarding their speed, position, travel direction etc., within an ad-hoc network [26]. This helps the driver be aware of the surrounding vehicles and enhances the overall safety and mobility of the traffic. V2V communications will be better than the current embedded system developed by the OEM's for vehicle safety and are dependent on the onboard sensors [27]. Unlike the V2V communications, V2I communications allow the vehicles to exchange information with the roadside units, traffic lights, parking meters, construction sites etc., [28]. They are similar to V2V communications and are implemented using the Dedicated Short-Range Communication (DSRC) frequencies to exchange information [29]. One of the applications of V2I information can be seen in the Signal Phase and Timing (SPaT) control to achieve fuel saving and manage traffic congestion by communication of traffic lights and vehicles [30]. A pictorial representation of CAV's is presented in Fig. 2.5, where the vehicle is communicating to other vehicles(V2V), to the roadside infrastructure (V2R), to the onboard sensors (V2S), and the internet (V2I) [3].

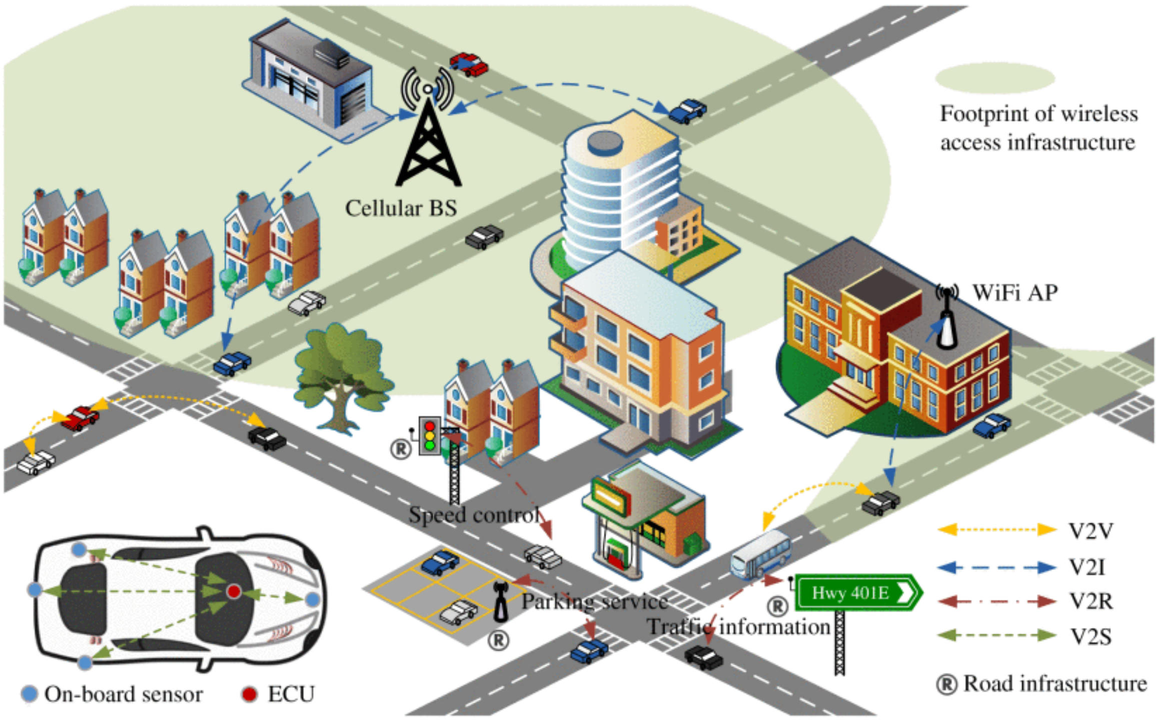


Figure 2.5: Connected Vehicles (adapted from [3] © 2014 IEEE)

CAV's are on the brink of bringing an imminent change that will revolutionize the intelligent transportation sector (ITS), leading them toward a greener, safer, and more autonomous capable environment. ITS has now become a massively international phenomenon, with support from around the world. ITS refers to the application of advanced communication, information and electronics technology to tackle the imminent problems of traffic congestion, mobility, emissions and safety [4]. It is a system where users, roads and vehicles all exchange information with each other to form a collective intelligent environment as shown in Fig. 2.6.

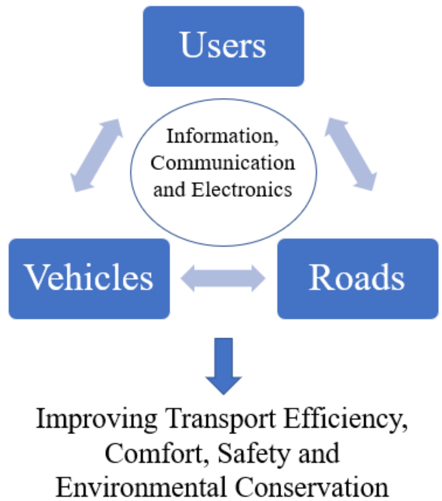


Figure 2.6: Intelligent Transportation System Overview (adapted from [4])

With the massive development of the intelligent transportation sector (ITS) and the underlying technology enablers, vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication are getting progressively possible and feasible. Companies like Google, GM, Uber are continually carrying out research to test the CAV technologies to fast-track their implementation in the real-life environment [31]. CAV's enables the vehicle to communicate for gaining real-time and historical road and traffic data, providing an opportunity to improve robustness, efficiency and safety of the vehicle [32]. Another significant advantage of CAV's results from the fact that it allows the actual randomness of the road condition to be modelled into a simulation environment to reflect actual vehicle performance for better exploitation of energy management strategies [33].

Over the past years, the powertrain components have reached their nearing peak efficiencies. Moreover, the focus has been shifted towards the effect of external factors, which significantly affect the performance and efficiency of the hybrid electric powertrains [34]. These actual road conditions significantly affect the performance of the vehicles in terms of fuel economy. Studies have shown that in the US, Canada and Europe, fuel economy can deviate from 15-25% from lab testing results when compared to actual on-road economy [35]. Hence, it is essential to encompass the environmental effects to correctly estimate the powertrain's performance for its fuel consumption and GHG emissions. The environmental effects can be classified into external factors, driver behaviour or traffic conditions. The external factors involve road slope, speed limits, ambient conditions, etc., where traffic conditions involve other vehicles, traffic lights, and route selection. The driver's behaviour also significantly affects the performance of the vehicle. CAV's technology provides a platform to incorporate these factors into the supervisory energy management algorithms to optimally determine the vehicle's response and estimate a close to real-world fuel consumption with enhanced safety and mobility.

# Chapter 3

# Energy Management Strategies (EMS)

This chapter provides a detailed introduction to the problem of energy management in hybrid electric vehicles. The first section provides an introduction to the energy management strategies (EMS's). The second section contains information on energy management strategies (EMS) for conventional vehicles, while the last section deals with the application of EMS's in CA-HEV's under the Intelligent Transportation System (ITS).

## 3.1 Introduction

Energy management strategies (EMS's) play a vital role in determining the fuel economy and carbon emission in hybrid electric vehicles (HEV's). EMS refers to algorithms developed to decide the power distribution from various energy sources and gear ratios in an HEV at each instant such that optimal desired behaviour can be achieved. In an HEV, the primary purpose of the EMS is to decide the power distribution between the ICE and the EM's, such that vehicle power demands and SOC constraints are satisfied, and an optimal fuel economy can be achieved. Various EMS's are available in the literature, with different advantages and limitations. To understand the role of the EMS controller in an HEV, Fig. 3.1 shows a representative pictorial description.

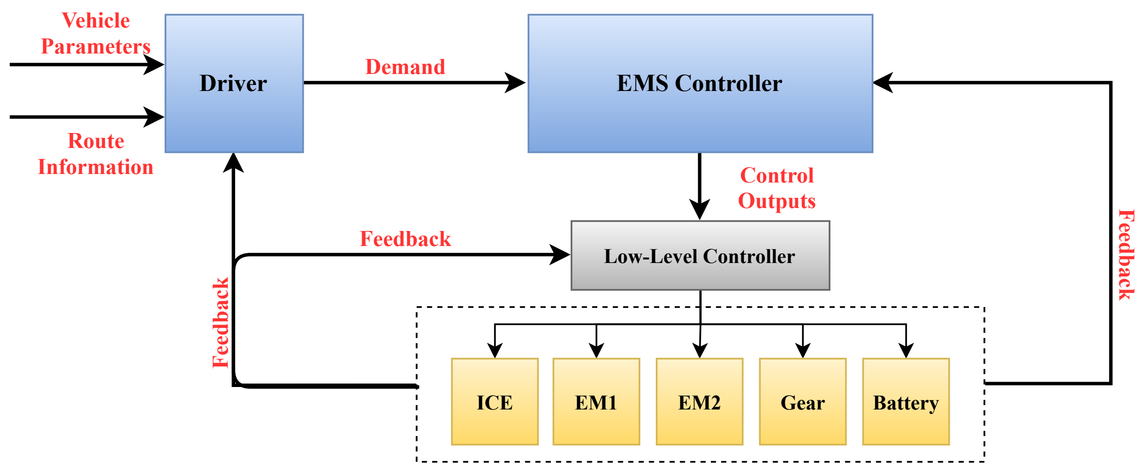


Figure 3.1: EMS operation overview

Fig. 3.1 shows a vehicle model controller, an EMS controller and a low-level controller which controls the individual components in an HEV. The driver model is responsible for modelling the driving characteristic of the driver considering the trip route and vehicle information. This decides the total vehicle power demand, which is then sent to the EMS controller. The EMS controller depending upon the driver request, the component feedback, and the utilized strategy, makes the reference control outputs decisions. These control outputs are sent to the low-level controller, responsible for operating the components, which determines the next state of the vehicle. These next states' outputs are feedback to the controllers as an input from the previous state, and the loop continues.

Now, as mentioned, several available strategies can be modelled in the EMS controller. The EMS selection depends on the vehicle configuration, available future information, computing capabilities of the embedded controller, and the algorithm's limitations. In the next section, the fundamentals of a few strategies are presented for the full HEV.

## 3.2 EMS for Conventional Vehicles

This section contains a brief introduction to some of the typical EMS's available in the literature. Each EMS has different applications, and based on the requirements, the appropriate one can be selected. EMS's can be broadly classified into three categories: rule-based strategies, optimization-based strategies and data-driven intelligent strategies [36][37]. The optimization-based strategies can again be classified into real-time strategies and global-optimal strategies, while rule-based can be of deterministic nature or fuzzy nature [38]. Fig. 3.2 shows the classification of strategies and some of the standard algorithms under each category.

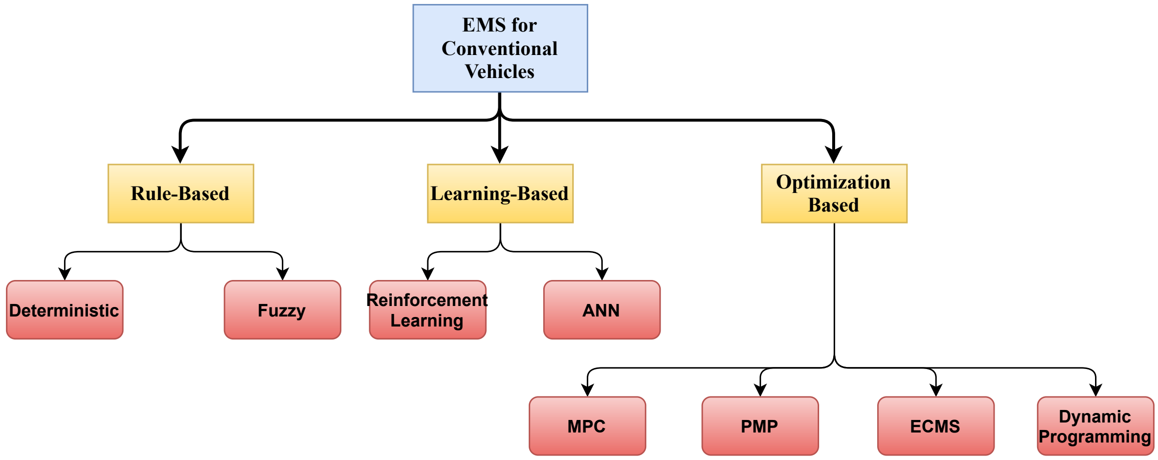


Figure 3.2: EMS for conventional vehicle

### 3.2.1 Rule-Based Strategies

Rule-based strategies are the most common strategy used by the automobile industries due to their simple implementation, and their real-time capability [39]. They are designed based on heuristic rules, intuition and human experience, without any *a priori* knowledge of the drive cycle [38]. Therefore, they are highly time-consuming, tedious and their performance is based on how well its designed [40]. As shown in Fig. 3.2, they are generally divided into two categories: deterministic and fuzzy rule-based.

The central concept behind the rule-based strategies is based on the "load-levelling" technique, where the actual operating point of the ICE is shifted towards the points that can provide the most optimal efficiency along with minimum emissions [41]. However, the rules in the rule-based strategies can be based on many factors. For example, in an HEV, the rules can be based on the power demand from the driver, the SOC of the battery, and the vehicle's velocity. All of them influence how the algorithm will determine the most optimal operating points for different components.

In deterministic rule-based strategies, the rules are implemented using a look-up table or if-else conditions. They are hardcoded, meaning they do not change with change in situation or time. Although in fuzzy rule-based strategies, the rules can be adjusted according to the situation so that they can lead to more optimal solutions [42]. Therefore, it is evident that it is not easy to develop the rules in a rule-based strategy, and it often leads to sub-optimal results, which is one of the most significant drawbacks of these algorithms. Therefore, optimization-based strategies are getting more and more focus due to their near-optimal results and robustness to system state change.

### 3.2.2 Optimization-based Strategies

Optimization-based strategies have been a hot topic in supervisory EMS in HEV for the last two decades. These strategies provide an optimal solution to the problem that has been formulated and decides the control action based on it. These optimization-based strategies can be of two types depending upon their optimality: global optimal strategies and local optimal strategies [43]. The global optimal strategies are acuausal in nature, i.e. they require the complete driving cycle information beforehand to provide the global optimal results. Therefore, they are offline in implementation and cannot be used in real-time. One of the most prominent global optimal strategies is DP, which gives the global optimal results [44]. On the other hand, the local optimal strategies work on the principle of optimizing an objective function instantaneously and are therefore causal in nature. They can be implemented in real-time; however, they do not guarantee global optimal results and provides sub-optimal results. The most recognized real-time instantaneous optimization-based strategies are Equivalent

Consumption Minimization Strategy (ECMS), PMP and MPC [2]. In the following briefs, these algorithms are explained with some of their advantages and drawbacks.

### Dynamic Programming

One of the most famous algorithms used for finding the global optimal solution in the case of EMS's is DP [45]. DP is a numerical method that can solve a problem for its optimal solution by making multistage decisions for its sub-problems [46]. However, DP is a computationally expensive offline algorithm that can only provide the optimum results if the complete drive cycle is known beforehand. Richard Bellman first established DP in 1957, where he introduced the concept of Bellman's principle of optimality [47]. The principle is stated as " *An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision*" [46][47].

As this thesis utilizes dynamic programming to develop an intelligent eco-driving strategy, a detailed mathematical explanation of DP is provided here. First of all, to apply the DP to any problem, it needs to be formulated in a particular manner. In the case of EMS for HEV's, the problem needs to be discretized. Consider the equation to describe the next state of the system as shown in Eq.3.1.

$$x_{k+1} = f(x_k, u_k) \tag{3.1}$$

here, $x_{k+1}$ is the system's next state at the time $k+1$, and $u_k$ is the action taken at the time $k$. Let's take a control policy over the problem horizon as $u =$

$\{u_0, u_1, u_2, ....u_{N-1}\}$ for the cost function $J$. The cost function $J$ is defined as:

$$J_\pi = g_N(x_N) + \phi_N(x_N) + \sum_{k=0}^{N-1} L_k(x_k, u_k) + \phi_k(x_k) \tag{3.2}$$

$J_\pi$ is the total cost of the function over the horizon of $N$ steps. Here $g_N(x_N) + \phi_N(x_N)$ is the terminal cost of the function, i.e. it represents how good it is to be at that state at the end of the horizon. $L_k(x_k, u_k)$ is the instantaneous transition cost (also called as arc cost), which tells the cost of transitioning from state $x_k$ by taking control action $u_k$. $\phi_k(x_k)$ is a penalty term generally included in the cost function to enforce state constraints over the system, so it does not take infeasible states. The optimal policy that results from optimizing the cost function can be written as shown in Eq. 3.3 for the minimization problem.

$$u^* = arg\ min_{\pi \in \Pi} J_\pi \tag{3.3}$$

Here, $u^*$ is the optimal control action sequence that minimizes the cost function over the given horizon, and $\Pi$ is all admissible control policies. The DP finds an optimal policy by working backwards on the time horizon, and finding the optimal cost-to-go at each time step and selecting and storing those control sequences in a matrix $u^*$, representing the optimal actions at each time instant and the state $x_k$. The cost-to-go $Y$, from time $k$ and state $x_k$ for the problem is defined as shown in Eq. 3.4.

$$Y_k(x_k, k) = g_N(x_N) + \phi_N(x_N) + \sum_{k=k}^{N-1} L_k(x_k, u_k) + \phi_k(x_k) \tag{3.4}$$

Therefore, as the DP starts from the last step in the backward simulation, the terminal cost of each state is the optimal cost-to-go at the $N$ time. The optimal cost-to-go at a given time and given state are represented as $Y(x_k, k)$. Now, for the $N-1$ time step, the optimal control sequence can be evaluated by finding the optimal cost-to-go, which can be written as shown in Eq. 3.5.

$$Y(x_k, N-1) = arg\ ming_N(x_N) + \phi_N(x_N) + \sum_{\substack{k=N-1 \\ u \in U_k}}^{N-1} L_k(x_k, u_k) + \phi_k(x_k) \tag{3.5}$$

Similarly, for the $N-2$, the optimal cost-to-go can be written as:

$$\begin{gathered} Y(x_k, N-2) = arg\ ming_N(x_N) + \phi_N(x_N) + \sum_{\substack{k=N-2 \\ u \in U_k}}^{N-1} L_k(x_k, u_k) + \phi_k(x_k) \\ \implies Y(x_k, N-2) = arg\ min(L_k(s_k, u_k) + \underset{u \in U_k}{Y_{N-1}}(f_k(x_k, u_k), u_k)) \end{gathered} \tag{3.6}$$

Hence, by understanding the above equation, it can be seen that optimal cost-to-go at a particular time-step can be written as a summation of the arc-cost and the optimal cos-to-go at one plus step. Thereby, by preceding back in the time horizon, the DP can calculate the optimal policy over the horizon. The optimal cost-to-go at the last iteration, i.e. $Y(x_1, 1)$, gives the optimal cost of the problem and its argument the optimal control policy. These evaluated and stored optimal cost-to-go matrix can be used in forward simulations by mapping the optimal cost with its optimal control sequence and utilizing interpolation wherever necessary.

As already mentioned, the DP is one of the most widely regarded algorithms for finding the optimal benchmark solutions in EM for HEV's. However, as the

DP algorithm formulation indicates, the algorithm's complexity depends on state-space and control-space. Therefore it makes it computationally very expensive, and the DP suffers from the curse of dimensionality. Furthermore, due to its backward propagation in the horizon, it is acausal in nature and requires a priori knowledge of the drive cycle. Hence, the DP is used as a benchmark for testing and validation of the real-time implementable computationally cheap and sub-optimal algorithms.

**Equivalent Consumption Minimization Strategy**

ECMS is one of the most widely investigated real-time strategies due to its straightforward implementation being computationally inexpensive. ECMS is a generic algorithm that works on the principle of instantaneous optimization of power distribution while respecting global and local boundary conditions on state and action vector space [48]. It is based on the concept of converting electrical energy consumption into equivalent fuel consumption and then minimizing their sum [49]. It is an instantaneous optimization approach that gives a sub-optimal solution. For the case of EM in HEV's, the optimal solution is decided based on minimization of equivalent fuel consumption of the powertrain to achieve maximum fuel economy [50]. Equivalent fuel consumption is the summation of fuel used by the engine, and the equivalent fuel that will be consumed by the electrical path to bring back the battery to its original SOC value (as it needs to in charge sustaining mode) in the future by the current engine running conditions [51]. As the comparison of fuel energy and electrical energy cannot be made directly, an equivalence factor was introduced. To physically understand the EF, it can be regarded as a sequence of efficiencies by which combustion energy is converted into electrical energy [46]. And therefore, it changes with changing operating points of EM and ICE. Hence accordingly, it can be considered that the overall fuel consumption is a function of EF and using systematic operations, the value of EF can be determined for the minimization of objective function constrained to sustainable SOC [52].

As mentioned, the ECMS is an instantaneous optimization of the equivalent fuel consumption, its cost function can be formulated as shown in Eq. 3.7.

$$J_t = \dot{m}_{fuel}(u) + SP_{elec}(u) \tag{3.7}$$

Here, $\dot{m}_{fuel}(u)$ is the fuel consumption of the ICE in a HEV, at given instant $t$, with the control policy being $u$. The $S$ is the equivalence factor, and the $P_{elec}$ is the electric power consumed at the given instant which is also a function of the control action. Instant minimization of the cost function $J_t$, at the given instant, yield the optimal control outputs for the given step and can be written as shown below:

$$u^* = arg \min_{u \in U} J_t \tag{3.8}$$

ECMS algorithm is susceptible to the equivalence factor and determines the behaviour and performance of the algorithm. Hence, much research has been carried out, and the emphasis has been on formulating a robust and accurate way to predict the value of $S$. To tackle this, the concept of adaptive ECMS came, where different techniques to evaluate the value of the equivalence factor were suggested [53] [54]. Although its implementation is straightforward, due to its sensitivity towards the equivalence factor and its sub-optimal results, research is now shifting towards more intelligent and robust algorithms.

### Pontryagin's Minimum Principle

PMP is an optimal control method that can be applied in the case of energy management for HEV's [55]. It provides a set of necessary conditions for the trajectory to be an optimal one; however, it is not a sufficient condition. Therefore, the optimal policy will always satisfy the PMP criterion, but the converse is not true. The PMP works on the principle of instantaneous optimization of the cost function that is formulated in terms of Hamiltonian of the function [53]. For the system, with the two-point boundary conditions as $x(t_0) = x_0$ and $x(t_N) = x_N$, and the cost function $(J)$ as shown in Eq. 3.9, the Hamiltonian $(H)$ can be defined as:

$$\begin{aligned} J &= \phi(x(t_N)) + \int_{t_0}^{t_N} L(x(t), u(t), t), dt \\ \implies H(x(t), u(t), \lambda(t), t) &= L(x(t), u(t), t) + \lambda(t)^T.f(x(t), u(t), t) \end{aligned} \tag{3.9}$$

where, $f(x(t), u(t), t)$ is the system's governing dynamic equation, $L(x(t), u(t), t)$ is the instantaneous cost, $\lambda(t)$ is the co-state of the system represented by a vector of variables. According to the PMP criterion for the EM of the HEV, if $u*$ is the optimal control policy, then the following needs to hold true [55]:

$$
\begin{gathered}
\dot{x}^*(t) = \frac{\partial H}{\partial \lambda} = f(x^*(t), u^*(t), t) \\
\dot{\lambda}^*(t) = -\frac{\partial H}{\partial x} \\
x^*(t_0) = x_0 \\
x^*(t_N) = x_N \\
H(x^*(t), u(t), \lambda^*(t), t) \geq H(x^*(t),^* u(t), \lambda^*(t), t) \forall u(t) \in U(t), \forall \in [t_0, t_N]
\end{gathered}
\tag{3.10}
$$

Now, in the case of EM for full HEV, due to the same initial and final boundary condition for the SOC, the $-\frac{\partial H}{\partial x}$ term becomes 0. Hence, the co-state for the optimal solution will be constant during the whole trip. Unfortunately, as the PMP only provides the necessary conditions, the optimal $\lambda$ cannot be found out beforehand and needs to be tuned or found out by using the shooting method. Hence, in real-time applications, the PMP provides sub-optimal results; however, if optimal $\lambda$ is pre-calculated, the PMP provides the global optimal solutions, which is not possible in real-time applications.

**Model Predictive Control**

In recent years, MPC has drawn the attention of researchers due to its robustness to environmental changes and increasing computational capabilities. They can be used in systems with higher-order dynamics and non-linearity. MPC-based strategies combine the advantages of global and instantaneous-based optimization techniques, and they can be implemented in real-time [56]. They are capable of handling multiple state constraints, and multivariable optimization [57]. They work on the principle of iteratively evaluating the optimal control for a receding horizon and then utilizing only first, or some initial control actions [58]. For the application of EM for HEV, the MPC techniques have been used in combination with other techniques like DP, RL, ECMS in the literature to exploit the advantages of the MPC [59][60][61]. The MPC provides the capability to predict the optimal future control inputs and be real-time implementable; however, this leads to its biggest drawback. The MPC technique is highly dependent on the accuracy of the prediction models, actual data, and horizon length, which significantly affect the performance. Furthermore, as it provides the optimal solution for the receding horizon, the overall solution is sub-optimal.

### 3.2.3 Learning-based Strategies

Apart from the rule-based and optimization-based strategies, where rules and optimization techniques govern the EM of HEV, the learning-based strategies are based on machine learning techniques [62]. These strategies are not a-priori, i.e. they do not require future information, can be model-free, and provide robustness to environmental changes [63]. Learning-based strategies can learn from the data pool collected from the real world or can learn from the historical data of the vehicle, which allows them to be real-time applicable [64]. The learning-based strategies provide sub-optimal results compared to global optimal strategies; however, with appropriate formulation and enough training, they can achieve results very close to optimal results [65]. The learning-based EMS for HEV's is an up-and-coming field due to its capability to learn, adapt and improvise according to the outside disturbances [66]. A detailed explanation of the most common learning-based strategy, Reinforcement Learning (RL), is given in the subsequent section.

### Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning algorithm that can be used for the application of energy management and has gained much attention in recent years due to its similarity with DP [67][68]. It can mimic the human decision-making capability, which improves over time as it gets more experience [69][70]. In the RL, there is a decision-maker controller that is known as the *agent*, and it interacts with the system (outside environment), which is known as the *environment*. The agent and the environment continually communicate with each other in terms of states, action and reward [71]. The interaction between the agent and the environment, which constitutes a sequential decision-making problem, is generally modelled as Markov Decision Problem (MDP) as shown in Fig. 3.3 [72].

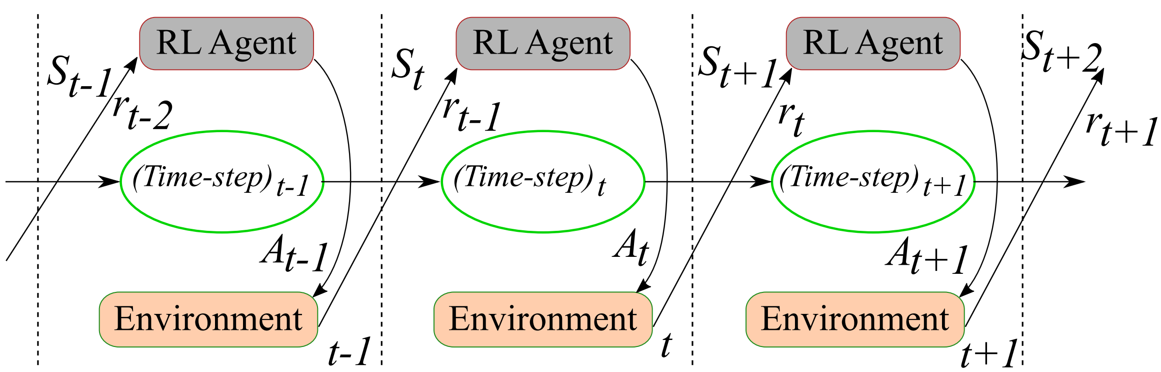


Figure 3.3: Interaction of Agent and Environment

At a given time-step in either a finite or an infinite horizon episode, the agent observes only those states of the environment ($S_t$), which are indispensable for deciding actions ($A_t$). The agent will observe two events at the next time-step ($t + 1$) due to the actions $A_t$ imparted on the environment at the current time-step ($t$).

- The agent will receive a reward $r_t$ from the environment.

- The agent will see the environment ($\mathscr{E}$) making the transition to a new state $S_{t+1}$.

For a finite horizon episode problem, as shown in Fig.3.3, the process mentioned above continues until the agent reaches the terminal state $S_T$ and receives a cumulative return ($\mathscr{R}_t$) at the end of an episode. The return ($\mathscr{R}_t = \sum_{k=1}^{T} \gamma^k r_{t+k}$) is denoted as the return caused by the action $A_t$. The Return $\mathscr{R}_t$ can be re-written as following equation:

$$
\begin{aligned}
\mathscr{R}_t &= r_t + \mathscr{R}_{t+1} = r_t + \gamma r_{t+1} + \mathscr{R}_{t+2} \\
&= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \mathscr{R}_{t+3} \\
&= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{T-2} r_{T-2} + \mathscr{R}_{T-1}
\end{aligned}
\tag{3.11}
$$

, where $\gamma$ is the discount factor ($\gamma \in (0, 1]$) regulating future rewards' contribution to the computation of return from the current time-step.

Due to application of an action $A_t \in \mathcal{A}$ by the agent at a state $S_t \in \mathcal{S}$, the environment makes a *transition* from $S_t$ to a new state $S_{t+1} \in \mathcal{S}$., based on a probability distribution over the set of possible transitions. The transition probability matrix is referred as $TPF : S' \times A_t \times S_t \rightarrow [0, 1]$, i.e. the probability of landing on state $S'$ after making control action $A_t$ in state $S_t$ is denoted by $TPM(S_t, A_t, S')$. $TPM$ satisfies the following mathematical equation for the entire feasible action-state and state-space.

$$\begin{aligned} 0 \leq TPM(S_t, A_t, S') \leq 1 \\ \sum_{S' \in \mathcal{S}} TPM(S_t, A_t, S') = 1 \end{aligned} \tag{3.12}$$

The transition probability matrix is conceptualized as the conditional probability of landing in state $S'$ at time $t+1$ if action $A_t$ was selected at state $S_t$ at time $t$ as $TPM(S_t, A_t, S'_{t+1}) = \mathscr{P}(S'_{t+1}|S_t, A_t)$. The transition probability matrix and the reward function are the two indispensable components of the Markov decision problem.

The rule which governs the agent's choice of action $(A)$ for the observed state $(S)$ is known as the control policy $(\pi)$. Policy is a mapping from n-dimensional state space $\mathscr{S} : S_1, S_2, \cdots, S_n \mapsto \mathbb{R}^n$ to m-dimensional action space $\mathscr{A} : A_1, A_2, \cdots, A_m \mapsto \mathbb{R}^m$. The policy $(\pi)$ of an RL agent is optimized continuously until it converges to the optimal policy $\pi^*$.

*Obtaining Optimal Policy in MDP*

The agent takes the assistance of two decision-making parameters for optimizing its control policy.

*State-value function*

First, the state-value function assists the agent in estimating the goodness of a certain state quantitatively. Second, the action-value function quantitatively estimates how good it to take certain action in a certain state. A value function of a state $\mathbb{S}$ under the policy $\pi$, denoted as $\mathbb{V}^{\pi}(\mathbb{S})$, is the expected cumulative return at the end of the episode from state $\mathbb{S}$ onward and following the policy $\pi$ thereafter.

$$\mathbb{V}^{\pi}(\mathbb{S}) = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} | S_t = \mathbb{S}\right\} \tag{3.13}$$

*Action-value function*

Similarly, an action value function, denoted as $\mathbb{Q}^{\pi}(\mathbb{S}, \mathbb{A}) : \mathbb{S} \times \mathbb{A} \to \mathbb{Q}$, is the expected return at the end of an episodic MDP when taking action $\mathbb{A}$ at the state $\mathbb{S}$ and following policy $\pi$ thereafter.

$$\mathbb{Q}^{\pi}(\mathbb{A}, \mathbb{A}) = \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} | S_t = \mathbb{S}, A_t = \mathbb{A}\right\} \tag{3.14}$$

The state-value function satisfy the following recursive property:

$$\begin{aligned} \mathbb{V}^{\pi}(\mathbb{S}) &= \mathbb{E}_{\pi}\left\{r_t + \gamma r_{t+1} + \gamma^{2} r_{t+2} + ...| S_t = \mathbb{S}\right\} \\ &= \mathbb{E}_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} | S_t = \mathbb{S}\right\} \end{aligned} \tag{3.15}$$

*Q-learning Algorithm*

The above description of RL provides the conceptualization and formulation of the algorithm and explains the details of the theory. The next task that remains is to find the optimal control policy using the RL algorithm. There are certain methods by which the RL algorithm can be implemented to find the optimal control policy like Temporal Difference learning, Deep Q-Network (DQN), SARSA, actor-critic, Q-learning etc. [71][66][73][74]. This thesis utilizes the Q-learning algorithm to implement the RL for the EMS.

In the real-time implementation of the RL algorithm, the instantaneous action is chosen by selecting the argument for the minimum action-value function at every state in the episode. It is based on the local optimization of the action-value function; however, it should be understood that the action-value function is the long-time cumulative return instead of the instant reward. Hence, it considers the future states while minimizing instantaneously and can provide global-optimal results if each action-value function for every state-action pair was known accurately beforehand.

With a full model of the MDP, the accurate evaluation of the action-value function can be achieved. However, without the TPM, it becomes cumbersome to do so. Therefore, there is a need for a systematic update of the action-value function ($Q(s,a)$) that can lead to accurate estimation. This is where the Q-learning algorithm comes into the picture based on Bellman's principle of optimality and provides an equation to update the $Q(s,a)$ as shown in Eq. 3.16.

$$Q_{k+1}(s_t,a_t) = Q_k(s_t,a_t) + \alpha \left\{ r_t + \gamma \max_a Q_k(s_{t+1},a) - Q_k(s_t,a_t) \right\} \tag{3.16}$$

In Q-learning algorithm, the $Q(s,a)$ is a key parameter for all state-action pair, and is stored as a Q-matrix, updated by the above-mention rule.

## 3.3 EMS for Connected and Autonomous Hybrid Electric Vehicles

This section contains detailed information about the EMS for CA-HEV's under the ITS. As mentioned, the CA-HEV's provides the capabilities to incorporate external factors into the EMS with the help of communication between the host and the surrounding, resulting in the different formulation of the energy management objective in the CA-HEV's. The above-listed techniques for energy management of conventional HEV's can still be used in the EMS for CA-HEV's, although in a different manner. Generally, EMSs under the ITS can be divided into two categories: Single-vehicle scenarios and multi-vehicle scenarios [75]. The figure shown in Fig. 3.4, provides an outlook for the energy management for CA-HEV's under the ITS.

In single-vehicle scenarios, only the ego vehicle (vehicle into consideration) is taken into account, and inter-vehicle interactions are not considered. In multi-vehicle scenarios, the EMS can be formulated for the whole platoon of vehicles or vehicle following scenarios are considered by accounting for inter-vehicle communications. This thesis focuses on the EMS for the ego vehicle, without its interaction with other vehicles. Therefore, in the subsequent subsections, the EMS of CA-HEV under single vehicle scenarios is explained in detail, and the multi-vehicle scenarios are not discussed.

### 3.3.1 Single Vehicle Scenarios

In single-vehicle scenarios, the primary objective is to minimize the fuel consumption and/or carbon emissions of the ego HEV by optimizing the power split considering

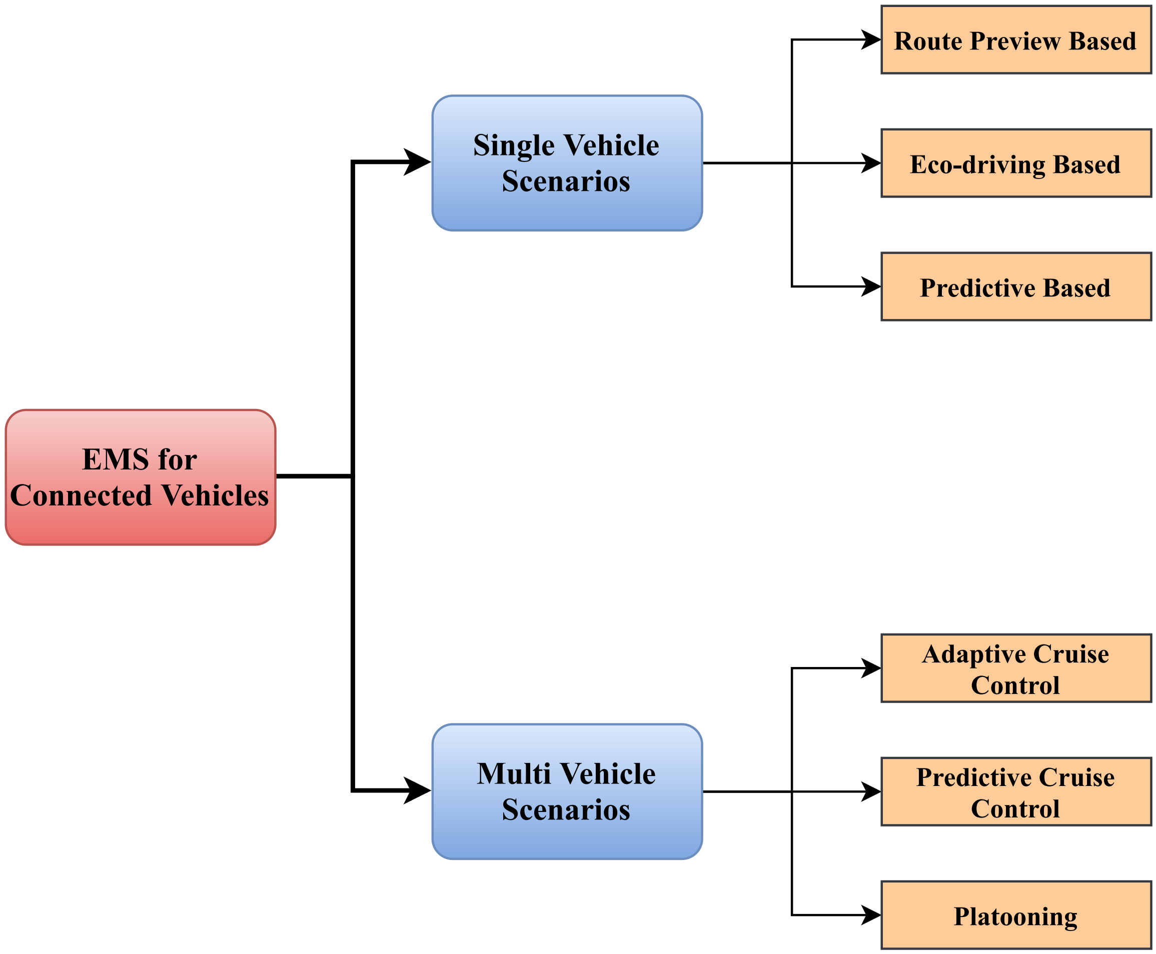


Figure 3.4: EMS for Connected vehicles

overall traffic or road condition information's [76][77]. They can be achieved by developing a single integrated upper, and lower level or as two separate layers exchanging information [78]. Availability of future information or its prediction in the defined horizon allows the acausal EMSs to exploit the data to enhance their performance in real-time, providing some robustness into the system. Many factors like selection of optimization algorithm, control objectives, the timescale of prediction, the fidelity of models play crucial roles in determining the performance of the EMSs [77]. One way of exploiting the full potential EMS is by collecting real-time data from the traffic to analyze and process them on the cloud servers and then feeding the optimal behaviour for the vehicle to the drivers as guidelines [79]. Depending upon the type of information gathered by the vehicle, the EMS for single-vehicle scenarios can further be classified into three types: Route-based, predictive based, and Eco-driving (velocity-based) [76]. The following section provides a brief introduction to all the three types of energy management in CA-HEV's.

#### Route-Based EMS

Route-based EMSs are devised for a longer time horizon considering road data, speed limits, average traffic data, etc., for a target trip to minimize fuel consumption or travel time. By obtaining the upcoming trip information, the performance level of the EMS can be significantly improved [80]. It is understandable that the route consisting of more traffic congestion, stop signs, and higher road grade will require more energy and hence more fuel consumption. In literature [81], the authors utilize the stochastic optimal control problem of HEV for energy management formulation of multiple routes incorporating uncertainties in traffic information with varying levels of available future data. A path planning strategy is proposed in [82] to provide the shortest path and the minimum travel time path by utilizing real-time communication through V2V, V2I and the cloud server. In short, the route-based EMSs examine the effects of available information for a longer horizon to optimize the performance in target trips [76]. Furthermore, they can be beneficial for increasing the response time of emergency vehicles and also help increase efficiency in collaborative parking trajectory planning for autonomous vehicles [33].

### Predictive EMS

Predictive EMS, as the name suggests, can predict future information with some certainty to take full advantage of CA-HEV's in the vehicle's energy management. The predictive behaviour of the algorithm incorporates robustness in the system to the external disturbances and allows for more efficient control concerning changes in the driving and road conditions [76]. Authors in [83] developed a predictive energy management controller utilizing the real-time road information available in the city to predict the velocity profile for the vehicle. Furthermore, a reference SOC estimation technique is developed with the help of the Monte Carlo strategy over the prediction horizon to enhance the vehicle's fuel economy. Similarly, the authors in [84] focus on predicting the velocity profile of the ego vehicle by utilizing a chaining neural network (CNN) and ECMS with the help of information extracted from V2V and V2I communications. In brief, the predictive EMS tries to take full advantage of information gathered to forecast the critical data required in the energy management to yield optimal results by adapting to the changing environment in real-time [76]. However, prediction accuracy is still one of the major concerns in these algorithms as they can significantly affect the results.

### Eco-driving Based EMS

Eco-driving is a cost-effective, relatively low-cost and time-efficient method to improve the fuel economy of the vehicle up to 45% [85]. A variety of factors such as acceleration/deceleration of the vehicle, idling time of the engine, road grade and speed limits, route choice, vehicle parameters play crucial roles in devising the eco-driving based EMS [86]. Eco-driving constitutes formulation of optimal control problem such that it results in optimal velocity profile for the specific driving route considering the energy consumption. The devised velocity profile is then tracked, and generally, the inner controller decides the power split between the electrical and mechanical sources [78]. A more common approach is to develop two separate layers: an upper layer and a lower layer to find the optimal velocity profile and perform energy management, respectively [78]. A typical case is shown in [87], where the authors develop a two-level control strategy to control the velocity and power distribution of the vehicle. The higher level utilizes a nonlinear time-varying Krylov subspace method to find the optimal vehicle velocity trajectory. The lower level decides the torque split with gear shift by using PMP in addition to the numerical framework of MPC. The literature aimed to improve computational efficiency by decoupling a bigger problem and achieve fuel economy improvements. A similar two decoupled layer algorithm can be found in [88], where the authors use signal phase timings information to generate optimal cruising velocity profiles for a group of connected vehicles. They utilize an MPC-based framework in the upper level to generate the velocity profiles that comprise fuel economy and safety and use a DP algorithm in the lower level for energy management. The figure in Fig. 3.5 shows a usual representation of the bi-layer structure of the Eco-driving implementation in an HEV.

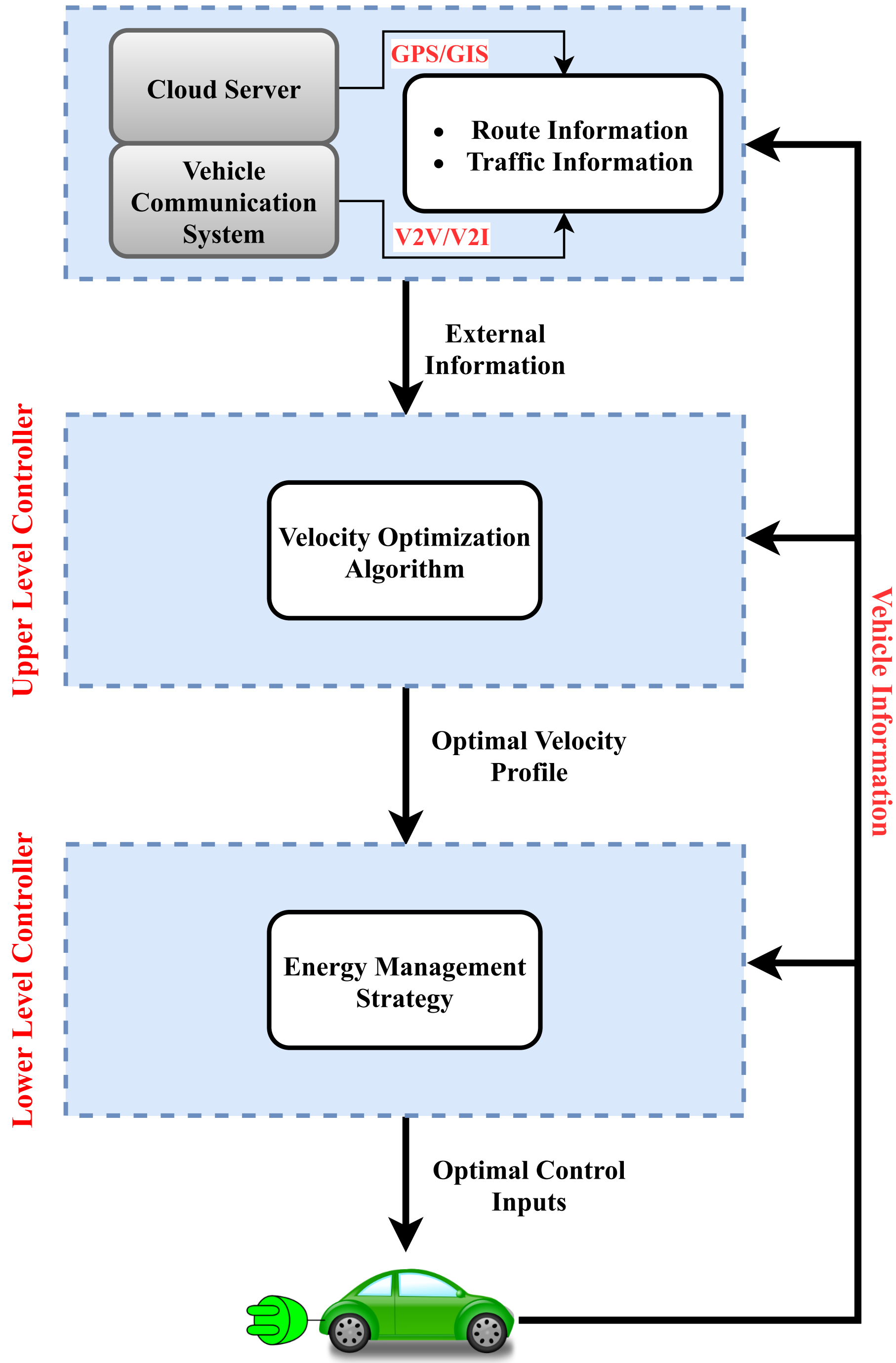


Figure 3.5: Overview of Eco-driving based EMS

Authors in [89] also developed a two-layer algorithm based on stochastic MPC in the upper layer and ECMS in the lower layer utilizing the signal phase timing. They additionally introduced some random error modelled in the form of a Markov process to add more realistic behaviour in the algorithm with improved robustness. In a two-layer structure, the velocity generated by the upper layer decides the power demand of the vehicle and is tracked by the lower layer through energy distribution. The authors in [90] introduce a flexible power demand architecture, where they propose some flexibility in tracking the power demand generated by the upper layer so that the fuel economy can be improved. PMP solves the optimization problem in the lower level in this brief. Apart from the decoupled separate layer, researchers have also developed integrated layers devising the optimal velocity profile based on energy consumption for eco-driving scenarios [91]. In literature [79], the authors solve the optimal control problem using the DP algorithm that shows an improvement of 5%-15% in fuel economy. In the mentioned paper, the driver is first required to send the trip information to the cloud server, which then collects the road and traffic conditions to generate an optimal velocity profile by minimizing the fuel consumption using the DP. Literature [92] is based on integrated two-layers for optimization of the velocity profile and fuel consumption in the rolling terrain scenario. The algorithm takes input road information such as speed limits and gradients to optimize the vehicle's acceleration at the vehicle level. It optimizes the power-split ratio at the powertrain level. An improvement of 5%-8.9% on major arterials is shown compared to conventional HEV and 15.7%- to 16.9% on collector roads.

In summary, eco-driving based EMS is to devise a control algorithm that can find the optimal velocity profile by collecting necessary input data for a given trip and incorporating it into the EMS controller to minimize fuel consumption. As mentioned, eco-driving can significantly reduce the fuel consumption of the vehicle and increase the safety and mobility of traffic [85][91]. Although, it also depends upon the data obtained via V2V and V2I communications and driver's driving characteristics which influence the level of fuel economy that can be enhanced. Due to easy-implementable and low-cost advantages, the eco-driving method has received critical attention from researchers over the past two decades, leading to extensive research in the field. This thesis also focuses on developing an intelligent eco-driving based EMS for the CA-HEV's, and its in-depth formulation is explained in the later chapter.

# Chapter 4

# Vehicle Modeling and Control Problem

A representative vehicle model is an essential component for the development of EMS as it governs the accuracy of the vehicle's behaviour. In this chapter, the formulation for the vehicle model is explained in detail. Furthermore, as the topic of this thesis is the EMS of CA-HEV's, the optimal control problem formulation for the EMS is also discussed in this chapter.

The main components in the HEV powertrain are the propelling devices, energy sources and power transmitting devices. For this thesis, the Toyota Prius 2010 vehicle is selected as it one most know power-split architecture. It contains an ICE, two electric machines (MGA and MGB), one high voltage battery, two planetary gear sets, and the final differential. The vehicle model is built as backward-facing and developed on MATLAB R2019b. The vehicle model is in a quasi-static manner, transmission transients are neglected, as it is necessary to solve dynamic programming.

## 4.1 Vehicle Components

This section describes the various components incorporated in the vehicle and explains their dynamics. The data for multiple components are collected from research papers, technical reports and colleagues.

### 4.1.1 Vehicle Model

The vehicle is modelled by considering its longitudinal dynamics. The lateral dynamics of the vehicle are ignored. Fig. 4.1 shows the free body diagram of the vehicle, where $v$ is the vehicle velocity, and $a$ is the acceleration of the vehicle.

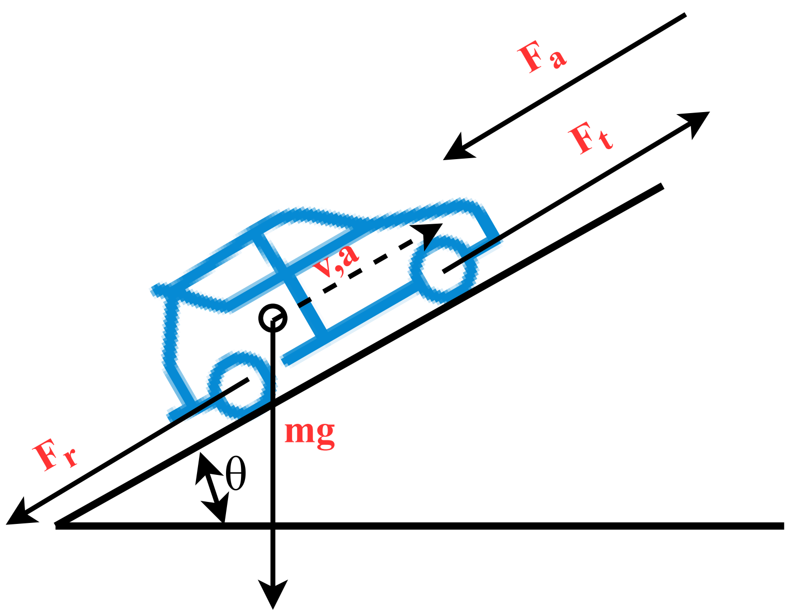


Figure 4.1: Free body diagram of longitudinal dynamics of vehicle

Here, $F_a$ is the aerodynamic resistance on the vehicle, $F_t$ is the traction force on the vehicle, $F_r$ is the rolling resistance, and $m$ is the mass of the vehicle. $g$ is the gravitational constant, and $\theta$ is the road slope. Now, according to Newtons, the second law of motion, the acceleration of the vehicle can be written as:

$$ma = F_t - F_a - F_r - mg\sin\theta \tag{4.1}$$

where $mgsin\theta$ is the force acting on the vehicle due to gravity in the longitudinal direction. The rolling resistance on the vehicle and the aerodynamic resistance is written as shown in Eq. 4.2::

$$\begin{aligned} F_r &= mg\cos\theta(\mu_1 + \mu_2 v) \\ F_a &= \frac{1}{2}\rho_{air}A_f C_d v^2 \end{aligned} \tag{4.2}$$

Here, $\mu_1$ and $\mu_2$ are the rolling resistance coefficients, $\rho_{air}$ is the density of the air, $A_f$ is the frontal area of the vehicle, and $C_d$ is the drag coefficient for the vehicle. Therefore, substituting the Eq. 4.2 in Eq. 4.1, the acceleration of the vehicle can be calculated, given that the traction force is available. As the vehicle's formulation is backward-facing, the traction force can be calculated based on Eq. 4.3 for a known drive cycle.

$$F_t = ma + \frac{1}{2}\rho_{air}A_f C_d v^2 + mg\cos\theta(\mu_1 + \mu_2 v) + mg\sin\theta \tag{4.3}$$

Once, the traction force is calculated the traction torque required on the wheels ($T_t$), and the angular velocity of the wheels can be calculated as:

$$\begin{aligned} T_t &= F_t r_{wheel} \\ w_{wheel} &= \frac{v}{r_{wheel}} \end{aligned} \tag{4.4}$$

The table shown in Tab. 4.1, shows the vehicle parameters used for modeling the vehicle.

Table 4.1: Toyota Prius 2010 Parameters

| Symbol | Parameter | Value |
|---|---|---|
| $m$ | vehicle mass | 1541.7 $kg$ |
| $r0,\ r1$ | rolling resistances | 0.002, 0.0002 $(m/s)^{-1}$ |
| $\rho_{air}$ | density of air | 1.225 $kg/m^3$ |
| $A_f$ | vehicle frontal area | 2.142 $m^3$ |
| $C_d$ | aerodynamic drag | 0.28 |
| $r_{wheel}$ | wheel radius | 0.287 $m$ |
| $g$ | acceleration due to gravity | 9.81 $m/s^2$ |

### 4.1.2 Internal Combustion Engine

As per the scope of the thesis, a quasi-static model of the ICE is used for the simulations. The Toyota Prius 2010 is a 1.8 L petrol engine with a maximum power of 73 kW and a peak torque of 142 Nm. The brake specific fuel consumption (BSFC) of the engine is shown in Fig. 4.2. It also shows the maximum torque line for the engine.

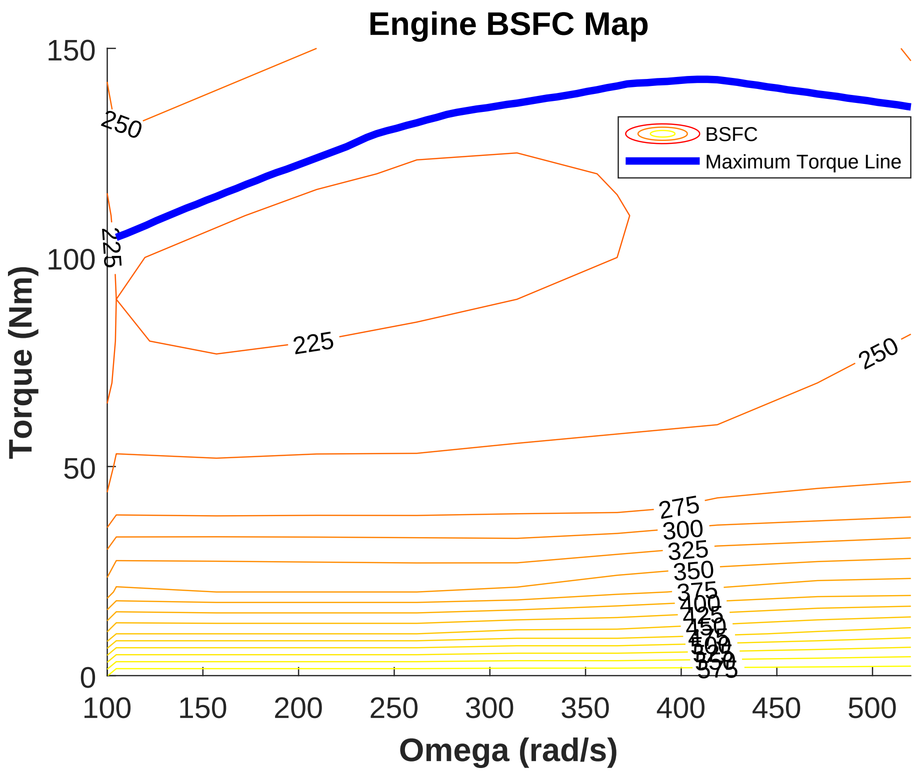


Figure 4.2: Engine BSFC Map (g/kWh) (adapted from [5])

At a given instant the BSFC of the engine is the function of its angular speed and its torque, and the mass flow rate of the fuel ($\dot{m}_f$) of the engine can be calculated as shown below:

$$\begin{aligned} BSFC &= f(\omega_{eng,T_{eng}}) \\ \dot{m}_f &= \frac{\omega_{eng} T_{eng} BSFC}{3.6 \times 10^6} \end{aligned} \tag{4.5}$$

Here the $\omega_{eng}$ is the instantaneous angular speed of the engine in rad/s, and $T_{eng}$ is the instantaneous output torque in Nm. The maximum torque line represents the maximum amount of torque the ICE can provide at a given instant. ICE's maximum torque output at a given instant is the function of its angular velocity. These are the physical limitation on the operation of the ICE and are shown in Eq. 4.6.

$$\begin{aligned} 0 \leq \omega_{eng} \leq \omega_{eng,max} \\ 0 \leq T_{eng} \leq T_{eng,max}(\omega_{eng}) \end{aligned} \tag{4.6}$$

### 4.1.3 Electric Machines

The Toyota Prius 2010 has two electric motors in its powertrain, one act as a generator motor (MGA), and the second act as an electric motor (MGB). However, it should be noted here that both electric machines can be used as generator motor or electric motor depending upon the power supplied by the ICE. Here also similar to ICE, a quasi-static model of the electric motors is used.

A generator motor (MGA) is generally used to crank start the engine, convert the power from the ICE to electrical energy, and store it in the battery. Its primary function is not to support the propelling of the vehicle, and therefore it is usually smaller in size. The MGA provides a peak power of 36 kW with a maximum torque of 46 Nm. The combined efficiency of the inverter plus MGA can be seen in Fig. 4.3. Furthermore, it also shows the maximum operating torque line for the generator motor.

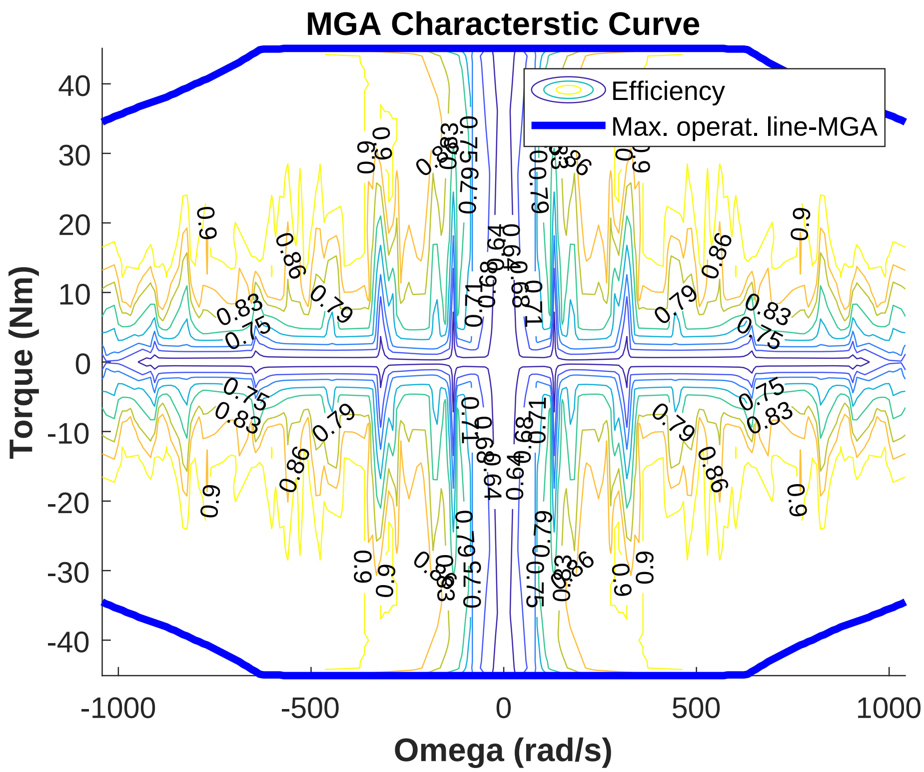


Figure 4.3: MGA Characteristic Curves (adapted from [5])

As it can be seen from Fig. 4.3, the generator efficiency is a function of output angular velocity and its output torque. Moreover, the maximum output torque is the function of its angular velocity. The motor output power ($P_{MGA}$) can be written as shown in Eq. 4.7.

$$\begin{gathered}\eta_{MGA} = f(\omega_{MGA}, T_{MGA}) \\ P_{MGA} = \omega_{MGA} T_{MGA} \eta_{MGA}^{-sgn(\omega_{MGA} T_{MGA})}\end{gathered} \tag{4.7}$$

where $\omega_{MGA}$ is the instantaneous angular speed of the MGA, $T_{MGA}$ is the instantaneous output torque of the MGA. The $sgn$ is a function that outputs +1 if the sign of $\omega_{MGA} T_{MGA}$ is positive and outputs -1 if the sign of $\omega_{MGA} T_{MGA}$ is negative.

Furthermore, due to physical limitations on MGA construction, boundary conditions are applied to its angular speed and torque, as shown in Eq. 4.8.

$$\begin{gathered}-\omega_{MGA,min} \leq \omega_{MGA} \leq \omega_{MGA,max} \\ T_{MGA,min}(\omega_{MGA}) \leq T_{MGA} \leq T_{MGA,max}(\omega_{MGA})\end{gathered} \tag{4.8}$$

Now, the primary purpose of the electric motor (MGB) is to assist the ICE in driving the vehicle. It also acts as a generator during regenerative braking and can further recharge the battery if the power provided by the ICE is greater than that of driver demand. Hence, it is necessary to keep the size of the MGB big to account for its requirements. The MGB provides a peak power of 60 kW and a maximum torque of 200 Nm. The combined efficiency map for the MGB and its inverter is shown in Fig. 4.4 along with its maximum torque line.

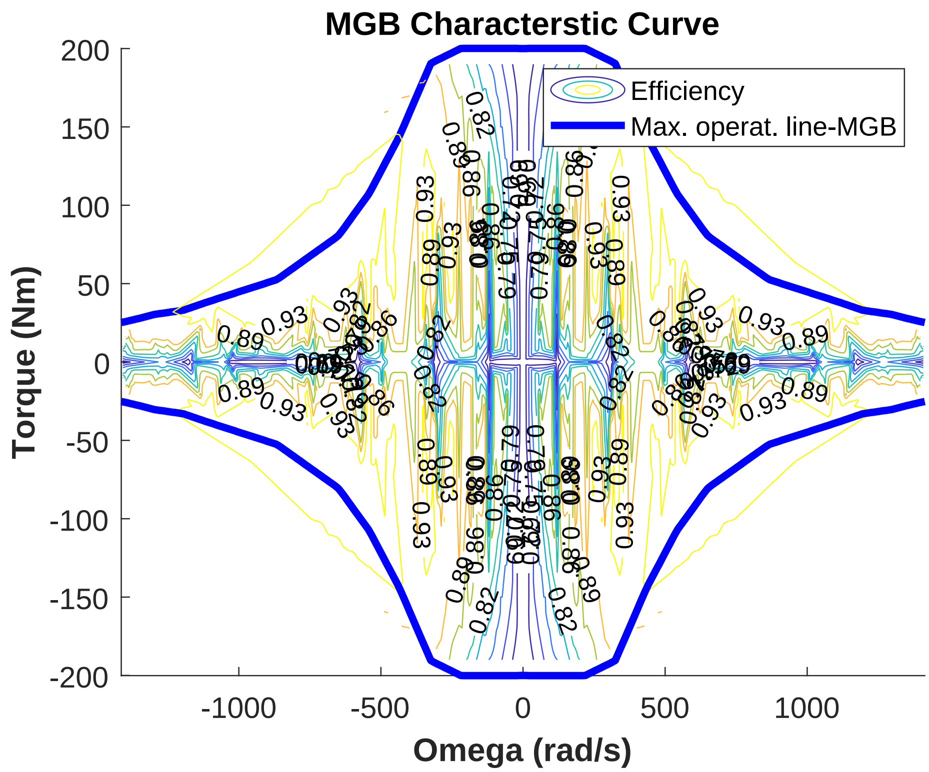


Figure 4.4: MGB Characteristic Curves (adapted from [5])

Similar to the MGA, the efficiency of the MGB is a function of its angular speed and output torque, and the maximum output torque capacity is the function of its angular velocity. The output power of the MGB ($P_{MGB}$) can be written as shown in Eq. 4.9.

$$\begin{aligned} \eta_{MGB} &= f(\omega_{MGB}, T_{MGB}) \\ P_{MGB} &= \omega_{MGB} T_{MGB} \eta_{MGB}^{-sgn(\omega_{MGB} T_{MGB})} \end{aligned} \tag{4.9}$$

where, $\omega_{MGB}$ is the instantaneous angular speed of the MGA, $T_{MGB}$ is the instantaneous output torque of the MGB, and the $sgn$ is a sign function. Furthermore, similarly due to physical limitations on the construction of MGB, boundary conditions are applied to its angular speed and torque as shown in Eq. 4.10.

$$
\begin{aligned}
-\omega_{MGB,min} \leq \omega_{MGB} \leq \omega_{MGB,max} \\
T_{MGB,min}(\omega_{MGB}) \leq T_{MGB} \leq T_{MGB,max}(\omega_{MGB})
\end{aligned}
\tag{4.10}
$$

### 4.1.4 High Voltage Battery

A high voltage battery is one of the most critical components in an HEV. A battery can decide the performance of the HEV as the fuel consumption in an HEV depends on the size of the battery, the peak acceleration can be governed by its specific power, and the weight of the vehicle is affected by its specific energy. Many factors govern the dynamics of the battery, like its construction, electrochemical composition, ambient operating temperature, cell arrangement, battery management system etc. Therefore, it is challenging and crucial to appropriately model the battery's fidelity appropriately such that its ability to show the desired behaviour. The battery contained in the Toyota Prius is a Lithium-ion battery. There are three techniques in which a Lithium-ion battery can be modelled i.e. physics-based models [93], data-driven methods [94], and equivalent-circuit (ECM) based models [95][96] .

Physics-based models are very complicated and computationally expensive modes, whereas data-driven models require much training. ECM-based models use electrical circuits active and passive components such as resistor, capacitor and impedance to emulate the battery actual behaviour [95]. They are straightforward to implement and offer good enough accuracy, hence selected for this thesis. In this thesis, a single resistance-based ECM model is used to evaluate the battery's behaviour. In this model, the temperature effect on the battery is ignored, and the battery's ageing is also ignored. The figure in Fig. 4.5 shows the equivalent circuit model for the lithium-ion battery used in this thesis.

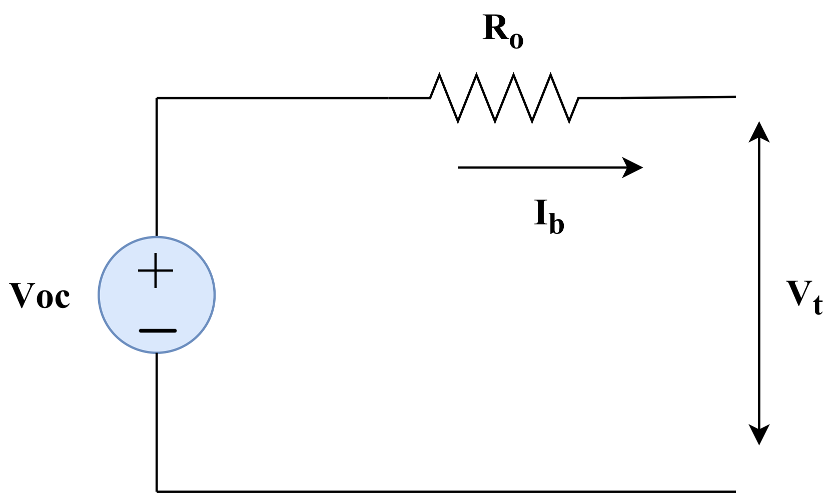


Figure 4.5: Equivalent Circuit Model for the Lithium-Ion battery

In Fig. 4.5, the $V_{oc}$ is the open-circuit voltage (OCV) of the battery, the $V_t$ is the terminal voltage of the battery, $I_b$ is the amount of current flowing through the battery, and $R_o$ is the internal resistance of the battery. The $V_{oc}$ and the $SOC$ of the battery are related to each other, such that the OCV decreases when the SOC decreases. Their relation is shown in Fig. 4.6. For the points lying outside the SOC window, the last nearest boundary OCV value is considered the OCV of that outside point.

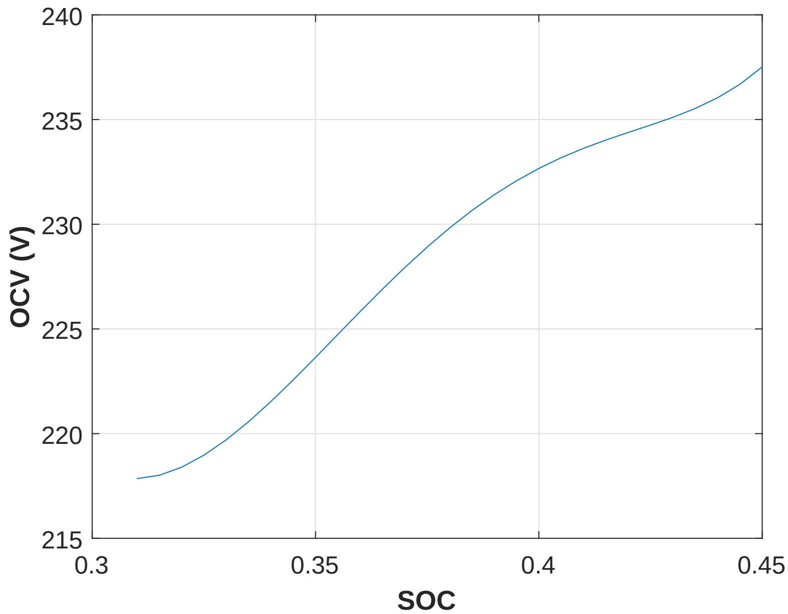


Figure 4.6: Battery open circuit voltage vs battery SOC

Furthermore, in a lithium-ion battery, the battery's internal resistance is a function of the battery SOC. The internal resistance of the battery can also differ depending upon whether the battery is charging or discharging. Therefore, Fig. 4.7 shows the battery internal resistance variation with SOC in case of charging and discharging. Moreover, the internal resistance value is taken as the nearest boundary value for the points lying outside the SOC window.

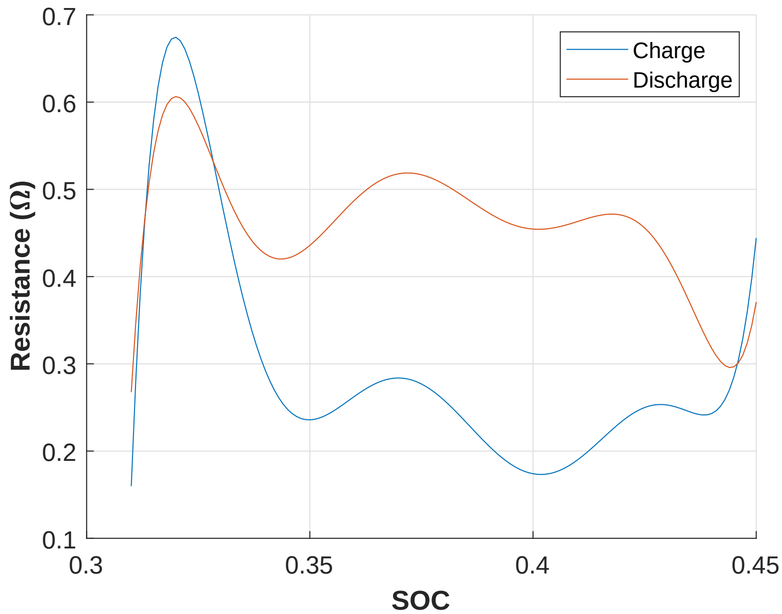


Figure 4.7: Battery internal resistance vs battery SOC

Now, to evaluate the SOC, the battery dynamics need to be understood. In a lithium-ion battery the rate of change of SOC ($S\dot{O}C$) can be written as:

$$\frac{d(SOC)}{dt} = \frac{1}{C_b}\frac{dQ}{dt} = -\frac{I_b}{C_b} \tag{4.11}$$

where $C_b$ is the battery capacity, $Q$ is the amount of charge flowing through the battery. Hence, as it can be seen from the Eq. 4.11 to evaluate the battery current $I_b$ needs to be calculated. According to Kirchoff's voltage law, the $V_{oc}$ can be written as shown in Eq. 4.12.

$$V_t = V_{oc} - I_b R_o \tag{4.12}$$

Now, as the power output of the battery can be written as product of voltage and current, i.e. $P_b = V_t I_b$, multiplying the Eq. 4.12 by $I_b$, and solving it we get:

$$\begin{aligned} V_t I_b &= V_{oc} Ib - I_b^2 R_o \\ \implies I_b &= \eta_b \frac{V_{oc} - \sqrt{V_{oc}^2 - 4R_o P_b}}{2R_o} \end{aligned} \tag{4.13}$$

where, $\eta_b$ is the battery's coulombic efficiency, which is taken as 1 if the battery power is positive and 0.9 if the battery power is negative. The battery power can be calculated by summing the power required by the MGA, MGB and the auxiliary power ($P_{aux}$). The auxiliary power is considered to be of constant 300W.

$$P_b = P_{MGA} + P_{MGB} + P_{aux} \tag{4.14}$$

At last, the lithium-ion battery has physical constraints applied due to its maximum discharge and charge capability. Also, in an HEV, a upper and lower bound is applied on the SOC so that battery aging can be minimized and unwanted situations can be avoided. These constraints are shown in Eq. 4.15:

$$\begin{aligned} P_{b,chr} \leq P_b \leq P_{b,disch} \\ SOC_{min} \leq SOC \leq SOC_{max} \end{aligned} \tag{4.15}$$

Here, $P_{b,disch}$ is the maximum discharge capacity of the battery, which is taken as 27kW, $P_{b,chr}$ is the maximum charge capacity of the battery, which is -22kW. The battery's capacity is 6.5Ah, and the maximum and minimum constraints set on the SOC in this thesis are 0.45 and 0.31, respectively.

### 4.1.5 Transmission

This subsection contains detailed information about the transmission of the Toyota Prius 2010. The Prius contains a pair of planetary gear couple to ICE, MGA, MGA and the vehicle through the differential. The incorporation of the planetary gear set makes the Prius 2010 power-split architecture. Here, the planetary gear set 1 (PGS1) ring is connected to the planetary gear set 2 (PGS2). The carrier of PGS1 is connected to the ICE and the sun of PGS1 to the MGA. The carrier of PGS2 is grounded, and its sun is connected to MGB. The ring of the PGS2 is further connected to the final differential, which is connected to the vehicle. A schematic representation of the vehicle transmission is shown in Fig. 4.8.

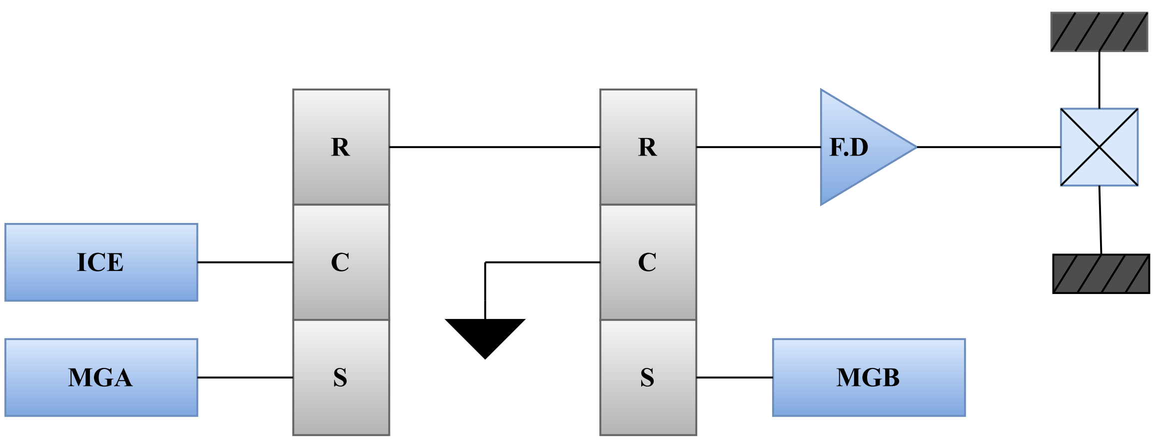


Figure 4.8: Toyota Prius 2010 transmission

The table in Tab. 4.2, shows the number of teeth in both planetary gear sets and their gear ratios, along with the final drive ratio.

The two planetary gear sets function to distribute the power keeping the ICE and the MGB working independently. For a planetary gear set, the relation between its

Table 4.2: Transmission Parameters

| | |
|---|---|
| Teeth in Sun 1 | 30 |
| Teeth in Ring 1 | 78 |
| PGS Gear Ratio | 2.60 |
| Teeth in Sun 2 | 22 |
| Teeth in Ring 2 | 58 |
| PGS Gear Ratio | 2.63 |
| Final Drive Ratio | 3.20 |

numbers of teeth, angular velocity and torques are shown in eq. 4.16.

$$\begin{gathered}\omega_s S + \omega_r R = \omega_c(S+R) \\ T_c = -\frac{S+R}{R}T_r = -\frac{S+R}{S}T_s\end{gathered} \tag{4.16}$$

$S$ is the number of teeth in the sun, $R$ is the number of teeth in the ring. $T_c$, $T_r$, $T_s$ are the torques in carrier, ring and the sun respectively. $\omega_s$, $\omega_r$, and $\omega_c$ are the angular velocities of the sun, ring and the carrier, respectively. In a PGS, the ratio of $R$ and $S$ is defined as its gear ratio. After solving the quasi-static equation for both the planetary gear sets, the angular velocity relations for the transmission can be written as shown in Eq. 4.17.

$$\begin{gathered}\omega_{fd} = \omega_{wheel}\beta_{fd} \\ \omega_{MGB} = \omega_{fd}\beta_2 \\ \omega_{MGA} = \omega_{eng}(1+\beta_1) - \beta_1\omega_{fd}\end{gathered} \tag{4.17}$$

$\beta_1$,$\beta_2$ and $\beta_{fd}$ are the gear ratios for the PGS1, PGS2 and the final differential respectively. $\omega_{wheel}$ is the wheel's angular velocity, and $\omega_{fd}$ is the differential's input angular velocity. Furthermore, the Eq. 4.18 shows the quasi-static torque relations in the transmission of the Toyota Prius 2010.

$$
\begin{aligned}
T_{fd} &= \begin{cases} \frac{T_t - T_{brk}}{\beta_{fd}\eta_{fd}} & \text{if } T_t < 0 \\ \frac{T_t}{\beta_{fd}\eta_{fd}} & \text{if } T_t \geq 0 \end{cases} \\
T_{MGA} &= -\frac{T_{eng}}{(1+\beta_1)} \\
T_{MGB} &= \frac{T_{fd} - (1+\beta_1)T_{eng}}{\beta_2}
\end{aligned} \tag{4.18}
$$

Here, $T_{fd}$ is the torque at the differential input, $\eta_{fd}$ is the final differential efficiency, $T_{brk}$ is the mechanical brake torque applied at the wheels. For this thesis, the final differential efficiency is considered to be a constant value of 0.95.

In a power-split configuration, the vehicle can operate in different modes. These modes are dependent upon the ICE operation and the direction of power flow from various components. In Toyota Prius 2010, the vehicle can operate in two modes, i.e. HEV mode and the EV mode. In HEV mode, the engine is running and supplying power to the vehicle and to the battery. Furthermore, depending upon the driver's power request, the battery can either be recharged or deplete in the HEV mode. The battery depletes if the MGB is also supplying power in assistance to the ICE, and it is greater than that of recharged by the ICE. If the ICE supplies power greater than the driver's power demand, the battery recharges in the HEV mode. In this case, the ICE is responsible for running the vehicle and also recharging the battery.

In the EV mode, the engine is not running and not supplying any power in the system. The electric machines meet the driver's power demand entirely. In this mode, the battery depletes if the driver power demand is positive, and the battery is recharged if the power demand is negative, i.e. regenerative braking is happening.

During the whole trip, the vehicle is sometimes operating in the HEV mode and sometimes in the EV mode, depending upon the engine's operation that decides the final SOC of the battery and the fuel consumption of the vehicle. Therefore, here in the energy management of the HEV, the control problem is the minimize the fuel consumption with various constraints on the system, and the control decision is the operating points of the engine.

### 4.1.6 Validation

Now, before using the vehicle model, it must be made sure that it is modelled correctly. As the vehicle model is built on MATLAB R2019b environment in a backward-facing approach, the need only exists to check for the governing equations and the data utilized to define the vehicle. Now the vehicle parameters, along with the battery data used in this thesis, were obtained from the author's colleague at McMaster Automotive Resource Center [97]. The parameters obtained were validated with the ANL experimental data in that previous study by building a Simulink-based forward-facing vehicle model. That previous study showed that the vehicle could give response very close to the experimental results. Furthermore, the efficiency and the BSFC maps used in this thesis, as mentioned, were obtained from a research paper [5].

An energy balance approach was taken to check that the governing equations and the vehicle model are accurately developed. A DP-based EMS was utilized on a particular drivecycle to evaluate operating points and efficiencies of all the components. Table. 4.3 shows the energy balance results, where the losses for every component are mentioned, along with the total energy supplied by the ICE and battery.

The above table shows that the actual energy supplied by the ICE and the battery

Table 4.3: Energy Balance for Toyota Prius 2010

| | Energy (kJ) |
|---|---|
| MGA Losses | -354 |
| MGB Losses | -929 |
| Auxillory Losses | -324 |
| Brake Losses | 0 |
| Road Load Losses | -7101 |
| Transmission Losses | -373 |
| Battery Losses | -434 |
| ICE Energy | 9497 |
| Battery Energy | 28 |
| Energy Difference | 10 |

is very close to the total amount of losses in the system. The energy difference of 10 kJ comes due to the integration of precision error at every instant. Noticing that the energy does balance for the system, it can be said that the governing equation for the system's dynamics is correctly modelled.

## 4.2 Optimal Control Problem

This section contains a detailed explanation of the optimal control problem and its formulation for the eco-driving based EMS in CA-HEV's. In eco-driving based EMS, the ultimate objective of the EMS is to minimize the fuel consumption of the vehicle for a given trip. In a conventional hybrid electric vehicle, this is achieved by optimizing the power distribution between energy sources to utilize minimum fuel. However, as mentioned, CA-HEV's offers the capability to incorporate external environment information, which offers an extra degree of freedom in the system to look for optimization techniques.

Eco-driving, in general, means driving in a manner to minimize energy consumption. Therefore, naturally, eco-driving based EMS also incorporates optimizing the vehicle's velocity profile along with its power distribution. Hence, the optimal control problem of the eco-driving based EMS has been divided into two problems: velocity optimization and power distribution optimization (Powertrain Optimization). The following two subsections provide the optimal control formulation for both objectives.

### 4.2.1 Optimal Velocity Problem

In an optimal velocity problem, the objective is to find an optimal velocity trajectory of the trip such that it can exploit the maximum potential of the HEV and result in an overall fuel-efficient path. It should not sacrifice the driver's comfort, mobility and safety. Determining a velocity trajectory is critical as it influences the HEV's engine operating points and determines how much fuel consumption can be improved [84].

To solve the problem, the control problem needs to be defined. The optimal solution depends upon the formulation of the cost function. In this thesis, the objective is to minimize the amount of total power required to finish the given trip. Considering, from an energy point of perspective, it can understand that the amount of energy required to drive the vehicle from starting to the endpoint will come from the system itself. The only external source of energy in the system is fuel's chemical energy, as the battery needs to remain in charge sustenance at the end of the trip. Therefore, minimizing the amount of total power required to complete the trip will essentially result in a fuel-efficient trajectory. Now, as the solution to minimum power consumption would result in a trivial solution (zero power consumption), there needs to be a penalty term introduce into the cost function. The trivial solution leads the vehicle to be stationary or accelerate and brake at the same frequency and amplitude, resulting in an infinite or large total trip time, respectively. Hence, to eliminate this undesired effect, a penalty term corresponding to the total time required to complete the trip is added in the cost function. The optimal control problem for velocity in the discretized domain is shown in Eq. 4.19.

$$\begin{aligned} J &= \frac{\alpha F_t v}{\beta} + (1-\alpha)\frac{T_{travel}}{\gamma} \\ v_{opt} &= arg\ \min_{a}\ J \end{aligned} \tag{4.19}$$

Here, $T_{travel}$ is the total travel time of the trip, $J$ is the cost function that needs to be minimized, and the control action is $a$, i.e. acceleration of the vehicle. $v$, the vehicle velocity is the state variable that is the outcome of the optimization.$\alpha$ is a weighting factor that ranges between zero and one. $beta$ is the normalizing factor for the traction force that makes it between zero and one, i.e. $beta = max(F_t)$. Furthermore, here $\gamma$ is also a normalizing factor, i.e. $\gamma = max(T_{travel})$. To completely define the control problem, the governing equation for the problem needs to be defined along with the state and control variable constraints. The governing equation of the system is shown in Eq. 4.20, where the relation between the state and control variable is shown. Furthermore, the global constraints on the state variable $v$ enforce it to be as zero at the start and end of the problem horizon as the vehicle starts from the rest position and ends at rest.

$$
\begin{aligned}
&\dot{v} = f(v, a, t) \\
&v(0) = v(D_N) = 0 \\
&0 \leq v \leq V_{max} \\
&a_{min}(v) \leq a \leq a_{max}(v)
\end{aligned}
\tag{4.20}
$$

$V_{max}$ is the maximum velocity of the vehicle at any given instant, $a_{min}$ and $a_{max}$ are the minimum and maximum acceleration of the vehicle at a given state, respectively. $N$ is the endpoint of the horizon, and $f(v, a, t)$ is the function determining the state dynamics. The method to solve this optimal control problem is discussed in the next chapter in detail.

### 4.2.2 Optimal Power-distribution Problem

The power distribution (powertrain) optimal problem is to find the optimal distribution of energy at every instant from the different energy sources. Here, it takes the input as optimal velocity and provides the optimal power for ICE, MGA and MGB at each instant such that the constraints on the system are met. This optimal problem is basically the EM problem in a conventional HEV vehicle.

Here, this optimal control problem aims to minimize the fuel consumption of the trip and maintain the end SOC value under a limit. Therefore, the cost function can be directly written as the fuel consumption of the trip. Due to the quasi-static modelling of the ICE and the transmission, the fuel required to crank up the engine is neglected. This leads to a frequent start-stop of the engine, which is undesirable considering the driver's perception. Hence, a penalty term related to engine cranking is also added in the cost function to minimize the mode changes. The cost function in the discretized domain is shown in Eq. 4.21.

$$\begin{aligned} J &= \dot{m}_f(\omega_{eng}, T_{eng}) + \psi E_{start} \\ SOC_{opt}, Mode_{opt} &= arg \min_{\omega_{eng}, T_{eng}, Mode} J \end{aligned} \tag{4.21}$$

Here, $J$ is the cost function, $E_{start}$ is the engine start penalty term which is 1 during engine cranking and 0 otherwise. $\psi$ is the constant weighing factor, $SOC_{opti}$ and $Mode_{opti}$ are the output optimal state variables representing the optimal SOC and optimal mode profile for the vehicle for the particular trip. As it can be seen from the Eq. 4.21 the ICE's angular speed and its torque are the control variables for the optimal problem.

To completely define the system, it is imperative to realize the system's constraints and its dynamics. As the Toyota Prius is a full HEV, the battery's initial and end SOC must remain at the same value. This provides the two-point boundary constraint in the system's state, as shown in Eq. 4.22. Furthermore, it can only vary in a certain range to avoid battery aging and undesirable effects. Due to the presence of physical limitation on the actuator components, there are bounds to the control variables in the system, also shown in Eq. 4.22.

$$\begin{gathered} SOC(0) = SOC(N) = SOC_{init} \\ SOC_{min}^{lim} \leq SOC \leq SOC_{max}^{lim} \\ 0 \leq \omega_{eng} \leq \omega_{eng,max} \\ 0 \leq T_{eng} \leq T_{eng,max}(\omega_{eng}) \end{gathered} \tag{4.22}$$

$SOC_{init}$ is the initial SOC, $N$ is the end point in the optimization horizon, $SOC_{min}^{lim}$

and $SOC_{max}^{lim}$ are the SOC's lower and upper bounds respectively, which needs to be satisfied at every instant. The system's governing equation can be written as:

$$S\dot{O}C = f(SOC, \omega_{eng}, T_{eng}, t) \tag{4.23}$$

where, $f(SOC, \omega_{eng}, T_{eng}, t)$ is a function dependent on $SOC$, $\omega_{eng}$, $T_{eng}$ and $t$. This function can be evaluated using the Eq. 4.11. The detailed method to solve this optimal control problem is explained in detail in next chapter.

# Chapter 5

# Intelligent Eco-driving Based EMS for Connected and Autonomous Hybrid Electric Vehicles

This chapter contains detailed information on the developed novel intelligent eco-driving based energy management strategy for connected and autonomous hybrid electric vehicles. The complete formulation of the algorithm is provided in the upcoming sections.

## 5.1 Introduction

Eco-driving based EMS for the CA-HEV's have become a hot topic in recent years due to development in communication and electronics technology which makes CA-HEV's to be practicable in the first place. As mentioned, the CAV's offers the capability to incorporate external information into the algorithms, therefore making them adaptive to changes. The V2V and V2I communications and the cloud server technology play an essential role in enabling the CA-HEV's to perform.

In the case of eco-driving, the V2I paves the data path, providing information on traffic lights, speed limits, road impedance through roadside units. The cloud server utilizes GPS and GIS technology to communicate with vehicle embedded controllers about the road slope, trip route, traffic congestion and vehicle state information. Combining the data from various sources, an intelligent eco-driving based EMS is formulated for the CA-HEV's.

As discussed in chapter 3, under the eco-driving based EMS section, the eco-driving-based EMS formulation common approach is to develop a bi-layer structure. The bi-layer structure offers multiple advantages by decoupling the problem into sub-problems and effectively optimizing them. A similar approach is taken in this thesis, where a bi-layer optimization technique is developed to solve optimal velocity and the optimal power distribution trajectories. The purpose of choosing the bi-layer structure is to reduce the algorithm's computation complexity as both the optimal control problems are solved using the DP algorithm. These optimal solutions are evaluated in the upper level controller. Due to the computation cost of the DP, the upper level controller can be solved on the cloud server. Now, as it is known that the DP provides the global optimal results and its acausal in nature, its real-time implementation is not possible. Hence, in this thesis, the DP results are only used as the benchmark for the lower level controller embedded in the vehicle responsible for energy management in real-time. This is very different from existing literature, which either uses sub-optimal strategies in bi-layer fashion for real-time implementation or utilizes the DP in combination with predictive strategies to make it real-time implementable and hence both providing far-from global optimal results [78][79][87] [88][90][91][92].

The lower level controller is responsible for the energy management of the vehicle in real-time. This lower level controller is based on the RL-based EMS, and hence it provides real-time ability. The RL controller utilizes the Q-Learning algorithm, which is a computationally cheap algorithm. As RL is based on data, it requires training based on the results evaluated from the upper level controller using DP. Therefore, the RL is trained on the global optimal results for the optimal velocity problem and provides near-optimal solutions for the EMS of the vehicle. The upper level controller's DP provides the global optimal power distribution solution for the evaluated global optimal velocity profile. However, due to external environmental disturbances or the driver's driving behaviour, the real-world velocity of the vehicle might differ from the optimal one. This is where the RL-based EMS advantage comes into the picture as it can adjust to the real-world drivecycle and still provide near-optimal results for power distribution. Thereby making the lower level controller robust and intelligent in the sense that it can respond well to deviations from the optimal profile.

An overview of the implementation of the complete algorithm, including the upper and the lower level controller, can be seen in Fig. 5.1. As shown, the DP algorithm takes input parameters for the CA-HEV along with the road data, trip data and vehicle data. It provides an optimal reference velocity and SOC trajectory that tunes the Q-learning controller parameters based on the training. The Q-learning based EMS controller then manages the power distribution on the vehicle in real-time. This accounts for the novelty of the algorithm, where the RL-based EMS is trained from the ideal optimal DP results and is then utilized for the online eco-driving based EMS of the CA-HEV in real-world scenarios. The next sections discuss the implementation of the DP algorithms in the upper level controller and the RL algorithm in the lower level controller.

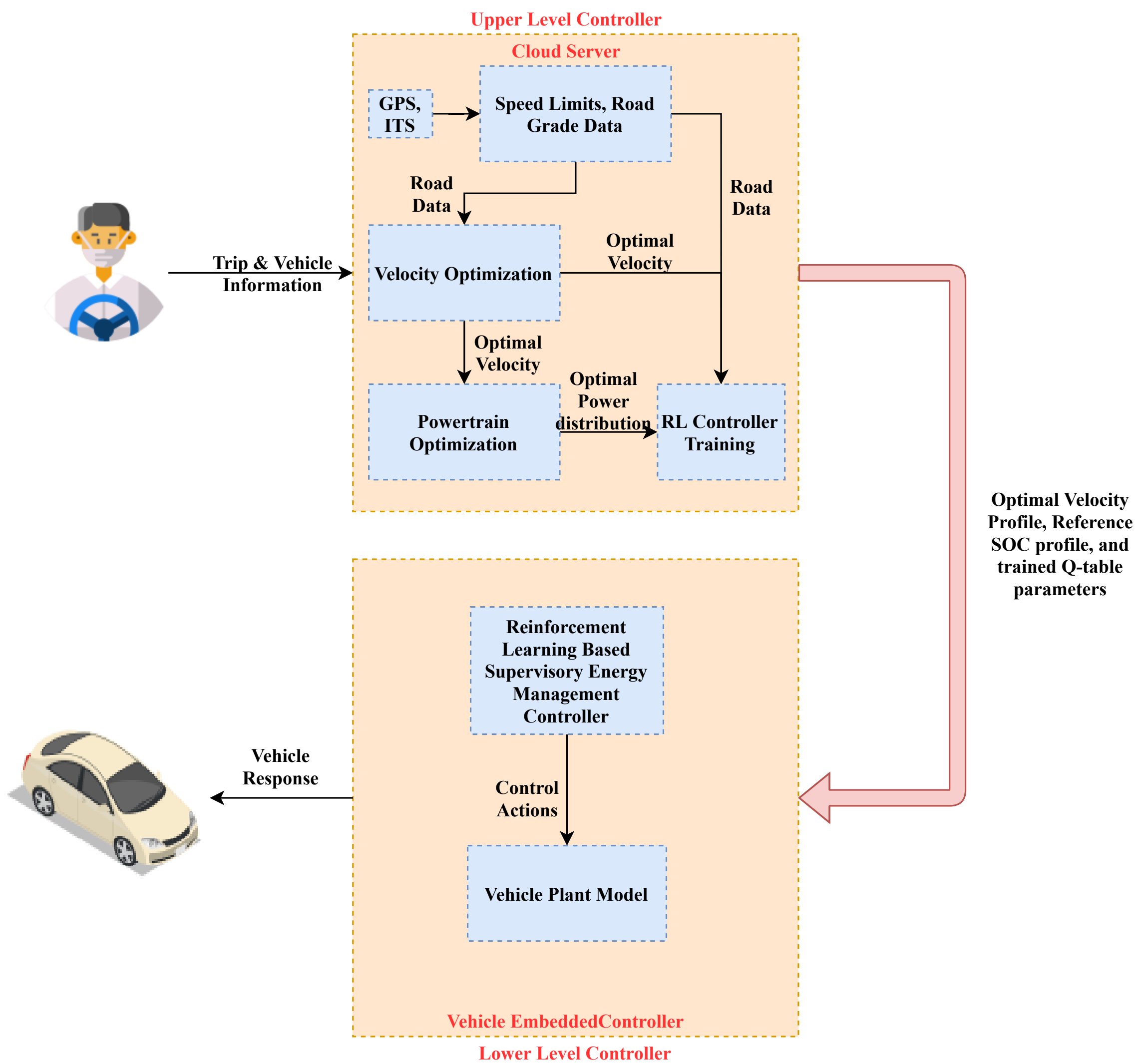


Figure 5.1: Overview of novel Eco-driving based EMS

## 5.2 Upper Level Controller

This section provides a detailed description of the algorithm's upper level controller, which is responsible for optimizing the velocity profile for a given trip and generating a reference SOC trajectory by optimizing power distribution for the lower level controller. As mentioned, the upper level controller can be solved on the cloud server and the RL controller's training. The upper level controller consists of the formulation of the DP algorithm for optimizing the velocity and the power distribution of the vehicle. The upper level controller takes the trip starting and ending point information from the driver and generates the velocity upper limit for the entire trip depending upon the speed limits for the route. For the current simulation work in this paper, the speed limit information is extracted using the Simulation in Urban Mobility (SUMO) software combined with OpenStreetMap. The road grade information is also provided as an input to the controller, which is taken as a constant value of zero due to the current scope of the research. All this information is fed into the velocity optimization algorithm and the powertrain optimization algorithm to be used as input parameters for the algorithm. The sub-sections below explain the detailed formulation of both the velocity and power distribution optimization techniques.

### 5.2.1 Velocity Optimization

The very first step in the eco-driving based EMS formulation is to determine the optimal velocity trajectory for the trip. Several techniques are available in the literature to solve the problem of velocity optimization, like the use of PMP with the use of V2V and V2I information [92]. In [98], authors formulate a cost function based on the dynamic power losses of the powertrain and minimize it using Sequential Quadratic

Programming (SQP) to find an optimal velocity trajectory. Authors in [99] utilized a DP approach in addition to multiple predictive algorithms in sub-problems to tackle the problem of velocity optimization and powertrain optimization using information V2V, V2I and vehicle advanced driver assistance system (ADAS). In this thesis also as mentioned, a DP approach is taken to estimate the optimal velocity profile for the given trip.

The control problem for the velocity optimization is shown in chapter 4 under the control problem section. Where the cost function for the velocity optimization is shown in Eq. 4.19 and the constraints to the problem in Eq. 4.20. Generally, the road grade data collected along with speed limits for the trip are evaluated with respect to distance, as time-domain does not affect them. Therefore, the DP is solved in the spatial domain here instead of the common time-domain approach.

To solve the problem using the DP, the state-space needs to be discretized, and the system's feasibility also needs to be defined. In this thesis, the velocity is discretized from 0 to 35 $m/s$ by 73 equally spaced points. The acceleration is divided from -3 $m/s^2$ to 2 $m/s^2$ by 51 equally spaced points. The governing equation in the case of velocity optimization in the spatial domain can be expanded as shown in Eq.5.1.

$$
\begin{aligned}
\dot{v} &= \frac{dv}{dt} \\
\implies \dot{v} &= \frac{dv}{dD}\frac{dD}{dt} \\
\dot{v} &= v\frac{dv}{dD} \\
\text{on Integrating} \quad \dot{v}\int_{D_n}^{D_{n+1}} dD &= \int_{v_n}^{v_{n+1}} v dv \\
&\text{assuming } \dot{v} \text{ constant from } D_n \text{ to } D_{n+1} \\
\implies 2a(D_{n+1} - D_n) &= v_{n+1}^2 - v_n^2
\end{aligned}
\tag{5.1}
$$

where $D$ is the distance covered by the vehicle and $D_n$ represents the distance at the $n$th step. Now, from the cost function shown in Eq. 4.19, $F_t$ and $T_{travel}$ still needs to evaluated. For finding the traction force required to drive the vehicle, its longitudinal dynamics are considered, and lateral dynamics are ignored. The Eq. 4.2 can be referred along with the vehicle components section to find the traction force in the vehicle. The $T_{travel}$ can be evaluated by integrating the time required to transition from $D_n$ ti $D_{n+1}$ for all $n$. It can be evaluated based on the equation shown below for one particular transition:

$$
\begin{aligned}
\dot{v} &= \frac{dv}{dt} \\
\int_0^{t_n} \dot{v} dt &= \int_{v_0}^{v_n} dv \implies t_n = \frac{(v_n - v_0)}{a_0}
\end{aligned}
\tag{5.2}
$$

where $t_n$ is the time required to travel from $D_0$ to $D_n$ meters and $a_0$ is the constant acceleration during that time. Furthermore, all the physical feasibility of the system must also be defined to find a valid solution. The physical limitation on the vehicle comes due to the maximum torque that the individual components can provide. Lets suppose, the vehicle is running at $v_{D_n}$ speed at $D_n$ distance from trip start. At the particular speed, the maximum traction torque on the wheels is $T_{t_{max}}(v_{D_n})$. This maximum torque can be achieved when the vehicle operates in mode 0 (i.e. engine on mode). For all engine operating speed points to be feasible, the maximum torque that the engine can supply will be the minimum value on the WOT line of the engine ( $T_{eng,min}$). Therefore,

$$\begin{aligned} \omega_{MGB}(D_n) &= v_{D_n} r_{wheel} \beta_{fd} \beta_2 \\ T_{MG2}(D_n) &= \frac{T_{t,max}(v_{D_n}) - (1+\beta_1) T_{eng,min}}{\beta_2} \\ \omega_{fd} &= W_{wheels} \beta_{fd} \end{aligned} \tag{5.3}$$

Now, if the motor 2 torque calculated in Eq. 5.3, is more than the maximum possible torque capacity of the the motor at that given velocity, the point becomes infeasible. Hence, there is an upper constraint on the traction torque provided at a given instant which is a function of engine operating point and the vehicle velocity, as shown in Eq. 5.4.

$$T_{t,max} = f(v, \omega_{eng}, T_{eng}) \tag{5.4}$$

The DP algorithm is built on the MATLAB R2019b platform, using the structure developed by authors in [100]. The results of the velocity optimization algorithm is then further utilized by the powertrain optimization algorithm, which is explained in the following subsection.

### 5.2.2 Powertrain Optimization

The objective of the powertrain optimization layer is to take the optimal velocity as the input and provide optimal power distribution and SOC trajectory as an output. The DP is also used here as an optimization-based EMS for finding the global optimal solution. The cost function for the powertrain optimization is shown in Eq. 4.21 along with the constraints on the system shown in Eq. 4.22. As the optimal velocity was found in the spatial domain, the optimal power distribution is also solved in the spatial domain. The control variables and the state variables for the DP are shown in Eq. 5.5.

$$x = \begin{vmatrix} SOC \\ Mode \end{vmatrix} \qquad u = \begin{vmatrix} W_{ICE} \\ T_{ICE} \\ Mode \end{vmatrix} \tag{5.5}$$

To keep the computational cost less, the $SOC$ is discretized into 50 equal parts ranging between 0.31 to 0.45. $Mode$ is discretized into 1 or 0 depending upon the engine status. $\omega_{eng}$ and $T_{eng}$ are both discretized into 9 parts from 0 to 5000 rps and 0 to 142 Nm, respectively. To define the system, the governing state equations shown in Eq. 4.23, must be solved, which can be done by using the equation shown in Eq. 4.11.The vehicle's mode depends upon the status of the engine and can be calculated as shown in Eq. 5.6.

$$Mode(D) = \begin{cases} 1 & \text{if } \ \omega_{eng} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Furthermore, the only term remaining to find the cost at a given step completely is the fuel consumption of the vehicle ($\dot{m}_f$), which can be evaluated based on ICE's operation points as shown in Eq. 4.5. The next set of constraints apart from that shown in Eq. 4.22 on the system comes from the physical limitation of the actuators present. There are three actuators present, each with its maximum power capacities depending on the maximum torque transfer at a given instant by the component. This can be written as shown in Eq. 5.7.

$$\begin{aligned} 0 \leq T_{MGA} \leq T_{MGA,max}(\omega_{MGA}) \\ 0 \leq T_{MGB} \leq T_{MGB,max}(\omega_{MGB}) \end{aligned} \tag{5.7}$$

Here, $T_{MGA,max}(\omega_{MGA})$ and $T_{MGB,max}(\omega_{MGB})$ are the maximum torque that can be supplied by electric motor A and B respectively at their respective angular velocities. The next limiting factor that has physical constraints is the high voltage battery of the system. Toyota Prius 2010 uses a 6.5 Ah Li-ion battery with maximum power discharge and charge capability of 27 kW and 22kW, respectively. Therefore, battery power constraint can be written shown in Eq. 4.15.

The powertrain optimization layer in the upper-level controller is also built on MATLAB R2019b using the DPM function structure developed by [100]. This sub-problem finds the optimal power distribution between the engine and the electric motors and provides the optimal SOC reference trajectory for the trip in the ideal environment. These optimal results are then used in the cloud server to train the RL agent and act as the RL algorithm's benchmark solution. The following subsection describes the training of the RL agent, which is used in the lower level controller.

### 5.2.3 Reinforcement Learning (RL) Based Controller Training

This section describes the process of training the RL controller, which can be conducted on the cloud server due to its processing power. The tuned controller parameters can then be communicated with the vehicle embedded controller for its real-time application. As it was mentioned, the RL controller is based on the Q-Learning algorithm, whose fundamentals were shown under the learning-based strategies subsection. In the Q-learning algorithm, the controller decides based on the action-value function of the state ($Q(s, a)$). Therefore, it is imperative to find a correct estimate of the action-value function for all the combinations of state-action pairs. Hence, the objective of training the RL algorithm comes down to find as close as a possible accurate estimate of the action-value functions for all state-space.

To begin the formulation of the Q-learning algorithm, it is necessary to define the state and action space for the optimal control problem of energy management in CA-HEV. The state of the system must be defined such that it should be able to completely define the whole system irrespective of the previous state and action required to get there, meaning it should have the Markov property. In this thesis, the battery state-of-charge ($SOC$), vehicle velocity ($v$), vehicle power demand at the final drive ($P_{drive}$) and the engine state ($E_{state}$) are selected as the state variables. These four parameters combine to form a four-dimensional state space, i.e., $s = (SOC, v, P_{drive}, E_{state})^T$, where each of them is the function of optimization domain, which in this case is the distance covered by the vehicle.

Now, as the problem of energy management in CA-HEV's revolve around the power distribution between the ICE and the electric machines, it is logical to choose the action space that can define this distribution. Therefore, here as similar to the DP, the engine speed ($\omega_{eng}$) and the engine output torque ($T_{eng}$) are chosen as the instantaneous action variables. Hence, the action space can be written as: $a = (\omega_{eng}, T_{eng})$, and depending upon the input action variables, the angular speed and torques of the electric machines can be calculated for a given state.

The next step in the formulation of the Q-learning controller is to define the immediate reward for the system ($r_D$). The immediate reward is a crucial factor as it significantly influences the algorithm's convergence and output performance. The immediate reward ($r^a_{s,s^+}$) represents the cost of transitioning from the current state $s$ to the next state $s^+$ with the control actions $a$. Therefore, the immediate reward should be such that it can encompass the objective of the optimization. Hence, in this thesis, the immediate reward is chosen, as shown in the Eq. 5.8. The SOC penalty term is introduced to keep the battery in charge sustenance in addition to the fuel consumption of the vehicle. Furthermore, during the regenerative braking, the engine ensures to be in *off* state.

$$r^a_{s,s^+} = \begin{cases} \dot{m}_f + C_p + \alpha\Delta E_{state} & \text{if } \; P_b \geq 0 \cap \; SOC \leq 0.31 \text{ or } SOC \geq 0.45 \\ \dot{m}_f + |SOC - SOC_{init}| + \alpha\Delta E_{state} & \text{if } \; P_b \geq 0 \cap \; 0.31 \leq SOC \leq 0.45 \\ \dot{m}_f & \text{if } \; P_b \leq 0 \cap \omega_{eng} = 0 \cap T_{eng} = 0 \\ C_{inf} & \text{otherwise} \end{cases} \tag{5.8}$$

Here, $\dot{m}_f$ is the instantaneous fuel consumption of the ICE, $C_p$ is a constant penalty term, $\alpha$ is a constant weighing factor. incorporation of the $C_p$ ensures that the battery always operates within the feasible operating zone. $\Delta E_{state}$ is the term defined to penalize the cranking of the engine, where $\Delta E_{state} = 1$ only when the engine is cranked; otherwise, it is zero. $SOC_{init}$ is the starting value of the SOC of the battery when the trip begins, and $|SOC - SOC_{init}|$ is the absolute difference between the current SOC and the initial SOC. $C_{inf}$ is a big constant value used to define the reward for the infeasible points. The infeasibility in the system comes from either violating the constraints defined in Eq. 4.22 and in the Eq. 5.7. The objective of the Q-learning implemented in this thesis is to minimize the cumulative reward of the episode. Moreover, with this formulation of the reward function, the fuel consumption can be minimized and meet certain constraints on the SOC value.

As for the finding the optimal control policy ($\pi^*$), mapping the state $\mathbb{S}$ and the control action $\mathbb{A}$, the action-value function $\mathbb{Q}^*(\mathbb{S}, \mathbb{A}) : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{Q}$, needs to be evaluated. The optimal control policy can be written as shown in Eq. 5.9.

$$\pi^*(s) = arg \ \underset{a}{min} Q^*(s,a) \tag{5.9}$$

To evaluate the $Q^*(s,a)$ for all state-action pairs, the equation shown in Eq. 3.14 needs to be solved, which can be done on the basis of Bellman's equation as shown in Eq. 5.10.

$$Q^*(s,a) = \mathbb{E}[r + \gamma \underset{a}{min} Q^*(s+, a+)|s,a] \tag{5.10}$$

where $s+$ represents the next state of the system reached after taking action $a$ at the state $s$. $a+$ is the optimal action taken by the state $s+$, and $\gamma$ is the discount factor such that $\gamma \in (0,1)$ is used for assuring the convergence of the cost. Now, the

$Q^*(s, a)$ would give the optimal value of the action-value given that optimal action-value at the next state, i.e. $Q^*(s+, a+)$ is known accurately beforehand, which is not the case. Therefore, the $Q(s, a)$ value needs to be evaluated by training the Q-learning algorithm to reach its correct estimate. The equation to update the estimated value of the action-value function in the spatial domain is shown in Eq. 5.11.

$$Q_D(s_D, a_D) = Q_D(s_D, a_D) + \alpha \left\{ r_t + \gamma \min_a Q_D(s_{D+1}, a) - Q_D(s_D, a_D) \right\} \tag{5.11}$$

where $\zeta$ is the learning rate such that $\alpha \in (0, 1]$. The update mechanism of $Q(s_D, a_D)$ is explained in Fig. 5.2. The environment includes all the driving scenarios, the powertrain dynamics, and vehicle dynamics. The RL agent or decision-maker is just the EMS controller that decides the action to be taken at a particular state. Suppose the environment is at state $s_{D-1}$ at the current step $(D-1)$. Now, depending upon the current state, the Q-learning algorithm will choose an action $a_{D-1}$ from all possible action space following an $\epsilon$-greedy policy. In $\epsilon$=greedy policy, the controller chose the action with minimum Q-value with a probability of $1 - \epsilon$, and chose a random feasible action with a probability of $\epsilon$.

Once, the action $a_{D-1}$ is chosen, the environment react to the action and goes to the next state $s_D$ while getting a reward $r_{D-1}$ for transitioning from $s_{D-1}$ to $s_D$. Now, at the new state $s_D$, the controller again chose an action, but this time selecting the action $a_D$, which gives the minimum Q-value. Now, the Q-value of the current state $Q(s_D, a_D)$ multiplied by $\gamma$ is added to the reward $r_{D-1}$. This sum is then compared to the Q-value of the previous state, i.e., $Q(s_{D-1}, a_{D-1})$ to see if their difference is smaller than a constant small real number $\epsilon$ and if the maximum number of iterations has reached or not. If both conditions are satisfied, the Q-value of the state-action pair $Q(s_{D-1}, a_{D-1})$ is updated by the equation shown in Eq. 5.11. This completes a one-step for $k^{th}$ iteration, and the loop is repeated for all complete episodes for the maximum number of iterations or convergence, whichever is reached first. Therefore, at the end of the complete training, an accurate estimate of the Q-table has been formulated, providing an action-value function for every state-action pairs.

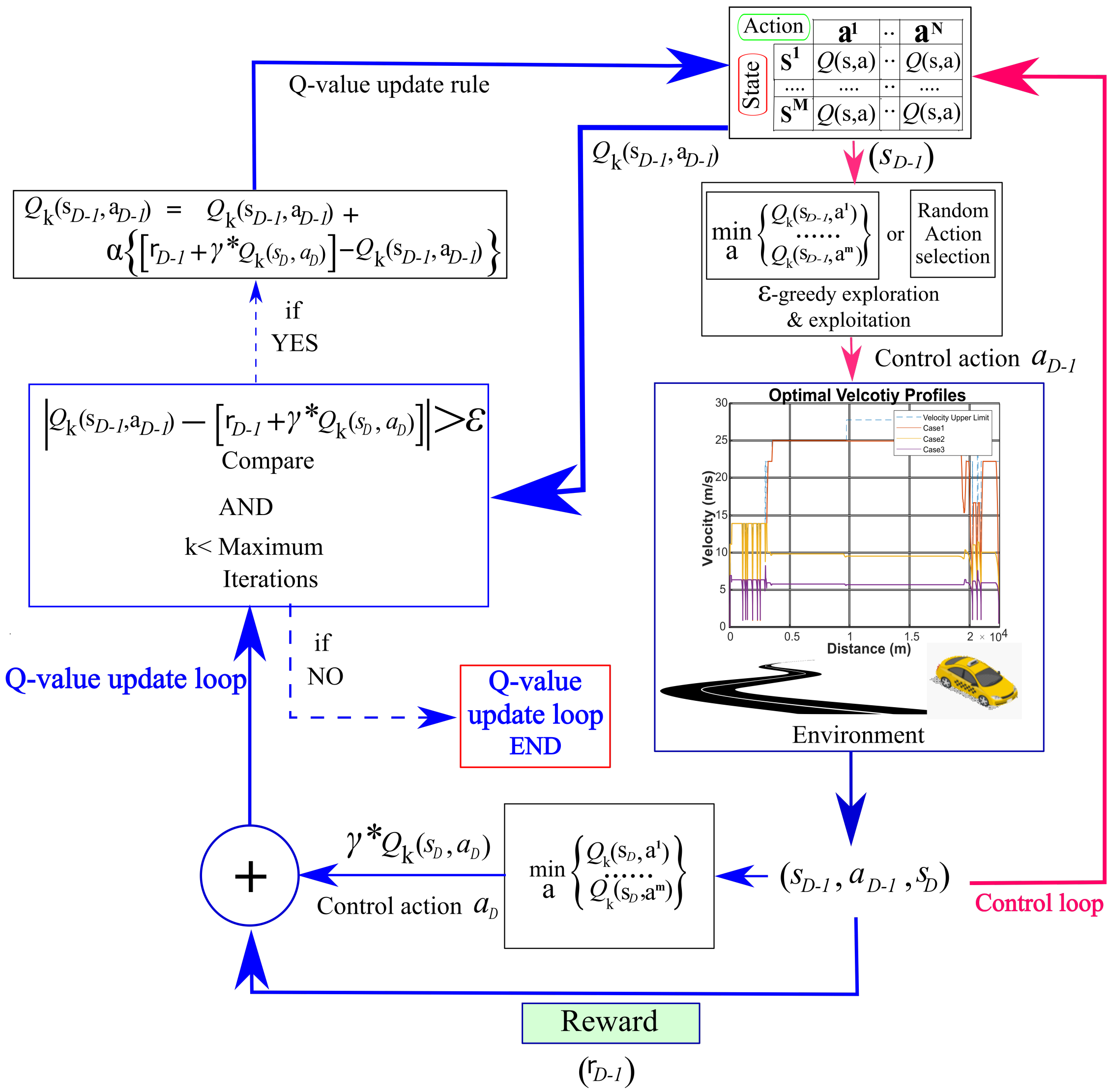


Figure 5.2: Schematic of Q-value function update

If the $Q(s, a)$ for all state-action pairs is correctly estimated, the following relation between $Q(s_{D-1}, a_{D-1})$ and $Q(s_D, a_D)$ given in Eq. 5.12 holds true implying that the convergence was reached.

$$
\begin{aligned}
Q^{\pi}(s_{D-1}, a_{D-1}) = r_{D-1} + & \\
& \mathbb{E}_{\pi}\left\{\sum_{k=1}^{\infty} \gamma^{k} r_{D-1+k} | s_{D-1} = s, a_{D-1} = a\right\} \\
& = r_{D-1} + \gamma Q^{\pi}(s_D, a_D)
\end{aligned}
\tag{5.12}
$$

## 5.3 Lower Level Controller

The lower level controller is responsible for the energy management of the CA-HEV in real-time. The lower level controller optimal control problem formulation is precisely the same as the optimal power distribution problem. However, unlike DP in the upper level controller, here it is solved by the RL agent based on Q-learning. The training of the RL controller can be done on the cloud server from the results obtained by the DP algorithm. Once the training is complete, an accurate estimate of the Q-table is generated, which then can be sent to the vehicle's embedded controller. These training results are dependent upon the vehicle type, road speed limits, road topography, trip route, and independent of the traffic. It can also be stored in the repository to be used for future references.

Once the vehicle embedded controller receives the estimated Q-table for the given trip, the Q-learning algorithm built on the controller is then responsible for optimal power distribution. As described in the RL controller training subsection, the state variables for the algorithm are the battery state-of-charge ($SOC$), vehicle velocity ($v$), vehicle power demand at the final drive ($P_{drive}$) and the engine state ($E_{state}$). The control actions are the engine speed ($\omega_{eng}$) and the engine output torque ($T_{eng}$).

As for the Q-leaning algorithm, discretization of the problem is a must condition. The state and the action variables are discretized in the same manner as done during training because of the size of the Q-table.

Now, during the vehicle's operation in real-time, the controller takes the current state variables of the system as the input and then decides the action based on the minimum Q-value for that particular state. This decides the action at the current step, which drives the system into the next step. Choosing infeasible points does not arise as the infeasible points were rewarded at a very high cost during the training. Here, with the same reward function as shown in Eq. 5.8, and the with the correct Q-values, the fuel consumption of the vehicle is minimized, meeting the constraints. The pseudo-code for the Q-learning algorithm implemented in the embedded vehicle powertrain controller is shown below.

**Algorithm 1:** Q-Learning Algorithm

Load the trained Q-table;

Define state variables discritization;

Define action variables discritization;

Load component data;

Build system dymanics model;

Set initial conditions;

**for** $D = 1, D_N$ **do**

  Find index of current state $s_D$;

  Choose $a_D$, depending on the minimum $Q(s_D, a)$ for all feasible actions;

  Use system dynamics to evaluate next state $s_{D+1}$;

  Set $s_{D+1} = s_D$;

  **if** *terminal* $s_{D+1}$ **then**

    end;

  **end**

**end**

The Q-learning was trained on the cloud server based on the results from DP, which were evaluated in the environment without external disturbance like traffic, other cars, driver behaviour, weather, etc. Although, in the real world, when implemented in the embedded controller, it will come across a variety of disturbances. Therefore, the actual performance of the Q-learning is sure to deviate from the training results. Hence, training the Q-learning on multiple possible drivecycles increase their adaptability and make it more robust to deviations from ideal.

## 5.4 Summary

This chapter explains the detailed development of the intelligent eco-driving based energy management strategy for connected and autonomous hybrid electric vehicles. The strategy is carefully developed to exploit the advantages of the CA-HEV's and provide high robustness to the algorithm and near-optimal performance. The algorithm is divided into two levels, i.e. the upper level controller and the lower level controller. The upper level controller is responsible for finding the ideal optimal velocity profile for the trip and finding the optimal power distribution trajectories for the found velocity profile, both using the DP. The ideal case refers to factors like driver behaviour, traffic, weather, and other vehicles not considered for simplicity. This task is carried out on the cloud server. Furthermore, the lower level is responsible for the real-time energy management of the vehicle and is embedded in the vehicle. This controller incorporates a Q-learning based algorithm for the EMS. As Q-learning is an RL-based algorithm, it needs to be trained before its real-time application, which is done on the cloud server. In the cloud server, based on the DP results, the Q-learning algorithm is trained for multiple ideal case results for the given trip. Training on multiple ideal scenarios ensures that the Q-learning algorithm implemented in the vehicle embedded controller can perform well in real-world scenarios where various external factors influence the driver's response. Furthermore thus, it can tackle the biggest challenge of incorporating external disturbances while developing the EMS for CA-HEV's.

# Chapter 6

# Results and Discussions

This chapter contains the results and discussions obtained after utilizing the developed intelligent eco-driving based EMS for CA-HEV's. The results are entirely based on simulation, with experimental verification being out of scope for the current thesis. The chapter is divided into three sections: test scenario, upper level controller and the lower level controller.

## 6.1 Test Scenario

In this section, the test scenario for evaluating the performance of the algorithm is discussed. As the algorithm is executed for the real-world energy management of the CA-HEV's it is imperative to choose a scenario that can best replicate the situation. For this purpose, the first task is to decide on a trip with both urban and highway driving. Hence, a course beginning from the author's house in Hamilton, Ontario, Canada, was chosen, which finishes at the Ron Joyce Centre, DeGroote School of Bussiness in Burlington, Ontario. In the route, McMaster Automotive

Resource Centre (MARC) was also added as a way-point in between the trip. This trip constitutes a total length of 22.2 km, in which there are 9 vehicle stops including stop signs and red light signals. A Google map view of the selected route is shown in Fig. 6.1.

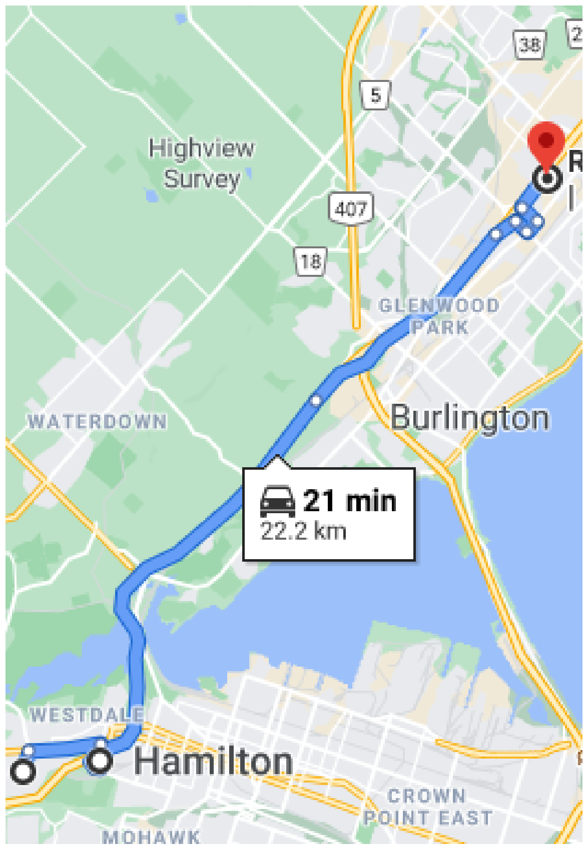


Figure 6.1: Google map view of the route

As already mentioned, the vehicle chosen for the validation of the algorithm is the Toyota Prius 2010 model, which has a power-split architecture. This information, once decided, constitutes the information that the upper level controller requires in the cloud server from the driver. The cloud server then can take this information to process the road data further, i.e. route speed limits and road grade for the current thesis, required by the DP algorithms. To imitate the process of the cloud server, taking the information from the driver and giving output as road data, an external open-source software named SUMO is used here. SUMO (Simulation of Urban Mobility) is a kind of traffic simulator that can be used for multi-modal microscopic traffic simulation [101]. SUMO has various tools that can be utilized to import road networks, algorithms, traffic, vehicle routing options, etc. It can also model real-world traffic according to requirements for real-world routes. It further has capabilities to be linked to MATLAB and extract various parameters from SUMO in real-time communication.

In this thesis, the SUMO, in collaboration with OpenStreetMaps, is used for extracting the trip data that incorporates the route's vehicle speed limits. OpenStreetMaps provide real-world road map data and are also open-source software. The OpenStreetMaps does not offer the topological road data, and therefore when used with SUMO, the road grade data for the trip cannot be evaluated. Hence, the road grade data for the route shown in Fig. 6.1 is considered as a constant value of zero radians.

To evaluate the road speed limits, the map of the city of Hamilton along with the nearby area is downloaded from OpenStreetMaps. Then using the SUMO, the route is created in the SUMO. An imaginary vehicle is also created in SUMO, which traces the selected route, as shown in Fig. 6.1. The SUMO is further interlinked with the MATLAB using the TraCI4Matlab, and then the created vehicle in the SUMO traces the trip. The information from the SUMO is communicated to MATLAB, which then can be processed to extract the road speed limits for the route. Fig. 6.2 shows the vehicle speed limit in $m/s$ as a function of the distance covered from the initial point.

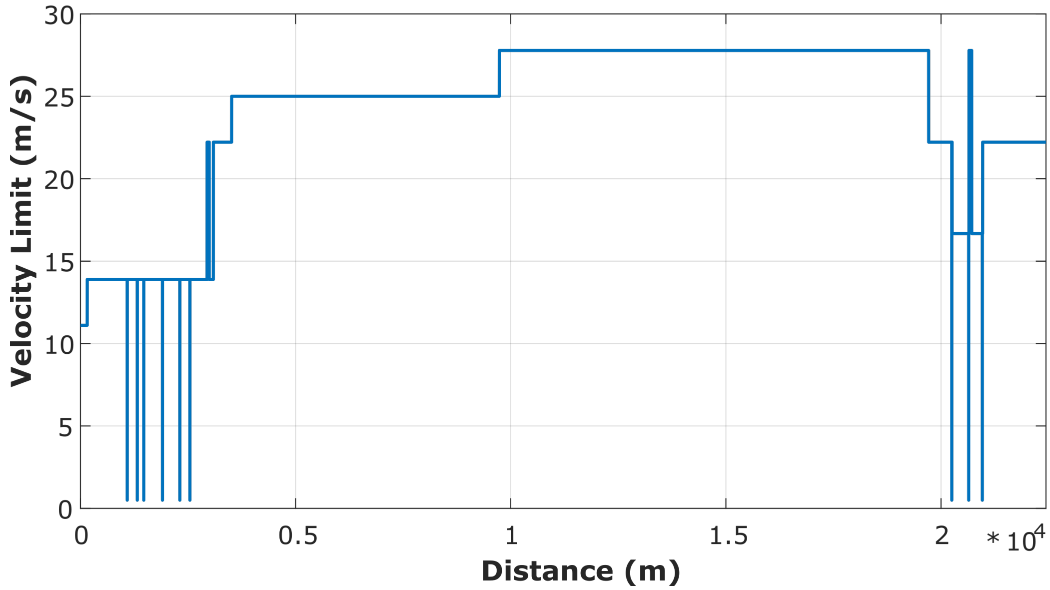


Figure 6.2: Speed Limit for the route

As it can be seen from Fig. 6.2 the speed limit of the route varies with the distance. This is because the route contains different sections of road in the city and on the highway. The speed limit plots also show sections where the speed limit is near zero, although this is not the case in the real world. The vehicle speed limits are set to near-zero values, where there are stop signs on the road or traffic lights in the real-world scenario. This ensures that the vehicle does stop at points where it is necessary to stop.

Once the road information is extracted, it can then be combined with the vehicle data to pass on the DP algorithm for formulating optimal solutions. The next task in the upper level controller involves finding the optimal velocity profile for the given route while not violating any constraints. The following section contains the results obtained from the DP in the upper level controller for the problem of optimal velocity and optimal power-distribution trajectories.

## 6.2 Upper Level Controller's Algorithm Results

This section contains the results for the optimal velocity problem and the optimal power distribution problem using the DP. The problem formulation for both of the cases discussed in chapter 4 under the optimal control problem section. As mentioned, these algorithms are solved on the cloud server as they require high computation power and then the results are used to train the RL controller. The following sub-sections discuss the results obtained for both the optimal control problems.

### 6.2.1 Optimal Velocity Resutls

The DP uses a combination of two processes for finding the optimal solution. First is the backward process, which evaluates the optimal cost-to-go for every state in the state-space grid and stores it along with its argument. The backward process begins from the endpoint and goes on to calculate cost till the start. The second step is the forward process which applies the action with minimum cos-to-go at every state beginning from the start. As the actions are executed at a given step on a state, the system progresses to the next state until the end is reached. The table shown in Tab. 6.1 shows the parameters used for the backward simulation, while the Tab. 6.2 shows the parameters for the forward simulation.

Table 6.1: DP Backward Simulation Parameters for Velocity Optimization

| Variable | Min | Max | Size |
|---|---|---|---|
| $v$ | 0 | 35 | 73 |
| $a$ | -3 | 2 | 51 |

Here, as shown in Table 6.2, the total number of steps in the simulation is 22439 $m$. The step size ($\delta D$) is $1m$. The initial velocity is 0 $m/s$ and the final velocity is

Table 6.2: DP Forward Simulation Parameters for Velocity Optimization

| Parameter | Min |
|---|---|
| $v(0)$ | 0 |
| $v(D_N)$ | [0,0.5] |
| $N$ | 22439 |
| $\delta D$ | 1 |
| $\alpha$ | [0,1] |

constrained between 0-0.5 $m/s$ due to discretization. The $\alpha$ is the weighing factor as shown in Eq. 4.19. As it can be seen from the Eq. 4.19, if the $\alpha$ is smaller; it favours $T_{travel}$ more, the velocity output profile will try to have less travel time. Whereas, if $\alpha$ is increased, the $F_t v$ is favoured, and the profile will try to minimize the power consumption required to complete the trip.

The figures shown in Fig. 6.3, Fig. 6.4 and Fig. 6.5 shows the results obtained after running DP for optimal velocity profile using the $\alpha = 0.07$.

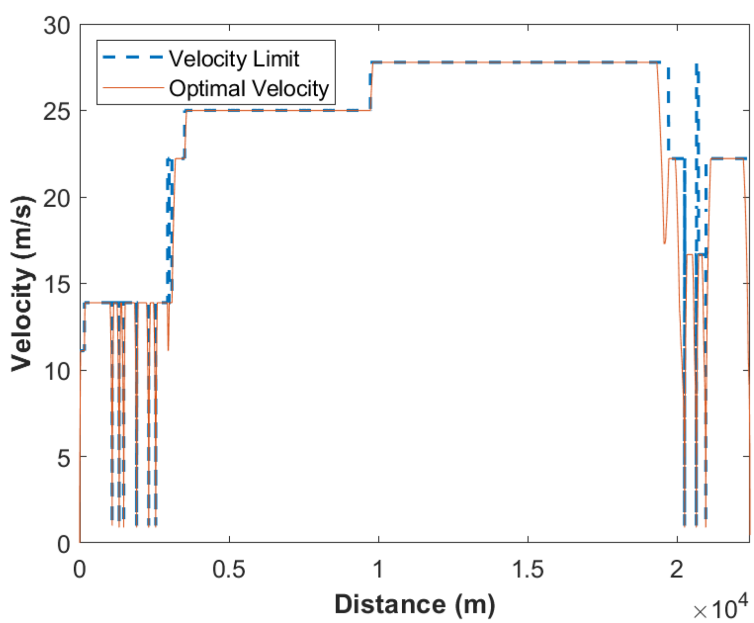


Figure 6.3: Velocity Profile for the route

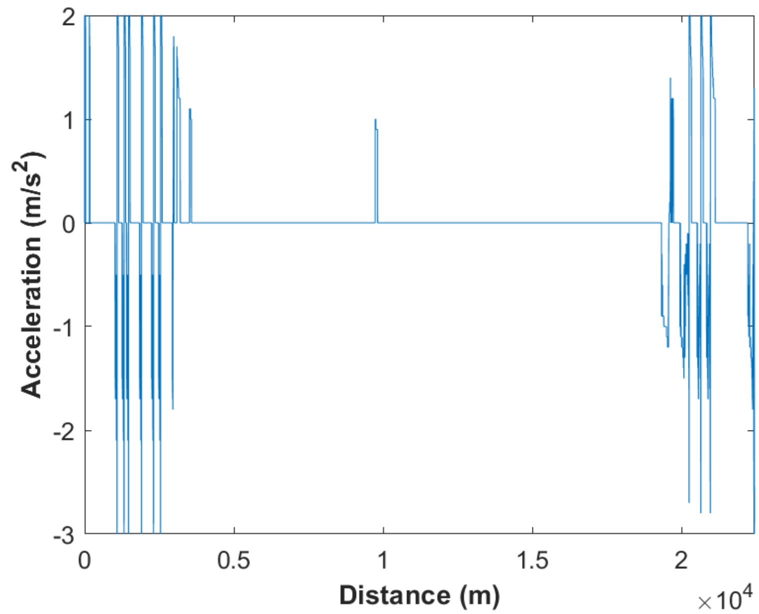


Figure 6.4: Acceleration Profile for the route

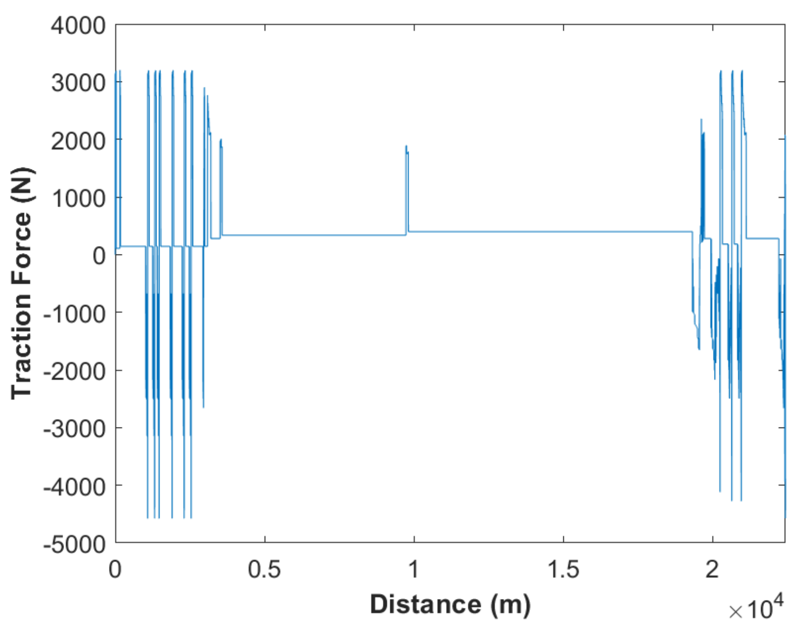


Figure 6.5: Traction Force Profile for the route

As it can be noticed from Fig. 6.3, because the $\alpha$ is a small number, the velocity always tries to be maximum while maintaining the upper limit constraints. Fig. 6.4 shows the acceleration for the evaluated optimal velocity, and the Fig. 6.5 shows the traction force required to drive the vehicle at the wheels. From the Fig.6.3 it can also be noticed that the DP tries to maintain the velocity of the vehicle constant, as, during the zero acceleration case, the traction force required to drive the vehicle is minimum, and so is the traction power.

Now, as the previous results were obtained when $\alpha = 0.07$, it is necessary to evaluate the results at different values of $\alpha$ to see their effect on the results; upon further evaluation, the optimal velocity solution was very sensitive to the value of $\alpha$. Hence, five different cases were taken to evaluate the optimal velocity results, where $\alpha_1 = 0.07$ for the case 1, $\alpha_2 = 0.075$ for case 2, $\alpha = 0.08$ for case 3, $\alpha = 0.085$ for case 4 and lastly, $\alpha = 0.09$ for case 5. The figure in Fig. 6.6 shows the velocity profile, acceleration profile and the traction force profiles respectively.

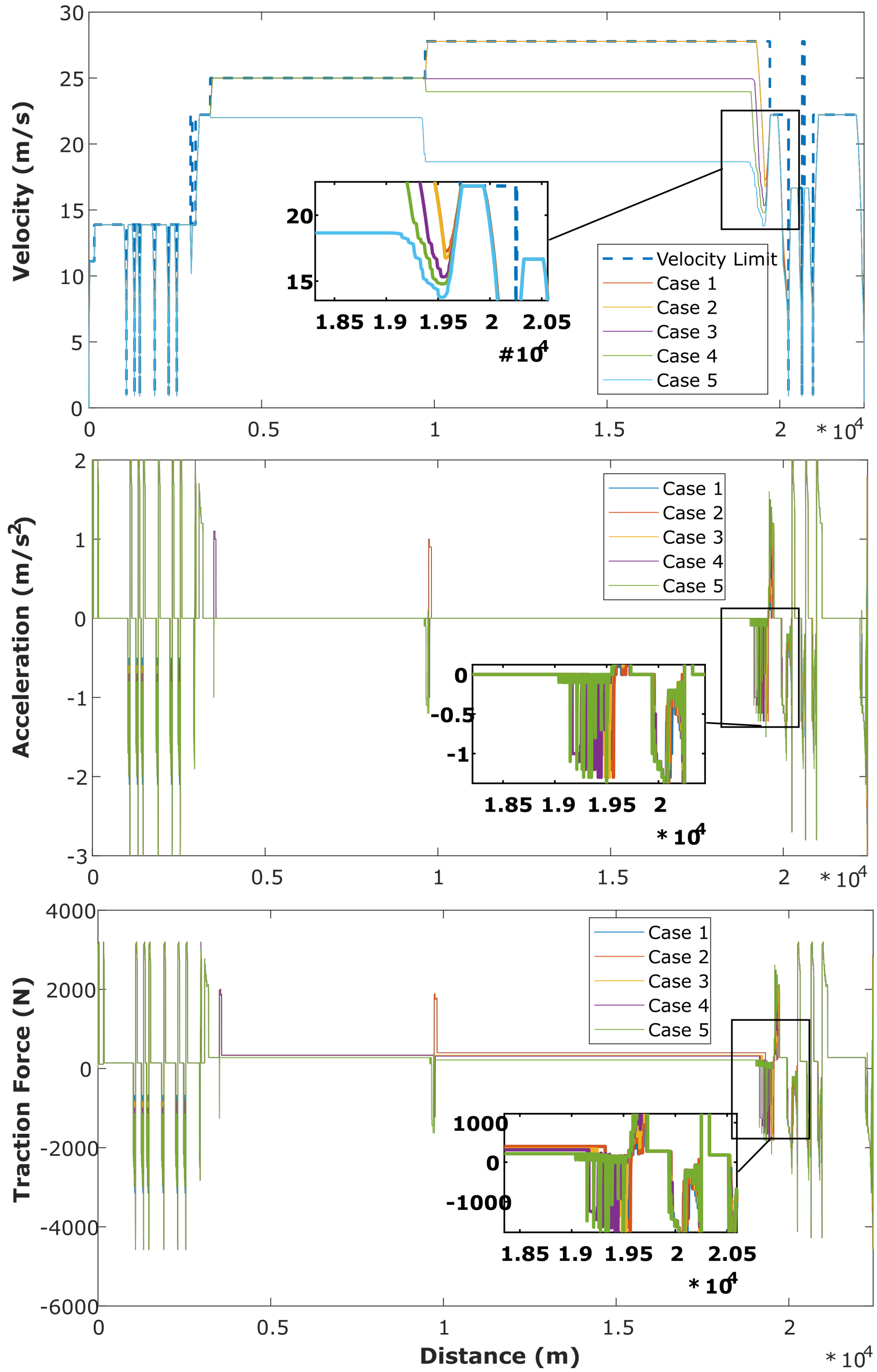


Figure 6.6: Different velocity, acceleration and traction force profiles for the route

From Fig. 6.6 it can be seen that the optimal velocity profile is susceptible to the value of $\alpha$. Furthermore, the velocity also starts to be less and less close to the upper limit as the value of the $\alpha$ increases. This difference is evident from the results of Case 1 and Case 5. However, in every case, it can be noticed that acceleration remains zero for a pretty significant distance, leading to less energy consumption.

A comparative analysis of all the five cases is shown in Table 6.3, where the total travel time of the trip $T_{travel}$, total cumulative energy consumption required to complete the trip i.e. integration $F_t v$ for whole trip, and the $\alpha$ values are shown.

Table 6.3: DP Results Summary for Velocity Optimization

| $\alpha$ | Cumulative Energy (MJ) | Travel Time (s) |
|---|---|---|
| 0.07 | 7.1 | 1079.1 |
| 0.075 | 7.1 | 1080.5 |
| 0.08 | 6.4 | 1124.5 |
| 0.085 | 6.2 | 1145.3 |
| 0.09 | 4.9 | 1297.2 |

As evident from Table 6.3, the cumulative energy consumption decreases with an increase in the value of $\alpha$. Moreover, the travel time keeps on increasing as the $\alpha$ increases. This shows a similar behaviour which was predicted from the nature of the cost structure of the DP for the optimal velocity problem.

### 6.2.2 Optimal Power Distribution Results

This subsection provides the results for the optimal power distribution of the CA-HEV. In the power-split Toyota Prius 2010, the power is provided by the ICE and the MGA and the MGB. Depending upon the vehicle's driver power request, the ICE either decides to provide power or the electric machines provide the propelling power. Hence, depending upon the distribution between the ICE, MGA and the MGB, the battery SOC is can be governed. It is imperative to keep the battery end SOC within the 1% deviation from the initial SOC due to the nature of HEV. The table shown in Tab. 6.4 shows the parameters used for the backward simulation of the DP, and the Tab. 6.5 shows the parameters for the forward simulation.

Table 6.4: DP Backward Simulation Parameters for Power Distribution Optimization

| Variable | Min | Max | Size |
|---|---|---|---|
| $SOC$ | 0.31 | 0.45 | 50 |
| $\omega_{eng}$ | 0 | 5000 RPM | 9 |
| $T_{eng}$ | 0 | 142 Nm | 9 |
| $Mode$ | 0 | 1 | 2 |

Table 6.5: DP Forward Simulation Parameters for Power Distribution Optimization

| Parameter | Min |
|---|---|
| $SOC(0)$ | 0.38 |
| $SOC(D_N)$ | [0.37,0.39] |
| $Mode(0)$ | 0 |
| $Mode(D_N)$ | [0.5,1.5] |
| $N$ | 22439 |
| $\delta D$ | 1 |

As shown in the Tab. 6.4, the $SOC$ of the battery is discretized into 50 points, $\omega_{eng}$ into 9 points, $T_{eng}$ into 9 points and the $Mode$ in either 0 or 1. The initial condition on the system, according to Tab. 6.5 state that the initial $SOC$ is 0.38, and the end $SOC$ should be within (0.37,0.39), i.e. within 1% deviation. Furthermore, the vehicle should start from the EV mode and should finish the trip in EV mode. The power distribution results obtained for the velocity profile obtained from the solution of optimal velocity profile as shown in Fig. 6.3, can be seen in the following figures.

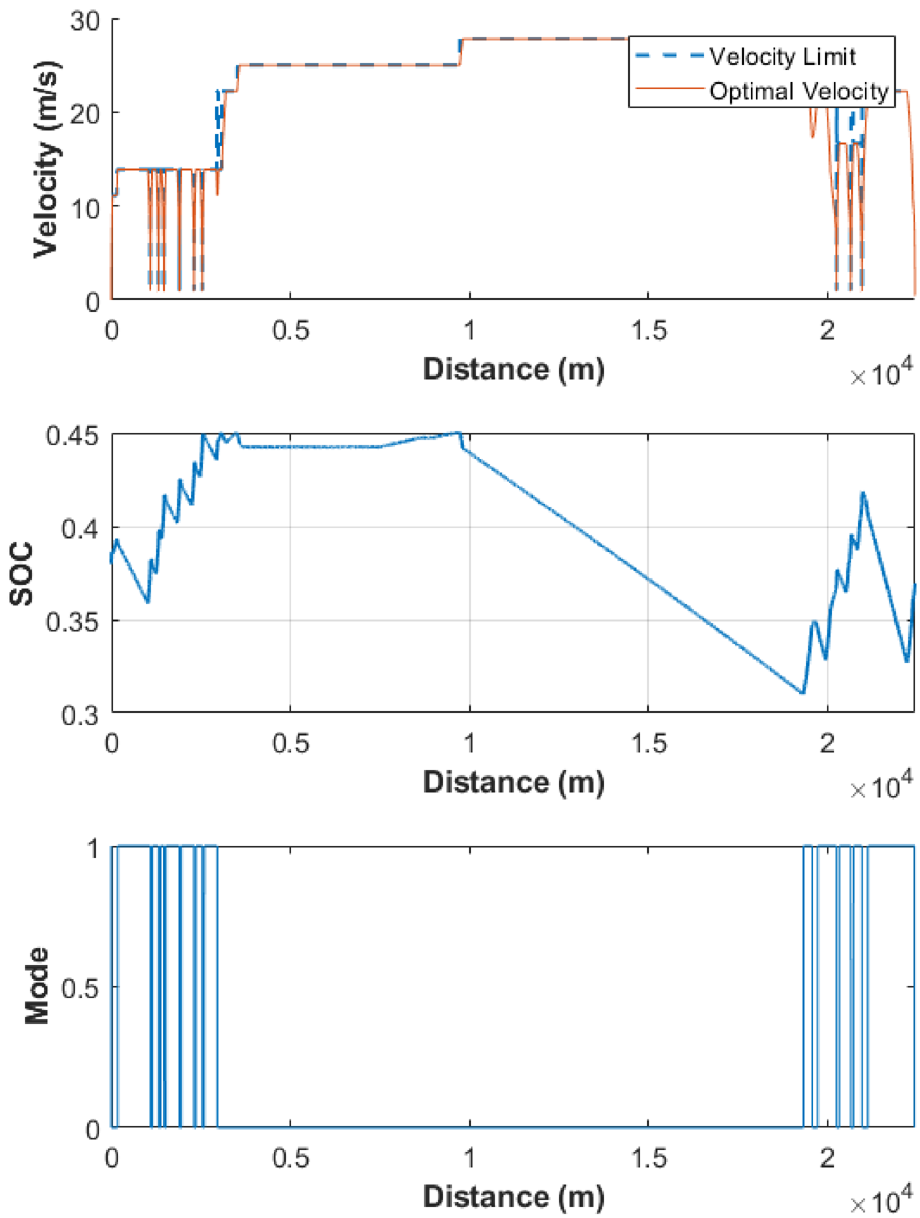


Figure 6.7: Velocity profile, SOC profile and Mode profile

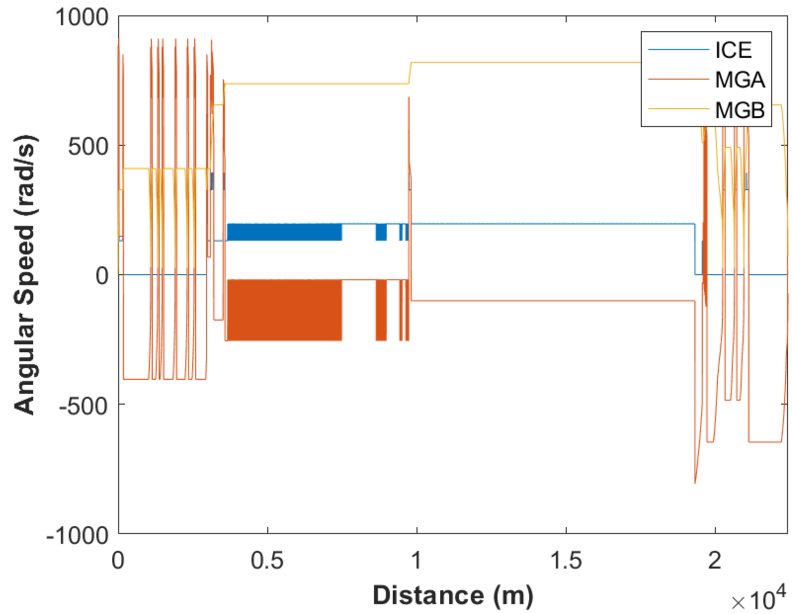


Figure 6.8: Speed profiles for ICE, MGA, and MGB

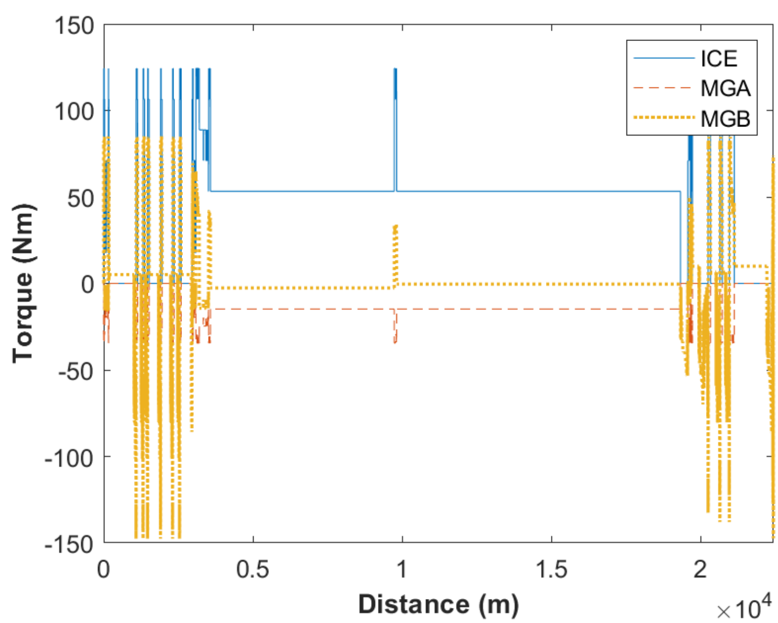


Figure 6.9: Torque profiles for ICE, MGA, and MGB

Fig. 6.7 shows the velocity profile for which the power-distribution optimal problem was solved, the SOC trajectory during that trip and the mode profile. It can be seen from it that the SOC begins from 0.38, and during the trip, it fluctuates between 0.3 to 0.45, and it ends at 0.37, i.e. within 1% deviation. The mode of the vehicle changes between 0 and 1, showing that in most of the journey, the ICE remains on, and it is sometimes off or is off during the regenerative braking.

In Fig. 6.8, the angular velocities of all the three components are shown. It can be noticed that the component speed of the MGB shows a response similar to vehicle output speed. As they both are mechanically linked to each other. As the ICE speed was a control variable in the formulation of the DP, some fluctuating response of the engine speed can be seen between 500 to 1000 m. For the remaining cycle, the speed response of the ICE can be seen as a smooth step response. The MGA and the ICE are connected to the same planetary gear; therefore, their angular velocities are coupled to each other. Hence, the angular speed of MGA will follow the same characteristic behaviours like that of the ICE speed.

In Fig. 6.9, the torques of all the three components are shown. The control input for the DP formulation was ICE torque, and depending upon the torque request by the driver, the torques of MGA and MGB are decided. Noticing from the shown response, it can be observed that the DP controller asks for constant positive torques demand from the ICE during the steady-state velocity and a positive demand when the vehicle is accelerating. The engine is switched off when the vehicle is decelerating, and the battery is recharged. Furthermore, the engine tries to operate at the peak efficiency zone, as the objective of the DP is to minimize the fuel consumption, which is evident from Fig. 6.10.

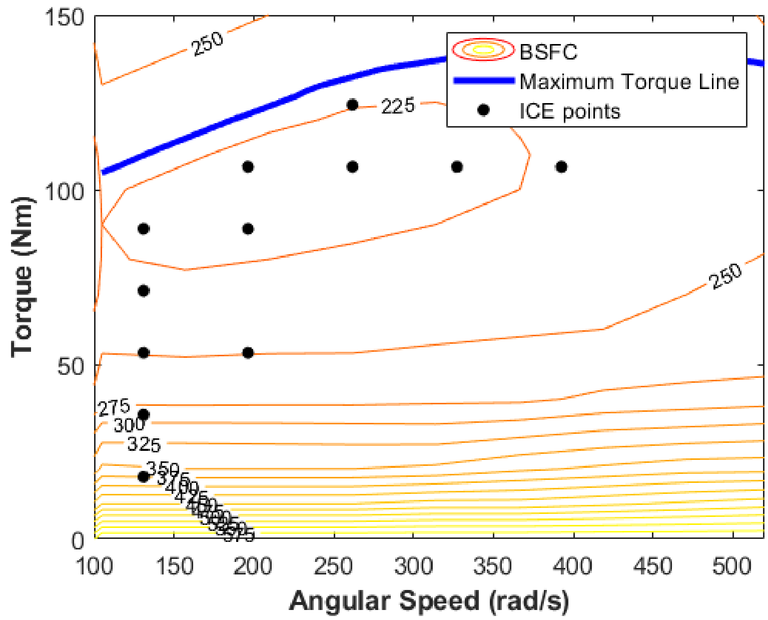


Figure 6.10: Operating points for ICE

As in the velocity optimization case, the optimal velocity was found out for 5 different cases. The optimal power distribution DP is run for those five different cases, and the results are shown below. Fig. 6.11, shows the velocity profile for all the five different cases evaluated earlier, along with the SOC profile for all the five different cases. Fig. 6.12 shows the cumulative fuel consumption for all the five cases.

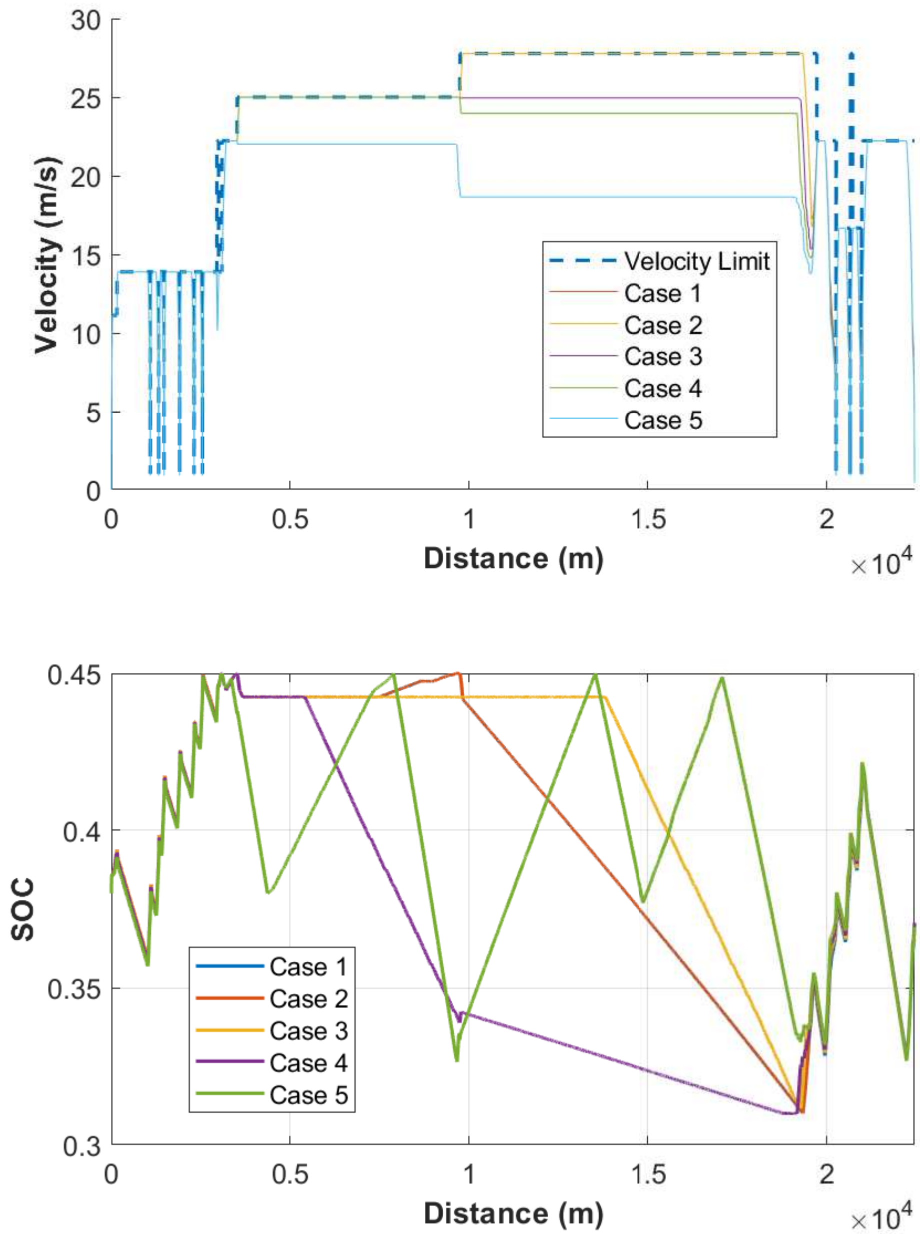


Figure 6.11: Velocity and SOC profiles for all cases

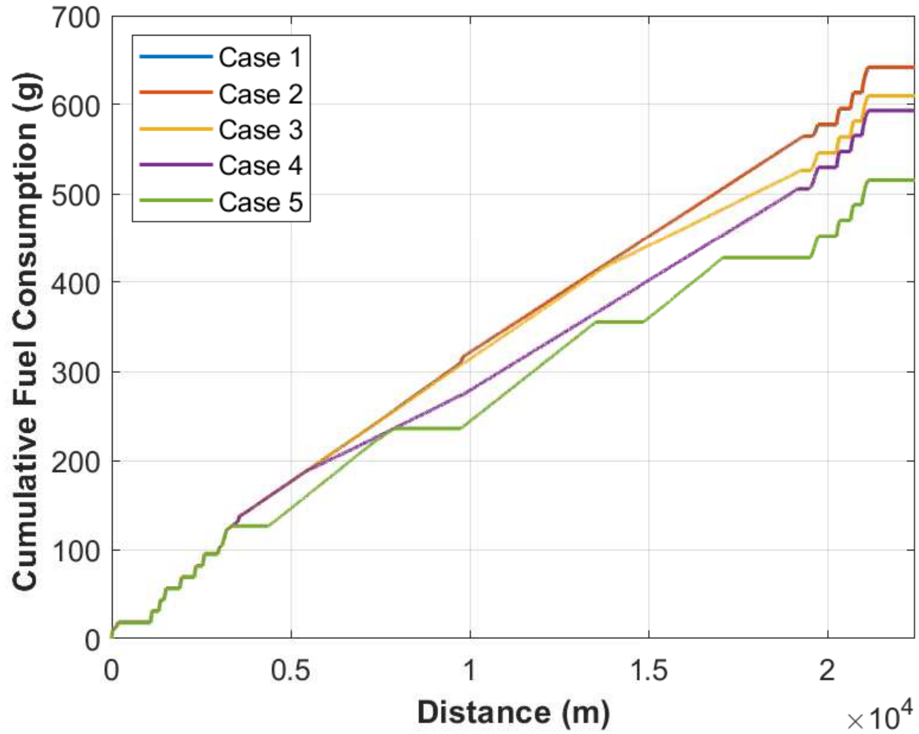


Figure 6.12: Cumulative Fuel consumption profiles for all cases

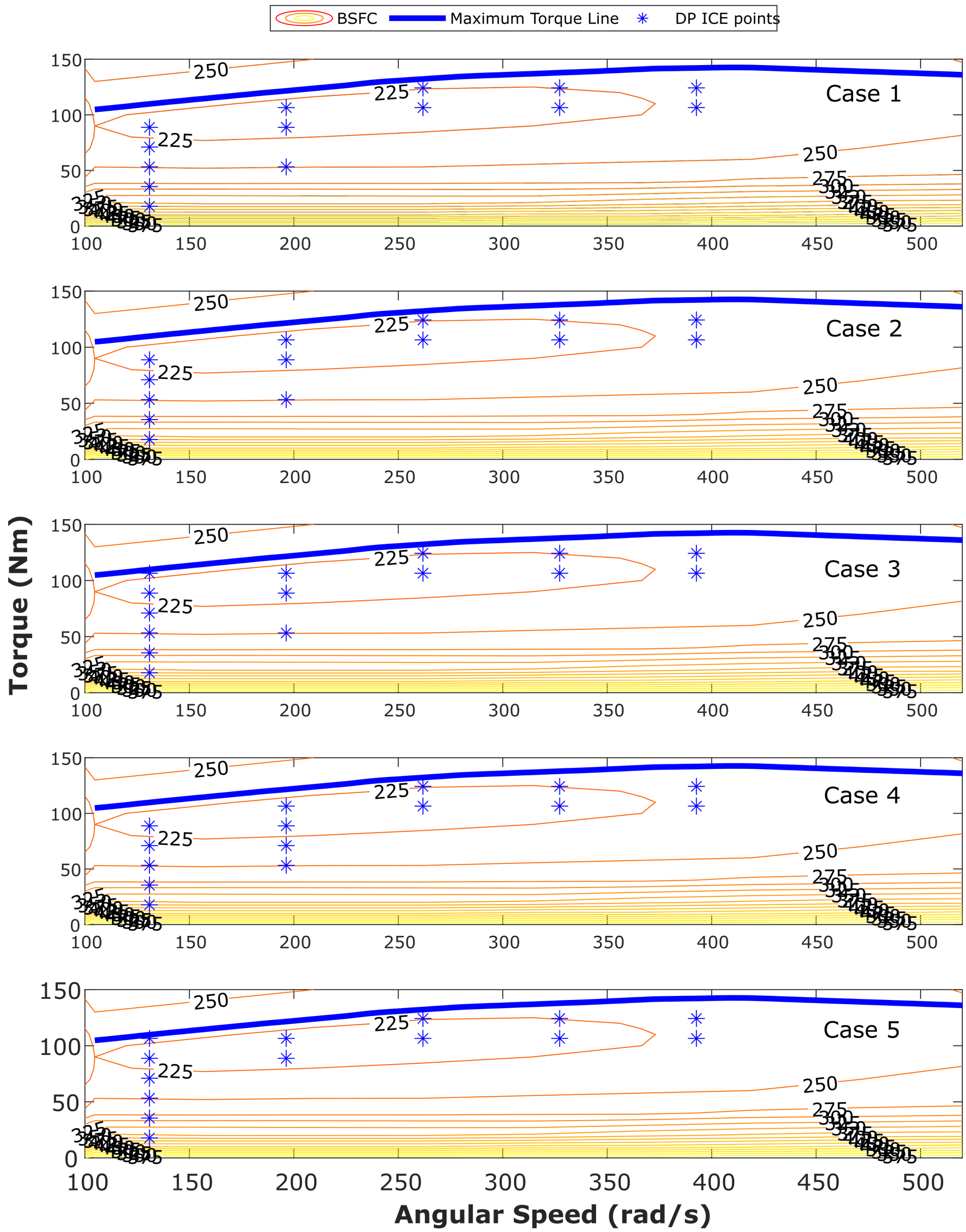


Figure 6.13: ICE operating points for all cases

From Fig. 6.11 it can be noticed that in all the five cases for the five velocity cases shown, the battery is in charge sustenance, and the final SOC is within 1% deviation. Furthermore, the battery always operate within the feasible operating zone. Similar to Case 1 results, there was some fluctuations observed in the operating speed of the ICE and MGA for Case 3 and Case 4. This is because in the DP, the $\omega_{eng}$ is the control variable, therefore to minimize the fuel consumption at every point, the DP chooses a point that can give minimum fuel consumption irrespective of the previous operating point of the engine. Hence, fluctuations can arise in the operation of the ICE with each passing step. The operating points of the ICE for all cases can be seen in Fig. 6.13.

The table shown in Tab. 6.6, summarizes the results obtained for the power-distribution optimal control problem. The table shows the cases, fuel consumption for the particular case, the cumulative energy supplied by the engine, the end SOC value in each case.

Table 6.6: DP Results for Optimal Power Distribution Problem

| Case | Cumulative ICE Energy (MJ) | Fuel Consumption | End SOC |
|---|---|---|---|
| Case 1 | 9.49 | 641.91 g | 0.3696 |
| Case 2 | 9.50 | 642.20 g | 0.3702 |
| Case 3 | 9.03 | 609.99 g | 0.3696 |
| Case 4 | 9.10 | 593.43 g | 0.3706 |
| Case 5 | 8.09 | 515.25 g | 0.3693 |

As it can be noticed from the Tab. 6.6, the fuel consumption of the ICE reduces with the increase in the case number due to velocity profiles associated with them and the ICE energy consumption. Although for Case 2, the fuel consumption is slightly higher due to more energy supplied by the ICE, it should also be noted that the end SOC is slightly higher, which accounts for the higher engine energy. The end SOC is almost within 1% deviation from 0.38, but in some cases, it is less than 0.37 due to a coarse discretization. Co-relating the results obtained from the Tab. 6.3 and Tab. 6.6, it can also be noted that the fuel consumption of the vehicle is reduced at the expense of the trip travel time. Therefore, the fuel consumption was decreased, but the travel time was increased. Hence, the value of $\alpha$ in the velocity optimal control problem can be decided on the trip's requirements or the driver's behaviour associated with the vehicle.

## 6.3 Lower Level Controller's Results

The lower level controller incorporates a Q-learning based RL agent, which is responsible for the energy management of the CA-HEV. As mentioned, for the Q-learning controller to perform well, the controller needs to be trained. This training can be done on the cloud server and is considered a part of the upper level controller. However, the Q-learning is implemented on the lower level controller; the following subsection shows the scope of comparative analysis, training results and testing results.

### 6.3.1 Comparative Analysis Scope

This subsection describes the scope and limitations of the lower level results obtained in this thesis. In this study, to analyze the performance of the lower level controller, first, its training results are evaluated. These results are compared to the optimal benchmark DP solution obtained from the upper level controller. This is done to understand how close an RL-based controller can perform compared to global optimal solutions. The comparative analysis of the RL-based controller training can signify whether the RL-based controller is sufficiently trained on known drivecycles to provide near-optimal solutions.

Subsequently, the RL-based controller is tested on two unknown near real-world drivecycles to evaluate its capability to perform in real-time. In this scenario also the RL-based controller results are compared with benchmark DP-based global optimal solutions. Comparing the RL-based controller results with DP-based results can provide a quantitative estimate of how well the algorithm performs, as the objective is to always tend towards the global optimal solution.

To further understand why it is compared against the DP solutions and not other real-time implementable strategies like ECMS is because it has been shown in the author's previous work that the RL-based controllers with enough training can perform better than ECMS strategy and provide better fuel consumption, and driveability [102]. Studies have also even shown that the well-trained Q-learning-based controllers can perform better than the ECMS controllers and can show up to improvement of 0.5 to 1.5 % in terms of fuel consumption [103, 104]. The Q-learning-based controller has also been shown to have better performance by 1.93 % than stochastic dynamic programming, which is also a real-time applicable strategy [105]. However, due to the stochastic nature of RL-based controllers and their training, it is now always possible that the RL-based controllers perform better than other deterministic strategies. Hence, it is essential to realize that the RL-based controllers should be considered for only fuel economy improvements. Rather their merits of being robust to disturbance and their potential to perform close to global optimal solutions should be emphasized.

### 6.3.2 Q-Learning Controller Training Results

The Q-learning controller is trained on the results obtained from the energy management using the DP. As the objective of the RL controller is to minimize the energy consumption and provide optimal power distribution at each instant, the RL controller does not take the reference as ICE or battery power; instead, it is trained to provide the optimal power distribution. The optimal power distribution or the optimal SOC trajectory cannot be taken here for training, as they are dependent upon the whole trajectory of the vehicle and hence acausal in nature. So for training, it is impossible to determine the optimal power distribution by looking at the current state of the vehicle, which constitutes a problem in defining the reward function.

Therefore, only the optimal velocity profiles and its transmission output power demand are used as the state variable to train the RL controller and the initial SOC of the vehicle and the engine status. The Tab. 6.7 shows the parameters used while training the RL controller, where the state variables, control variables and the definition of the parameters have already been discussed. The value of $\gamma$ was set as 0.95, and the learning rate $\alpha$ was set as 0.2. The maximum number of iteration ($N_{max}$) was set as 250.

Table 6.7: Parameters used for Training RL Controller

| Variable | Min | Max | Size |
|---|---|---|---|
| $SOC$ | 0.1 | 0.9 | 81 |
| $v$ | 0.01 m/s | 29 m/s | 29 |
| $P_{drive}$ | -31 kW | 52.5kW | 76 |
| $Mode$ | 0 | 1 | 2 |
| $\omega_{eng}$ | 0 | 5000 RPM | 9 |
| $T_{eng}$ | 0 | 142 Nm | 9 |

The RL controller uses an $\epsilon$-greedy policy to do the exploration and exploitation.

This means a random number $r$ is chosen, and if that random number is smaller than a pre-selected number ($j$), then the agent chooses a random feasible action. If $r$, is greater than $j$, the agent chooses the action with the minimum action-value function. Now, to ensure that the controller does not get stuck on the local optimal solution, the agent must do sufficient exploration. Therefore, for this purpose $j$ is made as a function of the maximum number of iterations ($N_{max}$) as shown in Eq. 6.1.

$$j = 0.7 - \frac{0.6i}{N_{max}} \tag{6.1}$$

Here, the $i$ is the current iteration number. As it can be understood from the Eq. 6.1, when the current iteration number is 1, the $j$ will be very close to 0.7. Moreover, as the iteration number increases, the $j$ will start to decrease, and at the end of the maximum number of iterations, i.e. $N_{max}$, the $j$ will become 0.1. Therefore, as the iteration increases with time, the RL agent will decrease the exploration and do more exploitation. This implies that the agent will be able to learn more during the initial stages, and with increasing iterations, it will exploit more of the policy with a minimum action-value function. The results for the case 1 drive cycle are shown in the figures below.

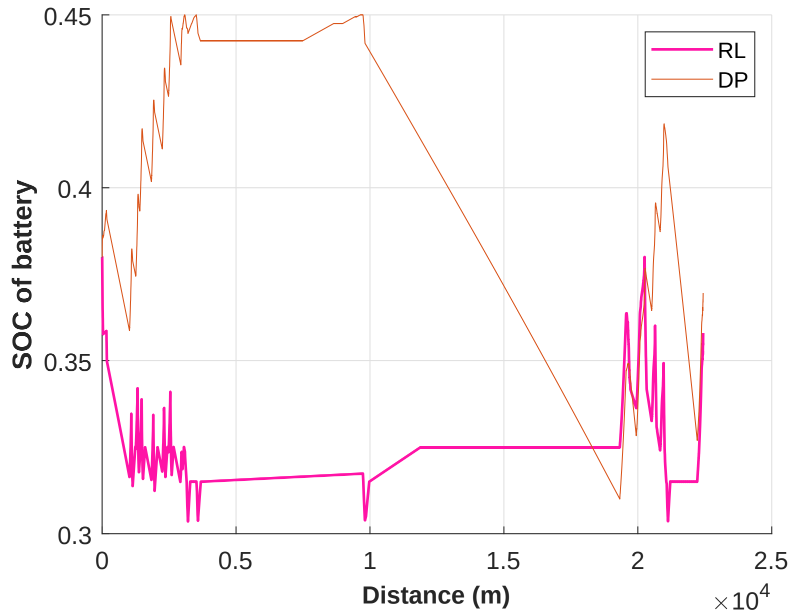


Figure 6.14: SOC for case 1

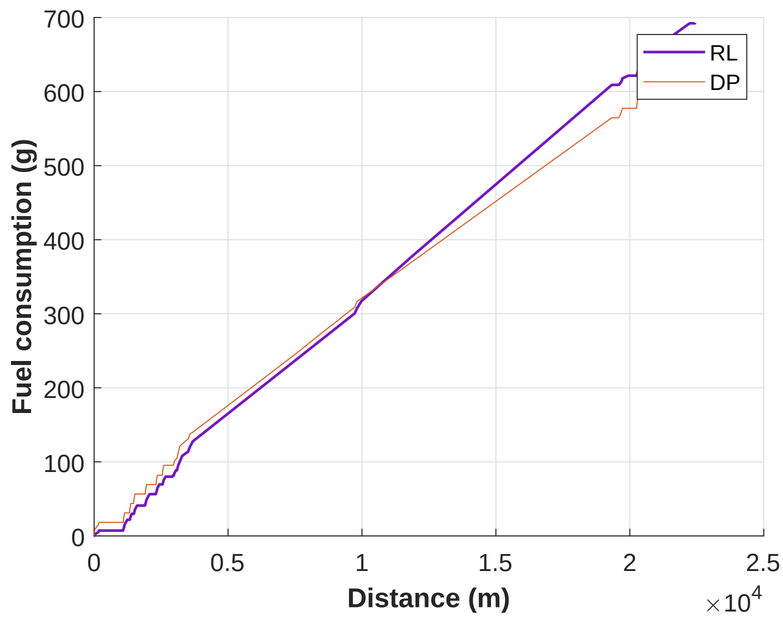


Figure 6.15: Cumulative fuel consumption for case 1

As it can be seen from Fig. 6.14, that although the SOC trajectories of the RL-based controller deviates when compared to the optimal trajectory from the DP-based controller, the end SOC are very close to each other. This indicates that the RL-based controller can give charge sustaining results. Furthermore, the cumulative fuel consumption trajectories of DP-based and RL-based controllers are shown in Fig. 6.15. In the initial 1000m, the RL-based controller's fuel trajectory is lower than the DP's; however, the overall fuel consumption of the RL-based controller is slightly higher. This increase in total fuel consumption is understandable as the DP is a global optimization technique, while the RL is a learning-based method.

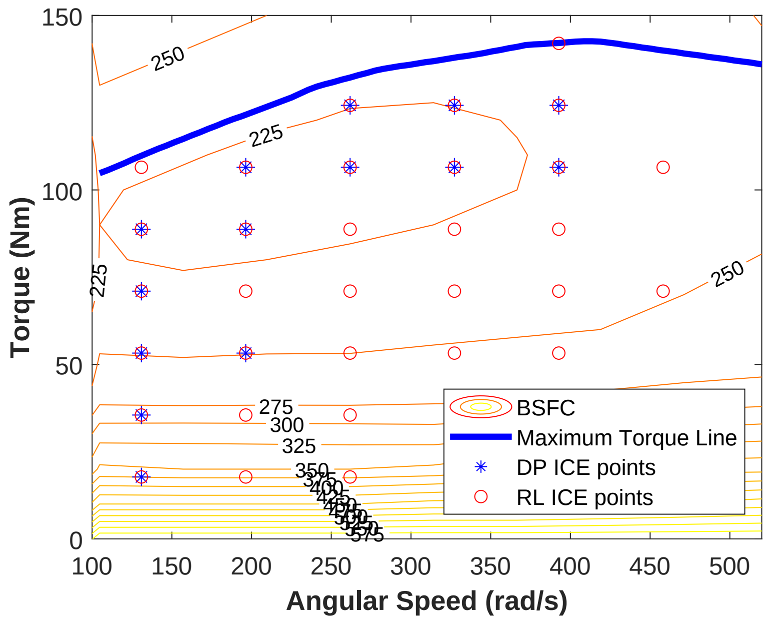


Figure 6.16: ICE operating points for RL and DP based controllers

The ICE operating points in the case of the DP-based and RL-based controllers can be visualized in Fig. 6.16. It is evident that the operating points of the RL-based controller vary much more when compared to the DP-based controller's ICE operating points. As the DP is offline, and its objective is to minimize the fuel consumption, it selects more efficient points while the RL-based controller selects points with a minimum action-value function.

Now, as mentioned, to appropriately train the RL-based controller, it was trained on five different optimal velocity trajectories. The results for all five cases are shown below:

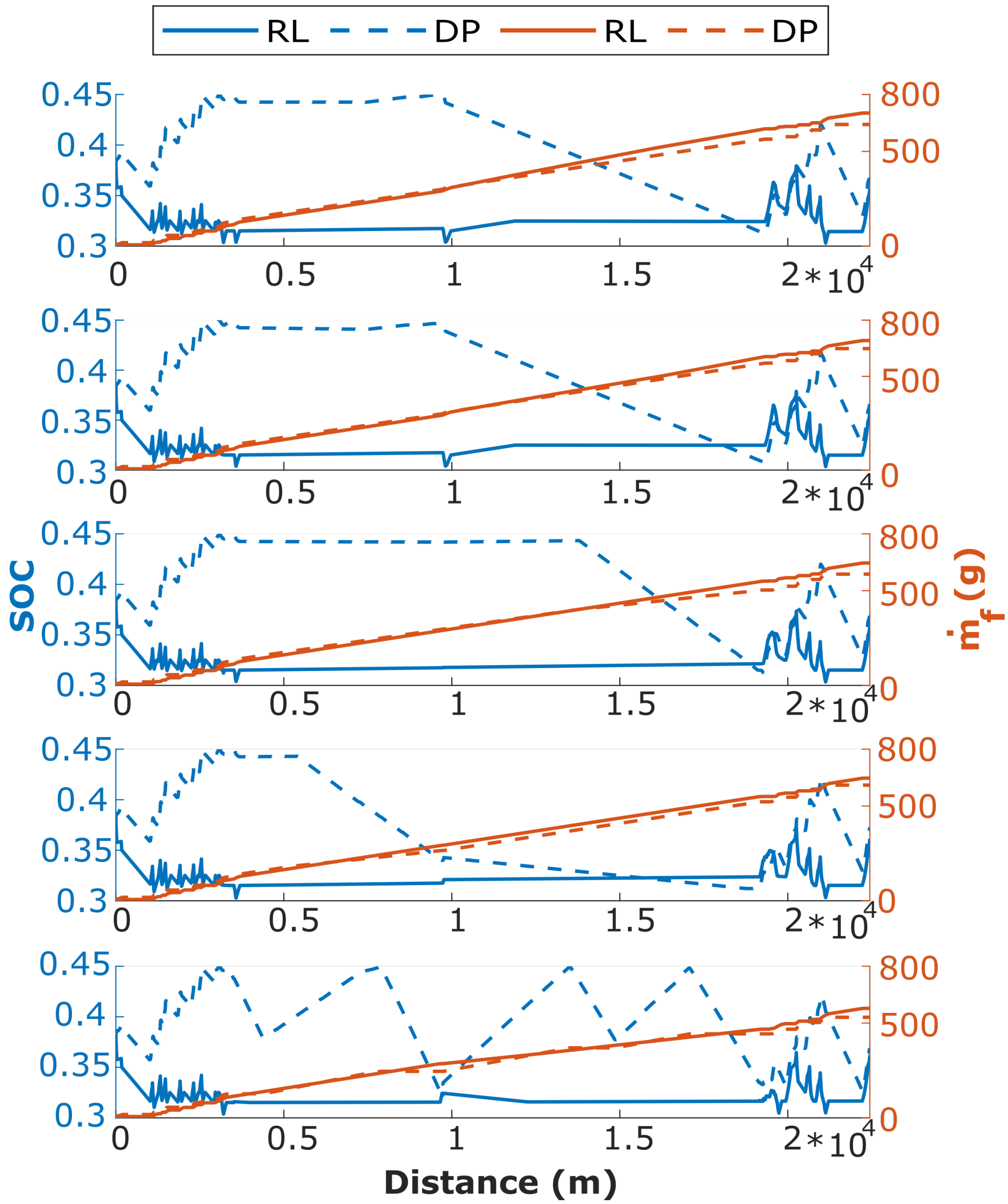


Figure 6.17: SOC and cumulative fuel consumption trajectories for RL and DP based controllers

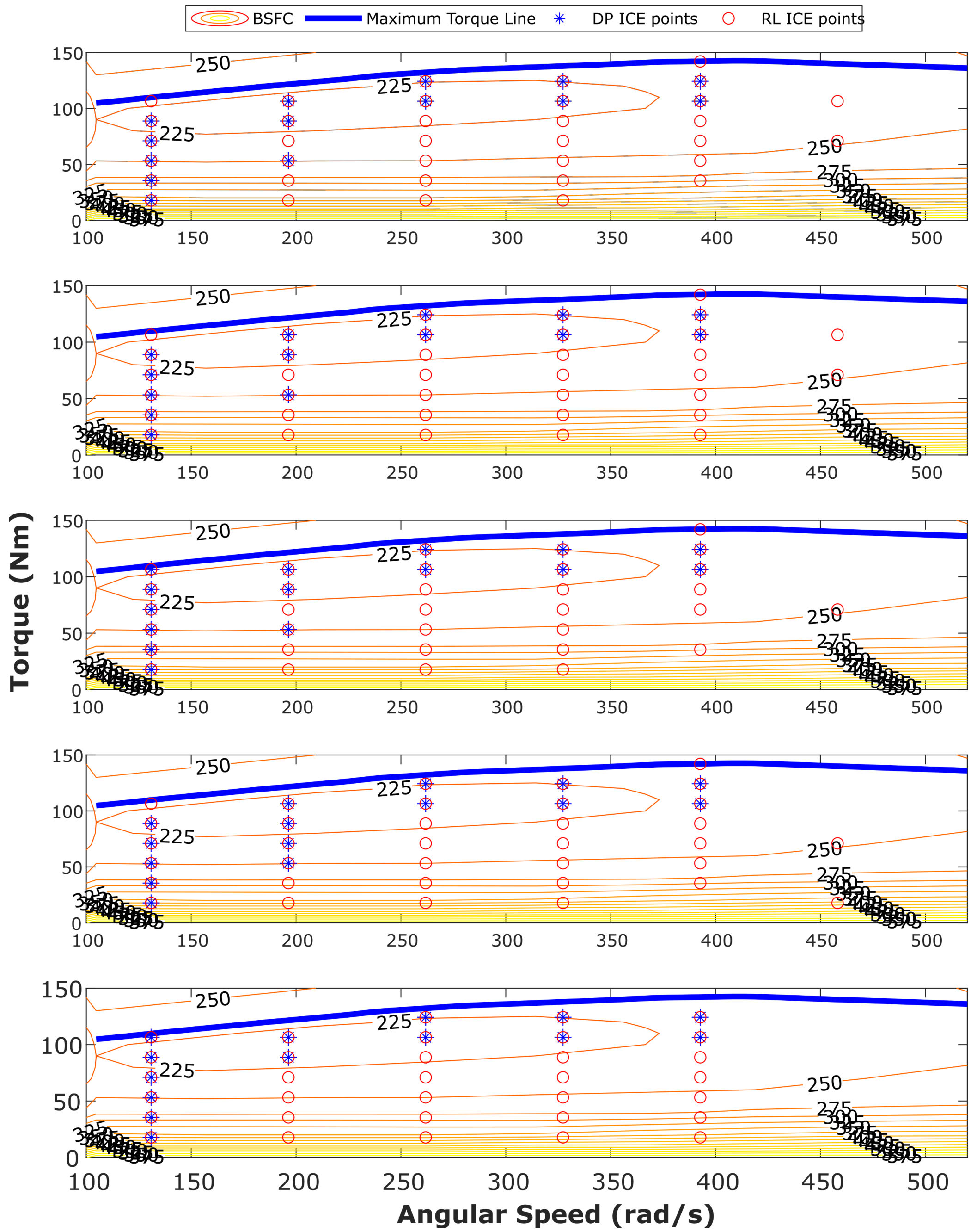


Figure 6.18: ICE operating points for RL and DP based controllers

Fig. 6.17 shows the SOC and the cumulative fuel consumption trajectories for both the DP and RL-based controller in all five cases. It can be analyzed from it that the SOC in all the five cases shows different behaviour from the optimal one; however, the RL-based controller is still charge sustaining in all the cases. Fig. 6.18 shows the operating points for the ICE in both the controllers on all the five drive cycles. It can be seen that the DP in all five cases selects more efficient points; however, the RL-based controller selects the points on the whole operating zone. Therefore, its effect can be seen in the cumulative fuel consumption trajectories in Fig. 6.17, where the DP being global optimal always performs slightly better. Hence, a similar trend in all five cases is observed for the SOC behaviour and the selection of the ICE operating points for the RL-based controller. For better understanding, the performance difference, the Tab. 6.8 shows the total fuel consumption of the RL-based controller along with its % deviation from the global optimal results in terms of fuel consumption.

Table 6.8: RL Results for the Optimal Velocity Trajectories

| Case | $\dot{m}_f$ | SOC(end) | % deviation |
|---|---|---|---|
| Case 1 | 693.93 g | 0.3593 | +9.70 |
| Case 2 | 691.77 g | 0.3588 | +9.28 |
| Case 3 | 667.20 g | 0.3590 | +11.01 |
| Case 4 | 630.57 g | 0.3611 | +7.76 |
| Case 5 | 584.65 g | 0.3604 | +15.26 |

Tab. 6.8 also shows the end SOC points achieved by the RL-based controller in each case. It can be seen that it is always close to the starting value of 0.38, however not exact charge sustaining due to coarse discretization of the $SOC$ and $P_{drive}$. Furthermore, fuel consumption deviation between the optimal DP results and the RL training results is less than 10% except for case 3 and case 5. This signifies good performance from the RL-based controller in terms of fuel consumption minimization against the benchmark solutions. The performance of the RL-based

controller can further be enhanced by training it over more drivecycles and increasing the number of iterations to reach convergence.

### 6.3.3 Q-Learning Controller Testing Results

The subsection results above show the RL-based controller's response while training on the ideal optimal velocity drivecycles obtained by the DP. As in the ideal environment considered while evaluating the DP results, the traffic was ignored along with external disturbances; it is essential to check the performance of the RL-based controller in real-world environment velocity trajectory. Therefore, for this purpose in this thesis, two approximate real-world drivecycles are generated using the DP-based velocity optimization technique.

For evaluating the two approximate real-world drivecycles, the $\alpha$ values were changed with respect to distance in the upper-level controllers' velocity optimization algorithm. Thereby a random distribution of $\alpha \in [0.07, 0.10]$ varying with distance was selected to obtain two drivecycles. This ensures that the obtained real-world velocity profiles show similar behaviour to the ideal velocity profiles and have deviations that make them closer to the real-world situation. Hence, the approximate real-world drivecycles will show the characteristics of the ideal drivecycles, although they will have disturbances that can mimic the traffic and uncertainties the vehicle will encounter in the real world.

The velocity for the two approximate real-world drivecycles can be seen in Fig. 6.19.

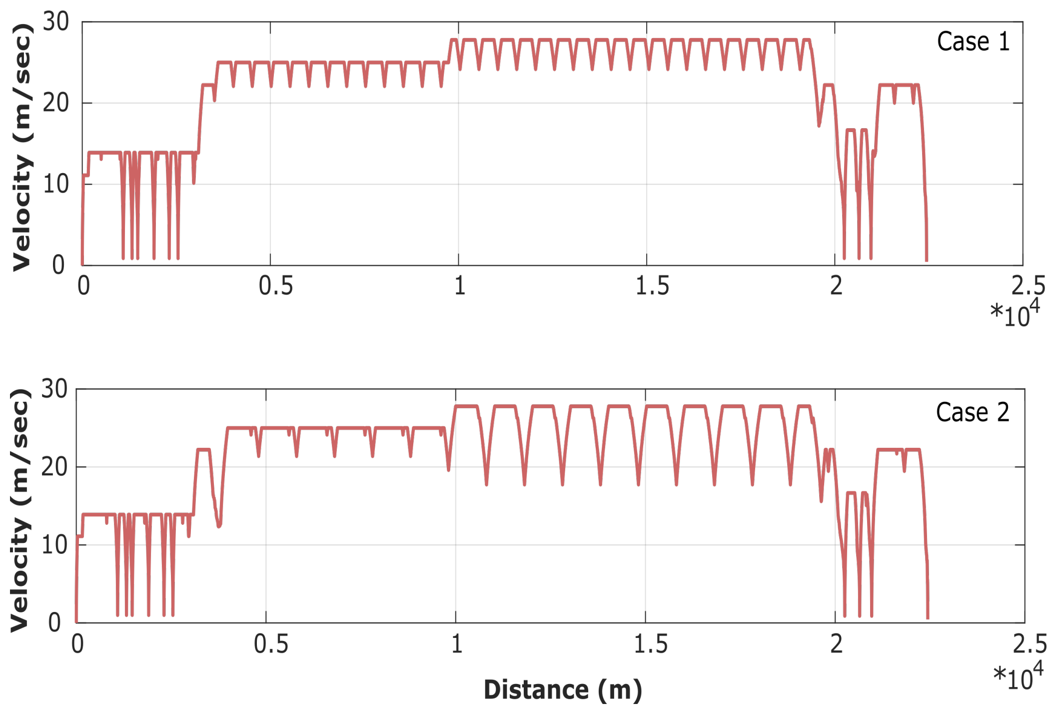


Figure 6.19: Two approximate real-world drivecycles

It can be observed from Fig. 6.19 that the velocity profiles indeed are very similar to the ideal optimal velocity profile obtained from case 1. However, there are many disturbances in the driveccyles due to the changing value of $\alpha$, which emulates the uncertainties in the real world. Case 1 in Fig. 6.19 is a profile with conservative regular deviations from the ideal case, which can happen due to traffic, leading to constant acceleration and braking to maintain the ideal optimal velocity. In Case 2, the disturbances are far significant, and the speed deviates to a greater extent. This signifies a heavy traffic flow, along with obstructions in the route, which explains the large distances covered while deviating from constant velocity. In this sense, Case 2 shows more aggressive behaviour, where the driver deviates more from the ideal optimal velocity trajectory.

Now, to evaluate the performance of the RL-based controller, it is tested on these two approximate real-world drivecycles and compared against the global optimal solution evaluated using the DP. It is important to note here that the RL-based controller was not trained on these two new approximate real-world drivecycles; instead, it was trained on five ideal optimal cases. Therefore, these velocity trajectories are unknown to the RL-based controller, and their decisions are based on the action-value function of the state at a given instant. Whereas the results evaluated using the DP are global optimal, where the whole driving cycle was known as a-priori. Fig. 6.20 and Fig. 6.21 shows the comparative results between the RL and the DP based controller for both the cases, respectively. In both the plots, the topmost plot shows the comparison of the SOC trajectories. The second plot is for the cumulative fuel consumption trajectory. The third and the fourth plot represent the ICE operating speed and torques for both the controllers.

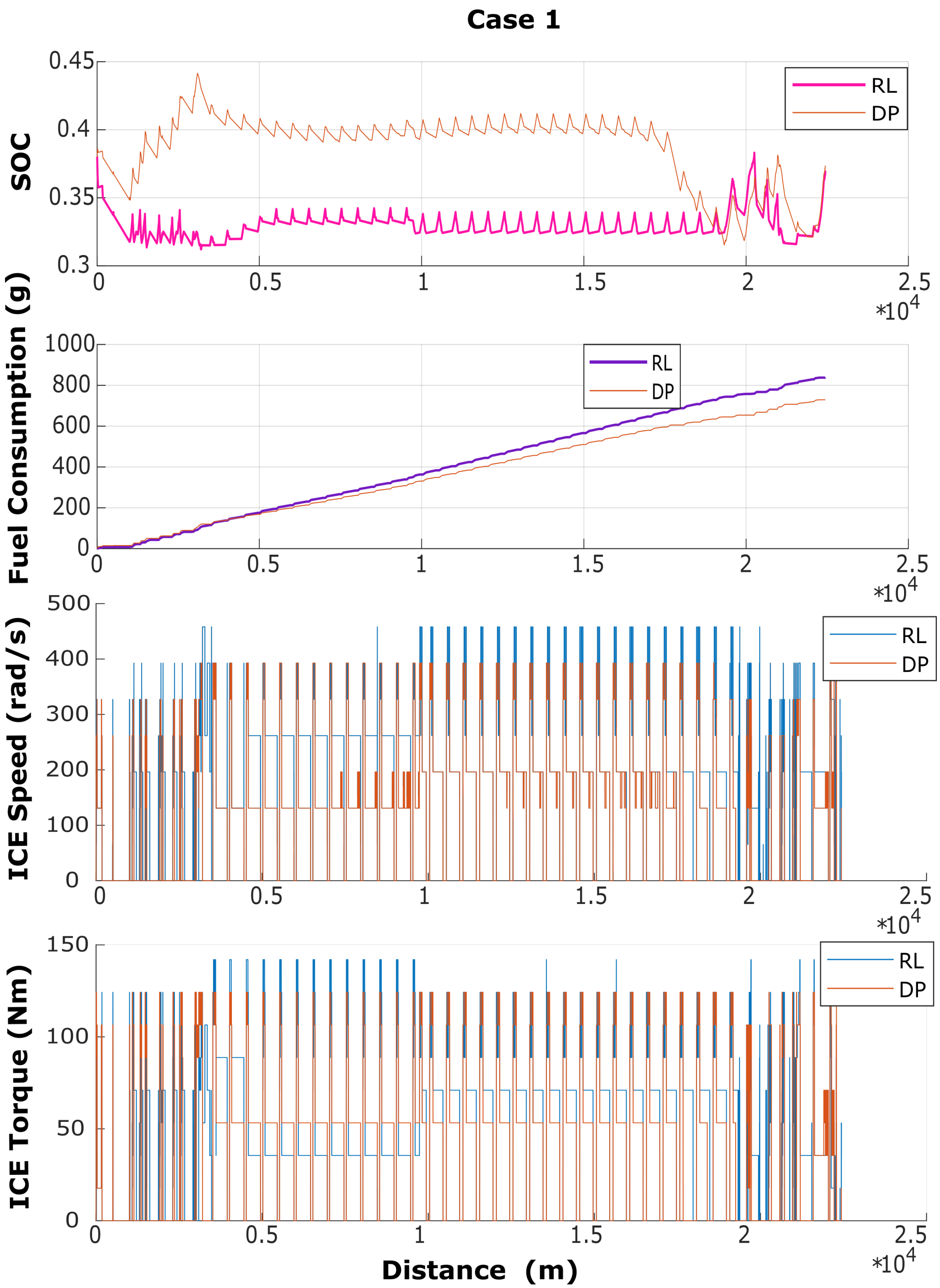


Figure 6.20: Comparative results between RL and DP based controllers for Case 1

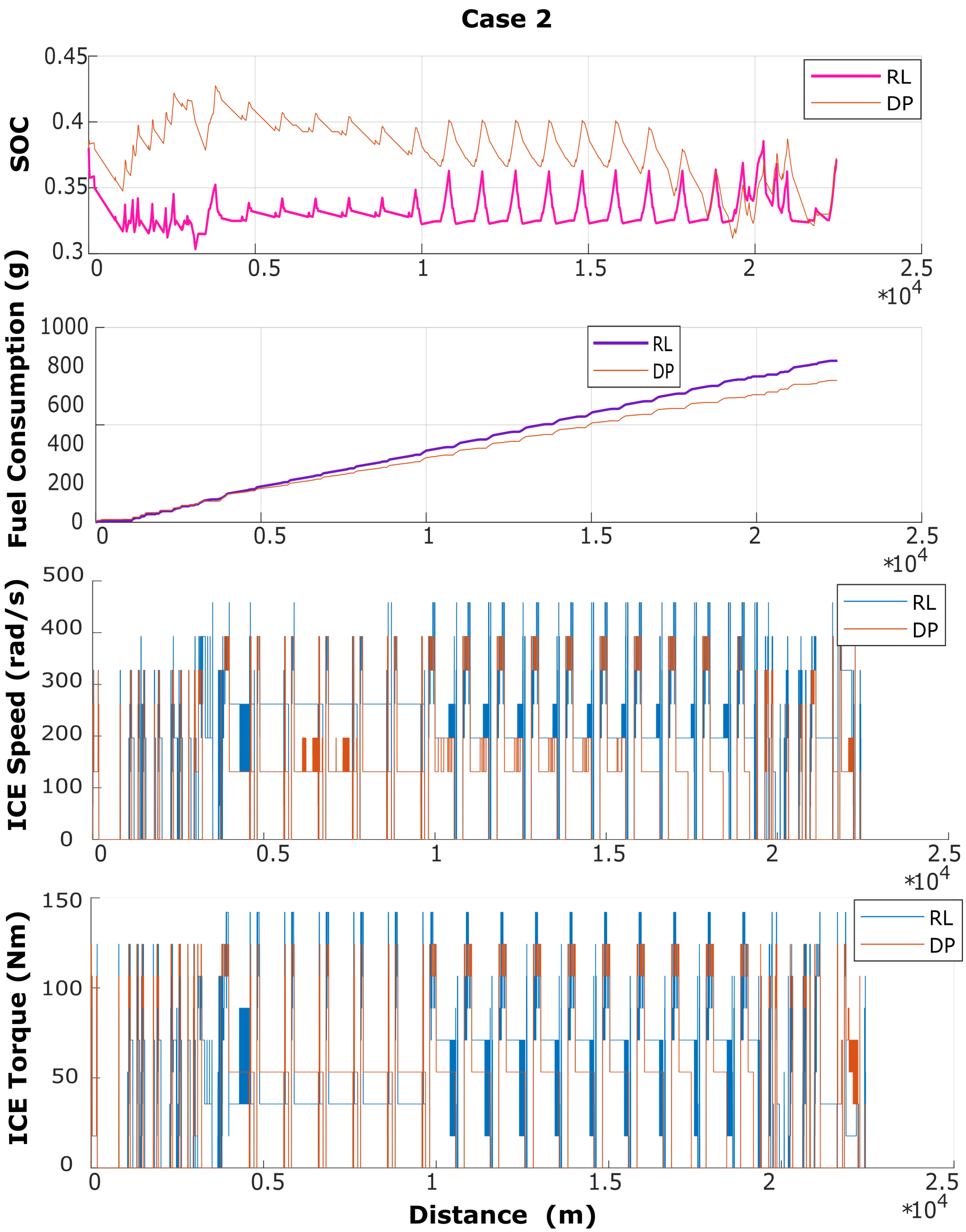


Figure 6.21: Comparative results between RL and DP based controllers for Case 2

For the Case 1 results in Fig. 6.20, it can be noticed that the end SOC from both the controllers is very close to each other and the starting value of 0.38. This means that, although the SOC profile of the RL-based controller differs from the DP-based controller during the trip, the RL-based controller can provide charge sustaining results. For case 1, as the velocity trajectory was less aggressive, it can be seen that the ICE operating speeds and torques give a similar response for the RL and the DP-based controllers. Though the exact ICE speed and torques are different, they are close, with similar trajectories and similar engine activation duration and instants. Therefore, a similar SOC trend was observed with a constant SOC delta during the drivecycle, and the cumulative fuel consumption profiles are close to each other.

For the Case 2 results shown in Fig. 6.21, it can again be noticed that the end SOC of the RL and the DP-based controllers are very close to each other and the starting value of 0.38. Hence, the RL-based controller again shows charge sustaining results. Now, in case 2, as the velocity profile was a little more aggressive in terms of deviations, it can be seen in the ICE operating speeds and torque that the behaviour of RL and the DP-based controllers differ more in this case. This is because while training, the RL-based controller would have had limited interaction with the particular state, which is new in the approximate real-world drivecycle. This causes the estimate of the action-value function of that state-action pair to be inaccurate. However, the engine activation duration and instants display similar behaviour. As the ICE operating points vary more, its effect can be seen in the SOC trajectory, which shows the different responses in the case of RL and DP-based controller. The SOC in the case of RL seems to be fluctuating around a constant value throughout the journey; however, for the DP-based controller, it increases in the beginning and then slowly decreases over the trip duration. Although, in this case also a very close response of the battery SOC is observe during the end duration of the drivecycle, and the SOC is always within the feasible region.

The total cumulative fuel consumption of the RL-based controller is a little higher than the DP-based controller, which was expected due to DP's global optimal nature. The table shown in Tab. 6.9 shows the testing results of the RL-based controller on the approximate real-world new drivecycles. It shows the total cumulative fuel consumption of the DP ($DP_{\dot{m}_f}$) and the RL-based controller ($RL_{\dot{m}_f}$). It also shows the end SOC value obtained from both the controllers and the % deviation between the DP and the RL-based controller after SOC correction.

Table 6.9: RL Testing Results on Approximate Real-World Drivecyles

| Case | $DP_{\dot{m}_f}$ | $RL_{\dot{m}_f}$ | $DP_{SOC(end)}$ | $RL_{SOC(end)}$ | % deviation |
|---|---|---|---|---|---|
| Case 1 | 719.72g | 838.17g | 0.3738 | 0.3701 | +17.10 |
| Case 2 | 723.72g | 839.07g | 0.3727 | 0.3711 | +16.52 |

As evident from the Tab. 6.9, the RL-based controllers' total fuel consumption is higher than the DP-based controller. However, it performs reasonably good given it was an unknown drivecycle for the RL-based controller. Furthermore, it can also be noticed that the end SOC in both cases is very close to the initial SOC value of 0.38, which implies that the controllers are charge sustaining. The % deviation between both the controller's total fuel consumption after SOC correction can be seen in the last column that indicates that the deviation in both the cases is less than 20%. This signifies good performance, considering that the fuel consumption obtained from the DP is a global optimal value. Therefore, the testing results indicate that the RL-based controller can give charge sustaining results and provide comparable results to global optimal solutions even on unknown approximate real-world drivecycles for the given route.

## 6.4 Summary

This chapter presented the results obtained from the intelligent EMS developed using the RL and the DP for eco-driving in connected and autonomous vehicles. A detailed discussion on results is done to understand the advantages and limitations of the novel algorithm thoroughly. The chapter explained the test case scenario in which the algorithm is tested and then moved onto the results obtained from the velocity optimization in the upper-level controller. After the results from the velocity optimization are discussed, its power-distribution results are evaluated and explained, used as the benchmark for the RL-based controller's training. Next, the RL-based controller's training results are described, which mentions the training process and compares its performance with the global optimal results obtained from the DP. Observing from the training results it was evident that the RL-based controller was able to find solution close to global optimal results from the DP. At last, the trained RL-based controller is tested on two approximate real-world drivecycles for the same trip to evaluate its performance capability in the real-world environment where uncertainties in the form of other vehicles and traffic will be present. The testing results showed that the RL-based controller could provide charge sustaining results and comparable results to the global optimal solution evaluated using the DP, proving its adaptability and robustness to uncertainties in the real world.

# Chapter 7

# Conclusions and Future Work

The automotive industry is currently witnessing a paradigm shift towards a greener, sustainable and more intelligent transportation sector. This shift emerges from advancement in two mainstreams, i.e., electrification and automation. Electrification focuses on developing technologies to convert conventional vehicles into more and more electrified vehicles. At the same time, automation focuses on developing technology to make existing vehicles more intelligent, user-friendly and connected to the external environment. The culmination of these two fields results in next-generation vehicles in the form of connected and autonomous hybrid electric vehicles (CA-HEV's). In this thesis, a brief introduction was provided for each of the fields, and detailed information was provided on how their interaction results in uplifting challenges encountered in the domain of energy management strategies (EMS). In this chapter, conclusions drawn from the development of novel EMS for eco-driving in CA-HEV are presented. Furthermore, potential methodologies that can be incorporated for improving the results are presented in future work.

## 7.1 Conclusions

The motive of this thesis was to develop a novel EMS for CA-HEV's and validate its performance. For this purpose, the first task was to do a comprehensive literature study to understand the concept of energy management and its role in CA-HEV's. Therefore, a detailed study was undertaken on the EMS's for conventional vehicles, and it was extended onto energy management of CA-HEV's. Based on the findings from the literature review, it was noticed that there exists a conceptual gap where the incorporation of learning-based EMS's has not been explored yet for the application of eco-driving in CA-HEV's that can overcome its challenges. Hence, the objective of this thesis was to develop a novel intelligent EMS for eco-driving in CA-HEV's utilizing learning-based EMS's to tackle the limitations of the optimization-based approach.

Second, to test and validate the developed algorithm, a vehicle and its type needed to be chosen for the simulative environment. For this purpose, the Toyota Prius 2010 was selected, a full HEV with one ICE, two electric machines connected by two planetary gear sets, resulting in a power-split configuration. A quasi-static backward-facing vehicle model was developed on the MATLAB platform for the vehicle, and its data was gathered using the available literature.

Next, to develop an intelligent learning-based novel EMS for eco-driving in CA-HEV's, a bi-layer algorithm was developed. The algorithm was divided into two levels, i.e. the upper level controller and the lower level controller. The upper level controller was responsible for finding the ideal optimal velocity profile for the trip and finding the optimal power distribution trajectories for the found velocity profile, both using the DP. Furthermore, for emulating the capabilities of the CA-HEV's, i.e.

to gather trip, road and vehicle information using the cloud-server, V2I and V2V communications, SUMO in combination with OpenStreetMaps was utilized. Once generated for a given trip for the Toyota Prius 2010, this data was then transmitted and used by the upper level controller to generate optimal results. The lower level controller was responsible for the real-time energy management of the vehicle. Hence, the learning-based algorithm was used in the form of an RL algorithm based on Q-learning for energy management of the vehicle. The training for the RL-based controller was proposed to be done on the upper level controller while the optimal Q-table parameters being transmitted to the vehicle's embedded controller for the real-time EMS of CA-HEV. In this manner, a novel bi-layer RL-based energy management strategy for eco-driving in CA-HEV's was developed.

To validate the performance of the RL-based controller, it was first trained on five optimal velocity trajectories obtained from the DP in the upper level controller and compared with the global optimal solution from DP. The training results showed that the RL-based controller could achieve charge sustaining performance and provide fuel consumption values comparable to the global optimal results, with a minimum percentage deviation of 7.76% and a maximum of 15.26%. Furthermore, as the RL-based controller was trained on the ideal optimal velocity trajectories, the controller was also tested on two approximate real-world drivecycles on the same route for understanding its effectiveness clearly. Results showed that the RL-based controller could provide charge sustaining performance and fuel consumption values greater by 17.10% and 16.52% to the global optimal results. This demonstrates that once the RL-based controller is trained on a route for some of its ideal optimal velocity trajectories, it could tackle the uncertainties present in the real-world environment that will deviate the driver from an optimal path. Moreover, it showed that the RL-based controller is robust to these changes and can provide charge sustaining and close to global optimal results in real-time applications.

## 7.2 Future Work

This section presents the potential future work that can be done on the developed novel EMS to improve its performance and extend its advantages.

### 7.2.1 High-Fidelity Modeling

The vehicle modelling presented in this thesis considers quasi-static techniques to model the behaviour of components and the vehicle. Furthermore, the vehicle dynamics are modelled in a backward-facing approach. Therefore, there is scope to improve the model's fidelity by considering the inertia effects of components and by modelling their transients. To further enhance the fidelity, the mode change dynamics can also be considered, along with temperature effects on the high voltage battery. Moreover, to accurately check the real-time capability and performance of the RL-based controller, a forward-facing Simulink-based high-fidelity vehicle dynamic model can be developed, and software-in-the-loop testing can be done.

### 7.2.2 Comprehensive Training and Sensitivity Analysis

The RL algorithm is learning-based, implying that the RL-based controller needs to be trained before it can be applied in the lower level controller. As this training is done on the cloud server of the upper level controller, there is computational power available to do extensive training of the RL-based controller so that it reaches convergence. Now, as in the current thesis, the training was done on an octa-core i7 8th Gen processor with 32Gb ram. In the future, the training can be carried out on the SHARCNET, i.e., the Shared Hierarchical Academic Research Computing Network, which is a high-performance computing environment. Enabling the high processing capability, more comprehensive training can be done, and sensitivity analysis can be performed to improve the RL-based controller's performance.

### 7.2.3 Bench marking

In the current study, to evaluate the performance of the RL-based controller, its results were compared with the DP-based controller. The DP-based controller is not applicable in real-time and can provide global optimal solutions; hence it will always provide better solutions. Therefore, other strategies like ECMS, MPC or PMP can be developed in the future, which are real-time applicable to be compared against a well-trained RL-based controller. A comparative study between different algorithms and RL-based controllers in the lower level controller for eco-driving in CA-HEV's can, in itself will be an enriching task and can form the base of a research paper.

### 7.2.4 Advanced RL-based algorithms

In this thesis, a Q-learning-based RL algorithm was utilized to be implemented in the lower level controller for real-time energy management of the CA-HEV's. The Q-learning algorithm has the advantage of being easily implementable, computationally cheap and simple action-value function estimation. However, it can only handle discretized states and cannot handle state-space in a continuous domain that can enhance the algorithm's accuracy. Hence, in the future, different learning-based strategies like Deep RL (DRL), actor-critic-based RL, and stochastic gradient-based RL can be explored in the lower level controller.

### 7.2.5 Realistic and Detailed External Environment Information

In this thesis, the cloud-server in the upper level controller was responsible for taking the information from the CA-HEV and processing them to be sent to the DP algorithm. Due to the thesis's current time and resource limitations, static road information like speed limits was only considered along with zero road grade. Although, in the real world, CA-HEV are capable of transmitting much more detailed information. Therefore, in the future, provisions can be made to extract more detailed information, including historical average road traffic information, road grade, signal phase and timing (SPaT), and driver behaviour. This can then be communicated to the upper-level controller to evaluate a more realistic optimal velocity trajectory. Thereby, incorporating more data will results in a more realistic solution by the DP, and the RL-based controller's performance can be drastically increased.

### 7.2.6 Hardware in the loop testing

Once the performance of the RL-based controller has been improved upon, by following the above-mentioned possible works, the final task that can be done to check the actual real-world performance of the controller is by implementing it in a hardware-in-the-loop simulator (Hil). The Hil simulation can guarantee the performance of the RL-based controller and can also highlight any possible error before the final step of testing the algorithm in an actual vehicle.

# References

[1] G. Wu, X. Zhang, and Z. Dong, "Powertrain architectures of electrified vehicles: Review, classification and comparison," *Journal of the Franklin Institute*, vol. 352, no. 2, pp. 425–448, 2015. Special Issue on Control and Estimation of Electrified vehicles.

[2] A. Biswas and A. Emadi, "Energy management systems for electrified powertrains: State-of-the-art review and future trends," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6453–6467, 2019.

[3] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 289–299, 2014.

[4] L. Figueiredo, I. Jesus, J. Machado, J. Ferreira, and J. Martins de Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pp. 1206–1211, 2001.

[5] J. Wu, J. Ruan, N. Zhang, and P. D. Walker, "An optimized real-time energy

management strategy for the power-split hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1194–1202, 2019.

[6] S. R., R. Schaeffer, F. Creutzig, X. Cruz-Núñez, M. D'Agosto, D. Dimitriu, M. F. Meza, L. Fulton, S. Kobayashi, O. Lah, A. McKinnon, P. Newman, M. Ouyang, J. Schauer, D. Sperling, and G. Tiwari, "Transport. in: Climate change 2014: Mitigation of climate change contribution of working group iii to the fifth assessment report of the intergovernmental panel on climate change [edenhofer, o., r. pichs-madruga, y. sokona, e. farahani, s. kadner, k. seyboth, a. adler, i. baum, s. brunner, p. eickemeier, b. kriemann, j. savolainen, s. schlömer, c. von stechow, t. zwickel and j.c. minx (eds.)] cambridge university press, cambridge, united kingdom and new york, ny, usa.," 2014.

[7] H. Gong, M. Wang, and H. Wang, "New energy vehicles in China: policies, demonstration, and progress," *Mitigation and Adaptation Strategies for Global Change*, vol. 18, pp. 207–228, February 2013.

[8] R. P. Joshi and A. P. Deshmukh, "Hybrid electric vehicles: The next generation automobile revolution," in *2006 IEEE Conference on Electric and Hybrid Vehicles*, pp. 1–6, 2006.

[9] K. Çağatay Bayindir, M. A. Gözüküçük, and A. Teke, "A comprehensive overview of hybrid electric vehicle: Powertrain configurations, powertrain control techniques and electronic control units," *Energy Conversion and Management*, vol. 52, no. 2, pp. 1305–1313, 2011.

[10] A. A. Malikopoulos, “Supervisory power management control algorithms for hybrid electric vehicles: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1869–1885, 2014.

[11] L. Serrao, S. Onori, and G. Rizzoni, “A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, 03 2011. 031012.

[12] M. Hofstetter, M. Ackerl, M. Hirz, H. Kraus, P. Karoshi, and J. Fabian, “Sensor range sensitivity of predictive energy management in plug-in hybrid vehicles,” in *2015 IEEE Conference on Control Applications (CCA)*, pp. 1925–1932, 2015.

[13] O. Servin, K. Boriboonsomsin, and M. Barth, “An energy and emissions impact evaluation of intelligent speed adaptation,” in *2006 IEEE Intelligent Transportation Systems Conference*, pp. 1257–1262, 2006.

[14] J. Zeyu, Y. Shuiping, Z. Mingduan, C. Yongqiang, and L. Yi, “Model study for intelligent transportation system with big data,” *Procedia Computer Science*, vol. 107, pp. 418–426, 2017. Advances in Information and Communication Technology: Proceedings of 7th International Congress of Information and Communication Technology (ICICT2017).

[15] M. Khanjary and S. M. Hashemi, “Route guidance systems: Review and classification,” in *2012 6th Euro American Conference on Telematics and Information Systems (EATIS)*, pp. 1–7, 2012.

[16] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy,

and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[17] E. Koenders and J. Vreeswijk, "Cooperative infrastructure," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 721–726, 2008.

[18] C. Raubitschek, N. Schütze, E. Kozlov, and B. Bäker, "Predictive driving strategies under urban conditions for reducing fuel consumption based on vehicle environment information," in *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*, pp. 13–19, 2011.

[19] C. Chan, *Overview of Electric, Hybrid, and Fuel Cell Vehicles*, pp. 1–14. American Cancer Society, 2014.

[20] M. Ehsani, Y. Gao, and J. M. Miller, "Hybrid electric vehicles: Architecture and motor drives," *Proceedings of the IEEE*, vol. 95, no. 4, pp. 719–728, 2007.

[21] A. Emadi, K. Rajashekara, S. Williamson, and S. Lukic, "Topological overview of hybrid electric and fuel cell vehicular power system architectures and configurations," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 763–770, 2005.

[22] K. Oh, J. Kim, D. Kim, D. Choi, and H. Kim, "Optimal power distribution control for parallel hybrid electric vehicles," in *IEEE International Conference on Vehicular Electronics and Safety, 2005.*, pp. 79–85, 2005.

[23] B. Conlon, "Comparative analysis of single and combined hybrid electrically variable transmission operating modes," in *SAE 2005 World Congress and Exhibition*, SAE International, apr 2005.

[24] J. Guanetti, Y. Kim, and F. Borrelli, "Control of connected and automated vehicles: State of the art and future challenges," *Annual Reviews in Control*, vol. 45, pp. 18–40, 2018.

[25] K. Jadaan, S. Zeater, and Y. Abukhalil, "Connected vehicles: An innovative transport technology," *Procedia Engineering*, vol. 187, pp. 641–648, 2017. TRANSBALTICA 2017: TRANSPORTATION SCIENCE AND TECHNOLOGY: Proceedings of the 10th International Scientific Conference, May 4–5, 2017, Vilnius Gediminas Technical University, Vilnius, Lithuania.

[26] J. J. Anaya, E. Talavera, F. Jiménez, J. G. Zato, N. Gómez, and J. E. Naranjo, "Geonetworking based v2v mesh communications over wsn," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 2421–2426, 2013.

[27] G. G. M. N. Ali, M. Noor-A-Rahim, M. A. Rahman, S. K. Samantha, P. H. J. Chong, and Y. L. Guan, "Efficient real-time coding-assisted heterogeneous data access in vehicular networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3499–3512, 2018.

[28] F. Arena and G. Pau, "An overview of vehicular communications," *Future Internet*, vol. 11, no. 2, 2019.

[29] K. A. Rahman and K. E. Tepe, "Towards a cross-layer based mac for smooth v2v and v2i communications for safety applications in dsrc/wave based systems," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 969–973, 2014.

[30] M. Sepulcre, J. Gozalvez, O. Altintas, and H. Kremo, "Context-aware heterogeneous v2i communications," in *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pp. 295–300, 2015.

[31] D. Elliott, W. Keen, and L. Miao, "Recent advances in connected and automated vehicles," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 6, no. 2, pp. 109–131, 2019.

[32] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[33] C. Yang, M. Zha, W. Wang, K. Liu, and C. Xiang, "Efficient energy management strategy for hybrid electric vehicles/plug-in hybrid electric vehicles: review and recent advances under intelligent transportation system," *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 702–711, 2020.

[34] P. Sharer, R. Leydier, and A. Rousseau, "Impact of drive cycle aggressiveness and speed on hevs fuel consumption sensitivity," in *SAE World Congress and Exhibition*, SAE International, apr 2007.

[35] H. Huo, Z. Yao, K. He, and X. Yu, "Fuel consumption rates of passenger cars in china: Labels versus real-world," *Energy Policy*, vol. 39, no. 11, pp. 7130–7135, 2011. Asian Energy Security.

[36] R. Ostadian, J. Ramoul, A. Biswas, and A. Emadi, "Intelligent energy management systems for electrified vehicles: Current status, challenges, and emerging trends," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 279–295, 2020.

[37] A. Sciarretta and L. Guzzella, "Control of hybrid electric vehicles," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 60–70, 2007.

[38] F. R. Salmasi, "Control strategies for hybrid electric vehicles: Evolution, classification, comparison, and future trends," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2393–2404, 2007.

[39] B. V. Padmarajan, A. McGordon, and P. A. Jennings, "Blended rule-based energy management for phev: System structure and strategy," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8757–8762, 2016.

[40] S. Stockar, V. Marano, M. Canova, G. Rizzoni, and L. Guzzella, "Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 2949–2962, 2011.

[41] C. G. Hochgraf, M. J. Ryan, and H. L. Wiegman, "Engine control strategy for a series hybrid electric vehicle incorporating load-leveling and computer controlled energy management," in *International Congress and Exposition*, SAE International, feb 1996.

[42] H.-D. Lee, E.-S. Koo, S.-K. Sul, J.-S. Kim, M. Kamiya, H. Ikeda, S. Shinohara, and H. Yoshida, "Torque control strategy for a parallel-hybrid vehicle using fuzzy logic," *IEEE Industry Applications Magazine*, vol. 6, no. 6, pp. 33–38, 2000.

[43] S. G. Wirasingha and A. Emadi, "Classification and review of control strategies for plug-in hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 111–122, 2011.

[44] R. Wang and S. M. Lukic, "Dynamic programming technique in hybrid electric vehicle optimization," in *2012 IEEE International Electric Vehicle Conference*, pp. 1–8, 2012.

[45] A. Brahma, Y. Guezennec, and G. Rizzoni, "Optimal energy management in series hybrid electric vehicles," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, vol. 1, pp. 60–64 vol.1, 2000.

[46] S. Onori, L. Serrao, and G. Rizzoni, *Dynamic Programming*. London: Springer London, 2016.

[47] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[48] G. Paganelli, G. Ercole, A. Brahma, Y. Guezennec, and G. Rizzoni, "General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles," *JSAE Review*, vol. 22, no. 4, pp. 511–518, 2001.

[49] *An Adaptive Algorithm for Hybrid Electric Vehicle Energy Management Based on Driving Pattern Recognition*, vol. Dynamic Systems and Control, Parts A and B of *ASME International Mechanical Engineering Congress and Exposition*, 11 2006.

[50] G. Paganelli, S. Delprat, T. Guerra, J. Rimaux, and J. Santin, "Equivalent consumption minimization strategy for parallel hybrid powertrains," in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*, vol. 4, pp. 2076–2081 vol.4, 2002.

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

[52] 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.

[53] A. Chasse, A. Sciarretta, and J. Chauvin, "Online optimal control of a parallel hybrid with costate adaptation rule," *IFAC Proceedings Volumes*, vol. 43, no. 7, pp. 99–104, 2010. 6th IFAC Symposium on Advances in Automotive Control.

[54] F. Zhang, J. Xi, and R. Langari, "An adaptive equivalent consumption minimization strategy for parallel hybrid electric vehicle based on fuzzy pi," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 460–465, 2016.

[55] H. P. Geering, *Optimal Control with Engineering Applications*. Springer-Verlag Berlin Heidelberg, 2007.

[56] Y. Zhang, L. Chu, Y. Ding, N. Xu, C. Guo, Z. Fu, L. Xu, X. Tang, and Y. Liu, "A hierarchical energy management strategy based on model predictive control for plug-in hybrid electric vehicles," *IEEE Access*, vol. 7, pp. 81612–81629, 2019.

[57] D. F. Pereira, F. d. C. Lopes, and E. H. Watanabe, "Nonlinear model predictive control for the energy management of fuel cell hybrid electric vehicles in real time," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3213–3223, 2021.

[58] J. Zhang and T. Shen, "Real-time fuel economy optimization with nonlinear mpc for phevs," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2167–2175, 2016.

[59] S. Xie, J. Peng, and H. He, "Plug-in hybrid electric bus energy management based on stochastic model predictive control," *Energy Procedia*, vol. 105, pp. 2672–2677, 2017. 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China.

[60] H. Wang, Y. Huang, A. Khajepour, H. He, and D. Cao, "A novel energy management for hybrid off-road vehicles without future driving cycles as a priori," *Energy*, vol. 133, pp. 929–940, 2017.

[61] C. Sun, S. J. Moura, X. Hu, J. K. Hedrick, and F. Sun, "Dynamic traffic feedback data enabled energy management in plug-in hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1075–1086, 2015.

[62] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[63] Q. Zhang, K. Wu, and Y. Shi, "Route planning and power management for phevs with reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4751–4762, 2020.

[64] H. Tian, S. E. Li, X. Wang, Y. Huang, and G. Tian, "Data-driven hierarchical control for online energy management of plug-in hybrid electric city bus," *Energy*, vol. 142, pp. 55–67, 2018.

[65] H. Lee, C. Song, N. Kim, and S. W. Cha, "Comparative analysis of energy management strategies for hev: Dynamic programming and reinforcement learning," *IEEE Access*, vol. 8, pp. 67112–67123, 2020.

[66] Y. Hu, W. Li, K. Xu, T. Zahid, F. Qin, and C. Li, "Energy management strategy for a hybrid electric vehicle based on deep reinforcement learning," *Applied Sciences*, vol. 8, no. 2, 2018.

[67] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[68] H. Lee, C. Kang, Y.-I. Park, N. Kim, and S. W. Cha, "Online data-driven energy management of a hybrid electric vehicle using model-based q-learning," *IEEE Access*, vol. 8, pp. 84444–84454, 2020.

[69] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[70] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[71] X. Lin, Y. Wang, P. Bogdan, N. Chang, and M. Pedram, "Reinforcement learning based power management for hybrid electric vehicles," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 33–38, 2014.

[72] T. Liu, X. Hu, S. E. Li, and D. Cao, "Reinforcement learning optimized look-ahead energy management of a parallel hybrid electric vehicle," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1497–1507, 2017.

[73] A. Biswas, P. G. Anselma, and A. Emadi, "Real-time optimal energy management of electrified powertrains with reinforcement learning," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–6, 2019.

[74] A. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110618, 2021.

[75] C. Yang, M. Zha, W. Wang, K. Liu, and C. Xiang, "Efficient energy management strategy for hybrid electric vehicles/plug-in hybrid electric vehicles: review and recent advances under intelligent transportation system," *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 702–711, 2020.

[76] F. Zhang, X. Hu, R. Langari, and D. Cao, "Energy management strategies of connected hevs and phevs: Recent progress and outlook," *Progress in Energy and Combustion Science*, vol. 73, pp. 235–256, 2019.

[77] C. M. Martinez, X. Hu, D. Cao, E. Velenis, B. Gao, and M. Wellers, "Energy management in plug-in hybrid electric vehicles: Recent progress and a

connected vehicles perspective," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4534–4549, 2017.

[78] B. HomChaudhuri, R. Lin, and P. Pisu, "Hierarchical control strategies for energy management of connected hybrid electric vehicles in urban roads," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 70–86, 2016.

[79] E. Ozatay, S. Onori, J. Wollaeger, U. Ozguner, G. Rizzoni, D. Filev, J. Michelini, and S. Di Cairano, "Cloud-based velocity profile optimization for everyday driving: A dynamic-programming-based solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2491–2505, 2014.

[80] C. Zhang and A. Vahid, "Real-time optimal control of plug-in hybrid vehicles with trip preview," in *Proceedings of the 2010 American Control Conference*, pp. 6917–6922, 2010.

[81] D. F. Opila, "Uncertain route, destination, and traffic predictions in energy management for hybrid, plug-in, and fuel-cell vehicles," in *2016 American Control Conference (ACC)*, pp. 1685–1692, 2016.

[82] Y. Regragui and N. Moussa, "Investigating the impact of real-time path planning on reducing vehicles traveling time," in *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–6, 2018.

[83] K. Bouwman, T. Pham, S. Wilkins, and T. Hofman, "Predictive energy management strategy including traffic flow data for hybrid electric vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10046–10051, 2017. 20th IFAC World

Congress.

[84] F. Zhang, J. Xi, and R. Langari, "Real-time energy management strategy based on velocity forecasts using v2v and v2i communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 416–430, 2017.

[85] M. Sivak and B. Schoettle, "Eco-driving: Strategic, tactical, and operational decisions of the driver that influence vehicle fuel economy," *Transport Policy*, vol. 22, pp. 96–99, 2012.

[86] Y. Huang, E. C. Ng, J. L. Zhou, N. C. Surawski, E. F. Chan, and G. Hong, "Eco-driving technology for sustainable road transport: A review," *Renewable and Sustainable Energy Reviews*, vol. 93, pp. 596–609, 2018.

[87] L. Guo, B. Gao, Y. Gao, and H. Chen, "Optimal energy management for hevs in eco-driving applications using bi-level mpc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2153–2162, 2017.

[88] L. Qiu, L. Qian, H. Zomorodi, and P. Pisu, "Global optimal energy management control strategies for connected four-wheel-drive hybrid electric vehicles," *IET Intelligent Transport Systems*, vol. 11, no. 5, pp. 264–272, 2017.

[89] S. Qiu, L. Qiu, L. Qian, Z. Abdollahi, and P. Pisu, "Closed-loop hierarchical control strategies for connected and autonomous hybrid electric vehicles with random errors," *IET Intelligent Transport Systems*, vol. 12, no. 10, pp. 1378–1385, 2018.

[90] M. Ghasemi and X. Song, "Powertrain energy management for autonomous hybrid electric vehicles with flexible driveline power demand," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 2229–2236, 2019.

[91] G. Ma, M. Ghasemi, and X. Song, "Integrated powertrain energy management and vehicle coordination for multiple connected hybrid electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 2893–2899, 2018.

[92] J. Hu, Y. Shao, Z. Sun, M. Wang, J. Bared, and P. Huang, "Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 228–244, 2016.

[93] D. Guo, G. Yang, X. Feng, X. Han, L. Lu, and M. Ouyang, "Physics-based fractional-order model with simplified solid phase diffusion of lithium-ion battery," *Journal of Energy Storage*, vol. 30, p. 101404, 2020.

[94] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2018.

[95] Z. Yang, D. Patil, and B. Fahimi, "Electrothermal modeling of lithium-ion batteries for electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 170–179, 2019.

[96] I. Baghdadi, O. Briat, A. Eddahech, J. Vinassa, I. Baghdadi, and P. Gyan, "Electro-thermal model of lithium-ion batteries for electrified vehicles applications," in *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pp. 1248–1252, 2015.

[97] R. Ostadian Bidgoli, "An intelligent energy management strategy framework for hybrid electric vehicles."

[98] B. Barik, P. Krishna Bhat, J. Oncken, B. Chen, J. Orlando, and D. Robinette, "Optimal velocity prediction for fuel economy improvement of connected vehicles," *IET Intelligent Transport Systems*, vol. 12, 12 2018.

[99] A. Plianos, T. Jokela, and M. Hancock, "Predictive energy optimization for connected and automated hevs," in *WCX World Congress Experience*, SAE International, apr 2018.

[100] O. Sundstrom and L. Guzzella, "A generic dynamic programming matlab function," in *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, pp. 1625–1630, 2009.

[101] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility) - an open-source traffic simulation," in *4th Middle East Symposium on Simulation and Modelling* (A. Al-Akaidi, ed.), pp. 183–187, 2002. LIDO-Berichtsjahr=2004,.

[102] A. Biswas, P. G. Anselma, A. Rathore, and A. Emadi, "Comparison of three real-time implementable energy management strategies for multi-mode electrified powertrain," in *2020 IEEE Transportation Electrification Conference Expo*

*(ITEC)*, pp. 514–519, 2020.

[103] B. Xu, F. Malmir, D. Rathod, and Z. Filipi, "Real-time reinforcement learning optimized energy management for a 48v mild hybrid electric vehicle," in *WCX SAE World Congress Experience*, SAE International, apr 2019.

[104] W. Lee, H. Jeoung, D. Park, T. Kim, H. Lee, and N. Kim, "A real-time intelligent energy management strategy for hybrid electric vehicles using reinforcement learning," *IEEE Access*, vol. 9, pp. 72759–72768, 2021.

[105] T. Liu, "Reinforcement learning-enabled intelligent energy management for hybrid electric vehicles," *Synthesis Lectures on Advances in Automotive Technology*, vol. 3, pp. 1–99, 08 2019.