A DEEP RECURRENT NEURAL NETWORK-BASED ENERGY MANAGEMENT STRATEGY FOR HYBRID ELECTRIC VEHICLES

A DEEP RECURRENT NEURAL NETWORK-BASED ENERGY MANAGEMENT STRATEGY FOR HYBRID ELECTRIC VEHICLES

BY

HELIA SADAT JAMALI OSKOEI, B.Sc.

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:	A Deep Recurrent Neural Network-Based Energy Management Strategy for Hybrid Electric Vehicles
AUTHOR:	Helia Sadat Jamali Oskoei B.Sc. (Mechanical Engineering), Sharif University Of Technology, Tehran, Iran
SUPERVISOR:	Dr. Ali Emadi
NUMBER OF PAGES:	xv, 116

To Iman and Mehdi,

and the beautiful lives who perished in the downing of flight PS752

Abstract

The automotive industry is inevitably experiencing a paradigm shift from fossil fuels to electric powertrain with significant technological breakthroughs in vehicle electrification. Emerging hybrid electric vehicles were one of the first steps towards cleaner and greener vehicles with a higher fuel economy and lower emission levels. The energy management strategy in hybrid electric vehicles determines the power flow pattern and significantly affects vehicle performance.

Therefore, in this thesis, a learning-based strategy is proposed to address the energy management problem of a hybrid electric vehicle in various driving conditions. The idea of a deep recurrent neural network-based energy management strategy is proposed, developed, and evaluated. Initially, a hybrid electric vehicle model with a rule-based supervisory controller is constructed for this case study to obtain training data for the deep recurrent neural network and to evaluate the performance of the proposed energy management strategy.

Secondly, due to its capabilities to remember historical data, a long short-term memory recurrent neural network is designed and trained to estimate the powertrain control variables from vehicle parameters. Extensive simulations are conducted to improve the model accuracy and ensure its generalization capability. Also, several hyper-parameters and structures are specifically tuned and debugged for this purpose.

The novel proposed energy management strategy takes sequential data as input to capture the characteristics of both driver and controller behaviors and improve the estimation/prediction accuracy. The energy management controller is defined as a time-series problem, and a network predictor module is implemented in the system-level controller of the hybrid electric vehicle model. According to the simulation results, the proposed strategy and prediction model demonstrated lower fuel consumption and higher accuracy compared to other learning-based energy management strategies.

Acknowledgements

This research was undertaken, in part, thanks to funding from the Canada Excellence Research Chair Program. I would like to express my sincere gratitude to my supervisor Dr. Ali Emadi, for his unwavering support, trust, encouragement, and guidance during my graduate studies. It has been truly a privilege to work with Dr. Emadi and his incredible team at McMaster Automotive Resource Center (MARC). The completion of this work would have been impossible without their help.

I am grateful to Dr. Carlos Vidal for his precious time and valuable suggestions on the development of this thesis.

I am thankful to Dr. Shamsuddeen Nalakath, Lucas Bruck, and Silvio Rotilli Filho, whom I collaborated with in various industrial and academic projects, for their valuable mentorship throughout my work and studies. I would also like to thank my colleagues Yuhang Yang and Yue Wang, who provided generous help in the initiation of this work. I want to thank all of my friends and my colleagues at MARC who made my graduate studies journey such a memorable experience.

Finally, I would like to especially thank my beloved parents, family, and my husband for their endless love, support, patience, and care throughout my entire life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivations

The automotive industry is a vital part of the global economy. Its distinctive role encompasses every aspect of the value chain, from raw materials through design and development, production, services, and disposal. Due to competitive factors and environmental forces, all of these value-creating domains are undergoing considerable innovative transformations. Among interesting trends, industry-based research shows that fuel efficiency is rapidly becoming the major priority of customers across all product-market categories. Both industry innovation and customer demand for alternative fuels and drive train technologies have been encouraged by government involvement [1].

For centuries, reciprocating piston engines were an inseparable power source component for land and water vehicles, including automobiles, motorcycles, boats, etc. An internal combustion engine (ICE) is a heat engine with intermittent combustion

cycles. The thermal efficiency of the ICE idealized thermodynamic cycle cannot exceed that of the Carnot cycle. Assuming the operation under ideal conditions, the Carnot efficiency of ICE could be approximately around 75%. In real-world conditions considering frictions and incomplete combustion, the engine efficiency is much lower. Many improvements have been made to enhance the efficiency of internal combustion engines. Peak thermal efficiency of 50% was achieved in Formula one engines by the latest technologies [2]. Besides the low efficiency of engines, serious environmental problems such as air pollution and global warming count as their most important drawbacks. According to Environmental Protection Agency (EPA), 29% of total U.S. greenhouse gas emissions in 2019 were from the transportation sector. Transportation, in fact, holds the largest share of greenhouse gas emissions [3].

While contemporary transportation relies primarily on fuel-powered cars, a worldwide contest is happening to prototype and create the next generation of vehicles. Electric, hybrid electric, and fuel cell-powered drive train technologies have long been considered as the most potential future solutions to the challenge of land transportation [4]. Different degrees of electrification are thoroughly discussed in the next chapter. Lower level powertrain electrification degrees result in hybrid electric vehicles (HEV), which take advantage of both electrical and heat energy in electric motors and internal combustion engines. Higher degrees of electrification results in plug-in HEV (PHEV) and all-electric vehicles (EV) with only battery electrical propulsion power. Figure 1.1 demonstrates the powertrain production forecast in 2030. It can be seen that the North American region is staying behind the global growth of transportation electrification with the highest percentage of ICE-only vehicles [5]. Thus, the need for research and development in this area is growing rapidly.

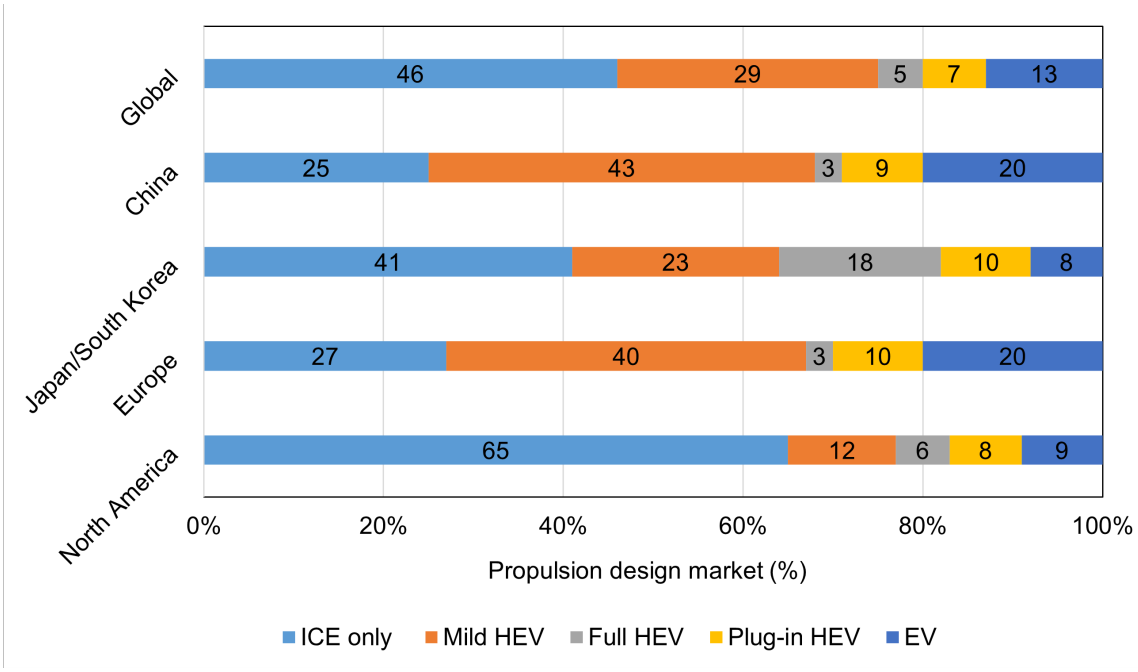


Figure 1.1: Powertrain Technology Outlook in regional and global markets in 2030(adapted from [6])

Besides improvements in ICE, powertrain electrification caused improvements in electrical subsystems of vehicles such as battery packs, electric motors, and power electronic devices. In the battery sector, there is an interesting trend towards designing the next generation of battery packs that are more powerful, lighter, and capable of fast charging. Lithium-Ion batteries are currently used in most PHEVs and EVs due to their high-to-power ratio and energy efficiency, good high-temperature performance, and low self-discharge. Nickel-Metal Hydride batteries also have been widely used in HEVs.

Permanent magnet synchronous motors and AC induction motors are two main types of electric machines used in the transportation sector. Top-selling HEV as of 2016, the Toyota Prius, has a 60 kW V-shaped interior permanent magnet traction

motor and Tesla Model S high-performance EV uses 310 kW, 600 N.m three-phase induction machine. Furthermore, researchers are working to produce more desired electric motors for traction applications, such as switched reluctance motors (SRM), which have demonstrated remarkable propulsion and manufacturing capabilities [7].

Electrified vehicles are the future of the automotive industry due to their great potential in saving fuel consumption and reducing greenhouse gas emissions. Such encouraging promise has attracted attention to the importance of power management in enhancing the fuel economy and decreasing emission levels. Also, with new data collection and transformation technologies in vehicles and with the help of artificial intelligence and machine learning algorithms, the overwhelming amount of data about roads, driving conditions, vehicle motions, and global position (GPS) can be used towards improving the energy management control strategies of electrified vehicles.

The powertrain control unit, an important part of hybrid electric vehicles, manages the power flow between different components. Various energy management strategies (EMS) employment result in different fuel economy, performance, and emissions. Most of the existing HEV control strategies are fine-tuned to obtain optimal fuel consumption under certain driving situations. Due to the difficulties in driving conditions prediction, those methods are not practical in real-world applications. Frequent change in driving behaviors and circumstances makes the condition-based control very hard to optimize a controller for every situation. As a result, control designers focus on real-time learning-based control methods such as Neural Networks (NN) and their issues for commercialization. The learning-based methods are particularly beneficial in applications where pattern recognition is difficult since they can learn from the experience provided by existing energy management control strategies to build an

optimal/sub-optimal controller [8].

This work presents an energy management strategy based on a Deep Recurrent Neural Network (RNN). The deep RNN-based strategy is specifically designed to learn various driving behaviors through an existing control strategy data in an offline framework and then use those historical data in real-time power flow control of the hybrid electric vehicle. The control strategy used to train the deep RNN in this work is similar to that used in an actual power-split HEV, Toyota Prius Model year 2010. The outline of the work and detail discussion of each part will be presented in the rest of this thesis.

1.2 Thesis Outline

Chapter 2, Hybrid Electric Powertrains, gives a brief explanation of electrification levels in electrified transportation with product examples and fuel efficiency improvements. The main focus is on hybrid electric vehicles powertrain components and their various architectures. Four popular architectures of mild, series, parallel, and power-split are demonstrated. Then three major powertrain supervisory control objectives of fuel economy, emissions and greenhouse gasses, and performance are presented. These are among the main objectives of this thesis which are explained through statistics and literature review.

Chapter 3, Fundamentals of Energy Management Strategies, presents a classification for energy management strategies in HEV powertrain, which divides EMSs into three groups: optimization-based, rule-based, and learning-based. Then, a brief overview of a few popular online and offline energy management of each classified

group is given. Deterministic and fuzzy rule-based, dynamic programming, equivalent consumption minimization strategy, model predictive control, and neural networks and machine learning strategies are evaluated for their benefits and drawbacks. In the end, a comparison is made to wrap up the review on various EMSs.

In chapter 4, Hybrid Electric Vehicle Modeling, the model for Toyota Prius model year 2010, which is a power-split HEV, is explained thoroughly. The model is developed in MATLAB/Simulink software using actual vehicle dynamometer test data. Model parameters are obtained from comparing model outputs to test data. The model has three major subsystems: Driver, supervisory controller, and vehicle models. Driver speed and torque demand are simulated using driving cycle input to a PID controller. The vehicle model consists of minor powertrain and chassis subsystems: internal combustion engine, motor/generators and power electronics, power-split device, battery, wheel, final drive, and chassis models. Each subsystem modeling is discussed briefly with relevant equations. A rule-based EMS is developed for the supervisory controller based on the vehicle behavior under four different driving cycles available in test data. Model Evaluations at the end show adequate model fidelity for investigating the impact of using other EMSs on fuel economy and vehicle performance.

In chapter 5, Offline LSTM Multi-Parameter Prediction Model, the process of developing a Long short-term memory (LSTM) model from the HEV model and controller outputs is presented. An introduction on the reasons for selecting recurrent neural networks and specifically LSTM networks for this work are highlighted. The LSTM network has trained on four input data of vehicle speed, acceleration, battery state of charge, and torque demand. It predicts the control variables of engine speed

and engine torque. A pre-processing is done on the training dataset, including normalization with standard score method before training the network. After a brief introduction on LSTM networks, the methodology for developing the LSTM multi-parameter prediction model is outlined into four major steps: Define, Compile & fit, Verify, and Predict. Evaluation metrics for choosing the best model performance are then described. At least 45 networks with different structures and hyper-parameters are simulated in the next step to find the optimal model and parameters. Ten network structures with different layers and neurons, three sizes of the testing dataset, four learn rate drop factors, five mini-batch sizes, and seven time window sizes are evaluated. The final model achieved 1.41 (rad/s), and 0.57 (N.m) validation root mean square error (RMSE) for engine speed and torque, respectively.

In chapter 6, Deep Recurrent Neural Network Based Energy Management Strategy, the developed LSTM model from the previous chapter is implemented online in the selected HEV model. The network predictor module takes the historical data from the model in real-time. It predicts the engine speed and torque variables which can be used in supervisory control of the HEV. For this purpose, a time-series interpretation of EMS is given to elaborate on using the LSTM prediction model as an online energy management controller. The integration of the LSTM prediction model in the HEV Simulink model is discussed, and the results on tested drive cycles have shown improvements in fuel economy over the initially developed rule-based strategy.

Chapter 2

Hybrid Electric Powertrains

2.1 Introduction

With significant technological breakthroughs in vehicle electrification, the future of the automobile industry is inevitably experiencing a paradigm change from fossil fuels to the electrified powertrain. HEVs (Hybrid Electric Vehicles), plug-in hybrid electric vehicles (PHEVs), and fuel-cell vehicles are the first step toward a more energy-efficient future [9]. Hybrid electric vehicles use multiple sources of energy and power in their system. Battery, internal combustion engine, fuel cell, ultracapacitor, etc., are some of the energy sources used primarily on HEVs [10]. In this chapter, the main focus is on hybrid electric powertrains and explaining the electrification level and their powertrain architecture. We specifically consider a hybrid electric powertrain consisting of an internal combustion engine, electric motors, power electronics units, and a battery pack.

2.2 Powertrain Electrification and Architecture

2.2.1 Electrification Level

The capacity of the electric path is determined by the electrification level, which is generally denoted by battery voltage, power, and performance of xEVs. Micro hybrid, mild hybrid, strong hybrid, plug-in hybrid, and fully-electric powertrains are the most common types of electrified powertrains [11]. The electrification level defines the ratio of electrical power available to the total power. Nowadays, most commercial vehicles have at least 10 % -20 % electrification [12].

As listed in table 2.1, HEVs have varying degrees of electrification, ranging from conventional vehicles to fully electric vehicles. The degree of electrification contributes directly to fuel economy improvement and CO_2 emission reduction. Also, it is highly dependant on the powertrain architecture, which is discussed in the following subsection.

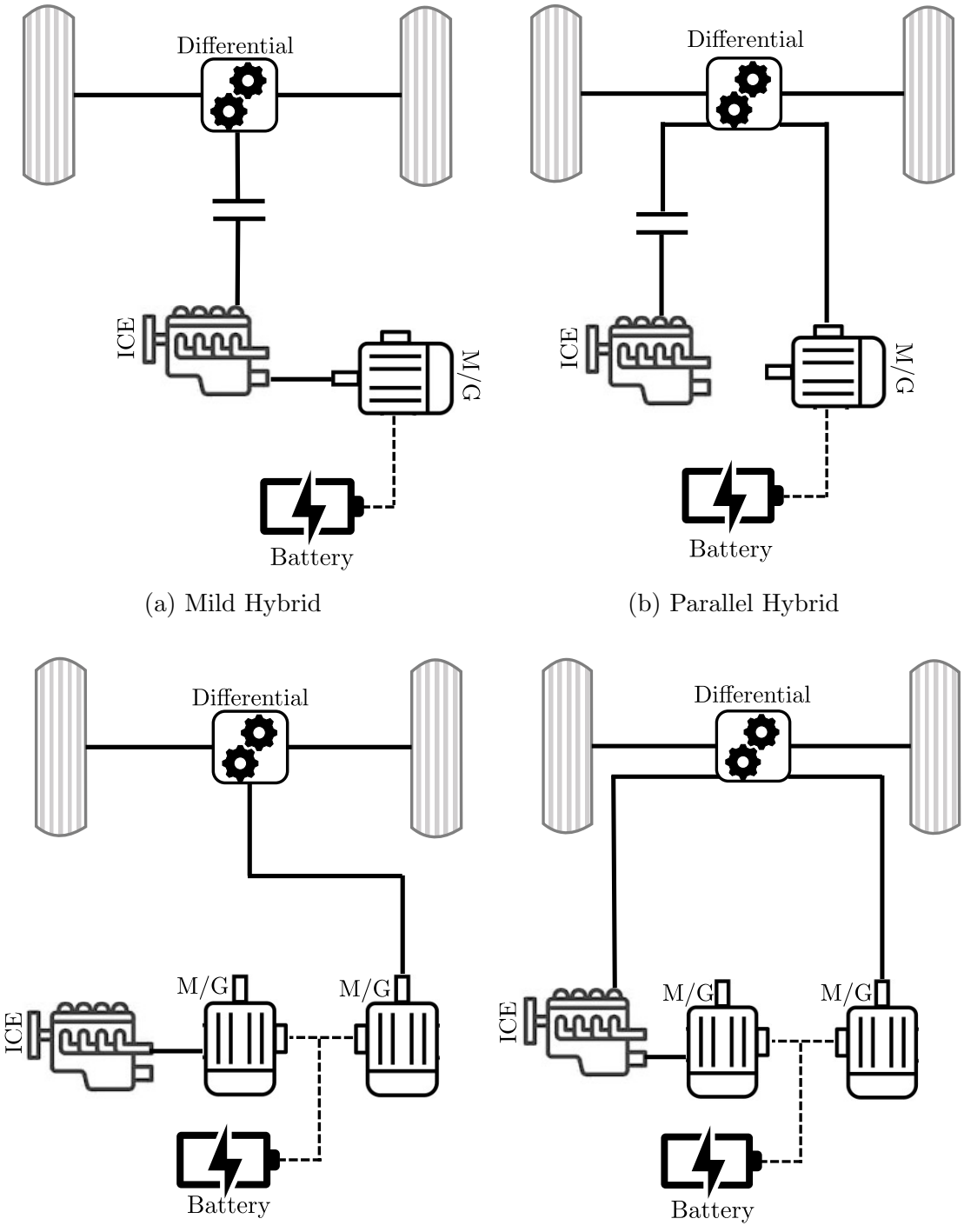
2.2.2 Powertrain Architecture

The architecture of a hybrid electric powertrain is dependant on the electrification level and the connection between drive train components such as internal combustion engine (ICE), electric motor or generator (M/G), battery pack, transmission, and differential (final drive). Conventionally, HEVs were classified into two simple architectures of series and parallel. After 2000, some new HEVs, such as Toyota Prius, Chevrolet Volt, and Fiat-Chrysler Pacifica, could not be classified into such conventional categories [4]. Their powertrain architecture was a combination of both classes called series-parallel, which benefits from the advantages of both series and parallel

Electrification	Classification	Electric motor power range	Fuel efficiency improvement	Examples
∼0-5%	Micro hybrid	3-10 kW	2-10%	Volkswagon Passat BlueMotion
∼10-15%	Mild hybrid	8-20 kW	8-20%	Mercedes benz S400 Honda Acura ILX Hybrid
∼20-40%	Parallel hybrid	20-50 kW	20-50%	Hyundai Sonata 2016 BMW activehybrid 5
	Series-parallel			Toyota Prius Chevrolet volt Cadillac Escalade
∼40-50%	Series hybrid			BMW i3 Nissan e-Power NISMO
∼50-70%	Plug-in hybrid	30-80 kW	40-100%	Toyota Prius 2016 Chevrolet Volt PHEV
100%	Fully electric (EV)	100 kW	100%	Tesla Model S Nissan Leaf Chevrolet Bolt EV

Table 2.1: Degree of electrification and classification of electrified powertrains: fuel efficiency improvement and electric traction motor power (adapted from [13, 14])

architectures and eliminates some disadvantages in both typologies [15]. The notion of series-parallel architecture can be achieved with the help of multiple clutches or a power-split device (PSD) [16]. Figure 2.1 demonstrates the arrangement of drive train components in the architectures discussed above. Note that solid and dashed lines indicate mechanical and electrical links, respectively. Also, ICE denotes engine, and M/G denotes motor/generator. In this work, the main focus will be on power-split architecture as they are the most advanced topology. Additionally, this architecture offers better fuel economy and performance.



(a) Mild Hybrid

(b) Parallel Hybrid

(c) Series Hybrid

(d) Series-Parallel Hybrid

Figure 2.1: Various types of hybrid powertrain architectures.

2.3 Powertrain Supervisory Control Objectives

The control of the power flow between fuel and electrical energy sources that leads to vehicle propulsion is the key problem in the development of HEVs. The problem emerges as a result of limited electrical energy source and the need to reduce fuel usage and emissions. The selected energy management strategy (EMS) significantly impacts the vehicle's performance, fuel economy, and exhaust emissions, which is the main focus herein [17]. The supervisory control for managing energy flow from powertrain components to the wheels in series-parallel configuration is more complicated than other topologies because more energy sources need to operate at optimal or near-optimal points [14].

2.3.1 Fuel economy

The oil and fuel resources are limited, and the amount of years Earth's oil resources can sustain our oil supply is entirely dependent on the discovery of fresh oil reserves and cumulative oil production, which is highly difficult and expensive. Based on this and the fact that the transportation section is one of the primary consumers of petroleum, improving the vehicle's fuel economy significantly influences the supply of oil [4]. Also, it has been predicted that global fossil fuels production will be maximized around the 2030s and 2040s, after which it would gradually decline [18]. Based on the U.S. Environmental Protection Agency (EPA) report, fuel economy has increased 29% or 5.6 mpg, since 2004 [19]. The EPA has devised a regulation that will significantly improve the fuel economy of new automobiles to ensure that automakers invest in fuel-saving technologies. The corporate average fuel economy (CAFE) standard requires passenger vehicles to achieve a minimum fuel economy of

43.1 mpg by 2026 [20].

Moving toward electrifying vehicles is one of the most viable solutions to meet the fuel economy standards in the future. Figure 2.2 shows the fuel economy for different types of vehicles from 1975 to 2020. As it can be seen, the trend is toward higher fuel economy [21].

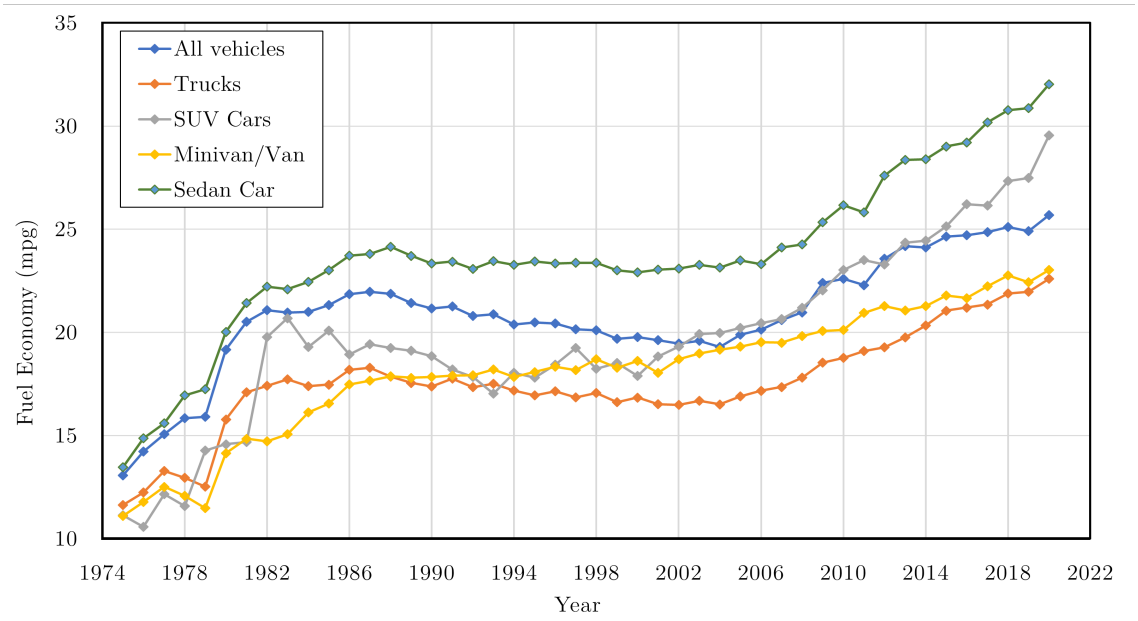


Figure 2.2: Estimated real-world fuel economy (adapted from [21])

2.3.2 Emissions and Greenhouse Gasses

The products of non-ideal hydrocarbon fuels combustion contain carbon dioxide (CO_2), nitrogen oxides (NO_x), unburned hydrocarbons (HC), and carbon monoxide (CO), which are harmful both to the environment and public health. From 1980 to 1999, 32% of carbon dioxide, which is one of the greenhouse gasses (GHG), has been emitted from the transportation sector resulting in global warming [4]. In most countries, an emission performance standard has been set to regulate greenhouse

gasses and exhaust emissions. For instance, in the US, EPA manages the emission standards. However, the state of California is allowed to set stricter standards due to severe air quality issues, which is set by California Air Resources Board (CARB) [22]. EPA and the National Highway Traffic Safety Administration (NHTSA) have issued final rules to reduce GHG emissions for light-duty vehicles from 2017 to 2025. In model year 2025, the final requirements are expected to result in an average fleetwide CO_2 level of 163 g/mi [23]. Based on the report published by EPA, since 2004, CO_2 emission rates have decreased 23% or 105 g/mi [19]. Figure 2.3 demonstrates the downward trend of real-world CO_2 emissions for the past 25 years.

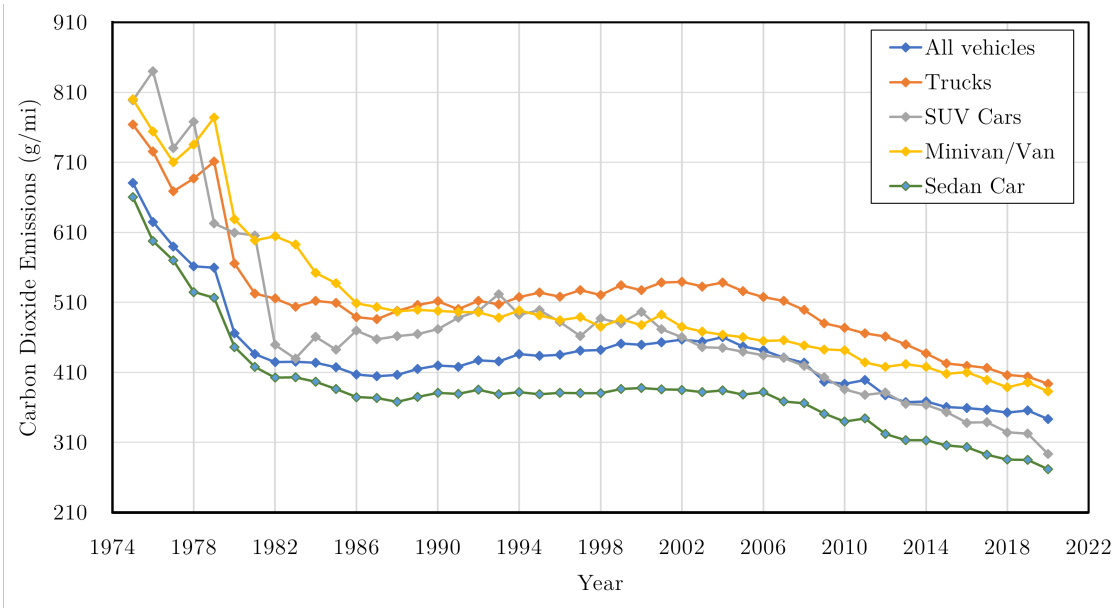


Figure 2.3: Estimated real-world CO_2 emissions (adapted from [21])

Large automotive manufacturing companies have set goals towards zero emissions. Stellantis targets over 70% of Europe sales to be low emission vehicles by 2030 [24]. Also, General Motors has aimed for an all-electric future with zero emissions by 2030 in the US market [25].

2.3.3 Performance

Besides the advantages of powertrains electrification in fuel economy and emission reduction, the driving performance of electrified vehicles is an important factor that should be discussed. Combining the HEVs available electric power with ICE power to create exceptional vehicle drivability is a tremendous opportunity. The acceleration time, maximum cruising speed, gradability, braking, and stability of a vehicle are generally used to evaluate its driving performance.

There are numerous techniques to improve the performance of internal combustion engines. One of them that is directly related to increasing the torque generated by ICE, is the 'forced induction', which simply increases the amount of air induced in cylinders. Forced induction can be done by variable intake manifold, supercharging, or turbocharging. Turbocharging can highly increase the power output of the engine. Also, the dynamic performance can be significantly improved by directly attaching an electric motor to the turboshaft [26]. Slow response time and low effectiveness at low engine speed are the most important disadvantages of turbocharging in engine performance [4]. The transient response time of the turbocharger can be improved by using an auxiliary electric motor [26].

The key factors in EV drive train design are the appropriate motor power rating and transmission characteristics to satisfy the performance criteria. Electric motors have an entirely different torque profile than internal combustion engines in which they can achieve their maximum torque at extremely low speeds. ICEs, on the other hand, typically require at least 33% of their maximum engine speed to achieve peak torque [27]. Based on this characteristic of electric motors, they can make up for the lack of available torque from the ICE, which improves the traction and propulsion

performance of HEVs.

Additionally, Regenerative braking reduces the amount of pressure applied to the brakes and recovers free energy in braking intervals, which can subsequently be used as an electric boost.

Chapter 3

Fundamentals of Energy Management Strategies

3.1 Introduction

Several factors are impacting the performance of HEVs and PHEVs from a system design standpoint, including the powertrain type, architecture, and energy management strategy (EMS). To achieve an energy-efficient powertrain, numerous control objectives, as discussed in section 2.3 , must be addressed, making the EMS one of the most critical parts of the powertrain system design [28].

EMS is responsible for deciding which energy source (e.g., battery pack, gasoline fuel) and propulsion unit (e.g., electric motor, ICE) must be used to provide the demanded power from the driver while meeting its objectives. Different control strategies would have various fuel economy, emissions, and drivability results. Thus, the EMS actions should be feasible to implement and be towards achieving near-optimal results.

Due to the importance of the EMS in HEV powertrain, it is essential to review different types of EMS. In [14], Biswas and Emadi presented a similar classification by dividing EMSs into three groups of premediated, casual, and blended EMS. Enang and Bannister in [29] classified HEV control strategies into online and offline energy management strategies. Figure 3.1 is demonstrating one of the many classifications represented for energy management strategies [28], classified EMSs into optimization-based, rule-based, and learning-based strategies. Some of these strategies are equivalent consumption minimization strategy (ECMS), model predictive control (MPC), Pontryagin's minimum principle (PMP).

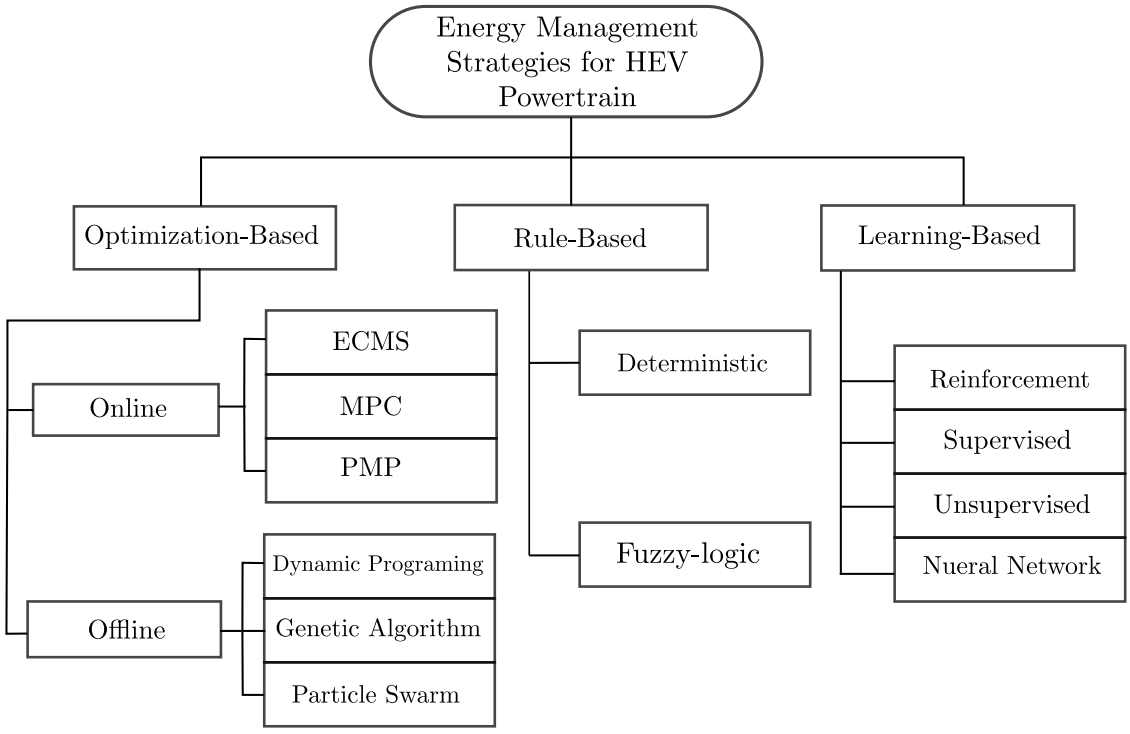


Figure 3.1: Classification of energy management strategies for HEV powertrains

In this chapter, a brief overview of a few popular energy management strategies

will be presented.

3.2 Deterministic and Fuzzy Rule-Based

Rule-based (RB) EMSs, as it comes from their name, rely on a set of predetermined rules based on intuitions, heuristics, and human skills without the need for a priori knowledge of the driving cycle. This kind of control strategy has the advantage of real-time implementation feasibility due to its simplicity and efficiency. However, RB strategies do not offer an optimal solution [28]. RB strategies divide into deterministic and fuzzy.

Deterministic strategies such as thermostat (on/off), power follower, and state machine-based strategies [30], lack the capacity to deal with uncertainties caused by model inaccuracy, as well as flexibility for various driving cycles, due to the fixed rules [31].

Fuzzy RB strategies differ from deterministic RB in being more robust to noises from measurement tools and component variability [32]. Thus, fuzzy RB strategies are particularly well suited to multi-domain, nonlinear, and time-varying systems [33]. The main structure is still similar to deterministic strategies with sets of if-then rules, including membership functions for possibly improved performance [34].

3.3 Dynamic Programming

The dynamic programming approach was created by Richard Bellman and sought to identify optimal control policies through a multi-stage decision process. This approach is one of the popular and widely used algorithms for the EMS offline optimization

problems [29]. This control strategy requires a priori knowledge of the future driving cycle, and the optimal cost-to-go function $J()$ is then formulated, which is expressed in the following equation:

$$J(x(t),t) = \min_{u(t)\in\mathcal{U}} \left(J(x(t+\Delta t), t+\Delta t) + \int_t^{t+\Delta t} g(x(\tau), u(\tau), \tau) d\tau \right) \tag{3.3.1}$$

Note that $u(t)$ and $x(t)$ are control inputs and state variables, respectively. Also, $\mathcal{U}$ is the set of all potential control inputs. A numerical-based DP method should be utilized to solve this finite optimization problem. By discretization of time in the format of $k \in [1, 2, ..., M]$ and discretization of states in the format of $i \in [1, 2, ..., N]$, previous equation can be written as:

$$J^*\left(x(k), k\right) = \min_{u(k)\in\mathcal{U}(k)} \left\{ J^*\left(x(k+1), k+1\right) + g\left(x(k), u(k), k\right) \right\} \tag{3.3.2}$$

Moreover, the model of the HEV in discrete-time format can be stated as

$$x(k+1) = f(x(k), u(k)) \tag{3.3.3}$$

$g(x, u)$ in equation 3.3.2represents a linear/nonlinear dynamical system with any cost function and any limitations on u and x. Practically, the discretization of numerical implementation is critical to the optimality of final results. Hence as the problem complexity increases, the computing time grows dramatically. In brief, there are two setbacks that DP suffers from, reliance on prior knowledge of driving cycle and heavy computational burden dependant on dimensionality. Therefore, DP control results are only helpful as optimal benchmarks for other controllers or foundations

for developing and improving other sub-optimal controllers [29, 31, 35].

3.4 Equivalent Consumption Minimization Strategy

Paganelli et al. [36] developed the equivalent consumption minimization strategy (ECMS) for parallel HEVs functioning in a charge-sustaining mode. ECMS, despite DP, does not require priori knowledge of driving cycle.In this strategy, an equivalency factor (EQF), depending on the average efficiency of the battery during charge/discharge, along with the efficiencies of electric motors and other components, transforms electrical power consumption into fuel consumption [37]. The total of actual fuel consumption $\dot{m}_f$ and battery fuel consumption $\dot{m}_{f,batt,eq}$ is known as equivalent fuel consumption, expressed in the below equation

$$\dot{m}_{f,eq} = \dot{m}_f + \dot{m}_{f,batt,eq} \tag{3.4.1}$$

The idea is to minimize the instantaneous cost, which is stated in the equation below

$$J^* = \arg\min_{u \in \mathcal{U}} \dot{m}_{f,eq}(u) \tag{3.4.2}$$

More operation metrics such as emissions can be added to the instantaneous cost function with tuning factors. This approach will achieve locally optimized results, and it has the potential for online implementation. The definition of equivalence factors is crucial in ECMS. That is due to the change of EQFs depending on the driving conditions [38]. Therefore, variations of ECMS, such as Adaptive ECMS and

Telemetry ECMS, have been evolved to adjust the equivalence factor based on driving data and future predictions [39, 40].

3.5 Model Predictive Control

The MPC is a popular and commonly used approach in HEVs for dealing with multi-variable restricted control problems. This strategy can be thought of as a compromise between DP and ECMS [41, 42]. The MPC algorithm is usually implemented in three iterative steps: (i) predicting the outputs based on the optimization horizon by using the plant model; (ii) determining the cost function for the future outputs of the plant; and (iii) implementing the control policy with minimum cost [34]. Figure 3.2 shows these steps in a flow diagram.

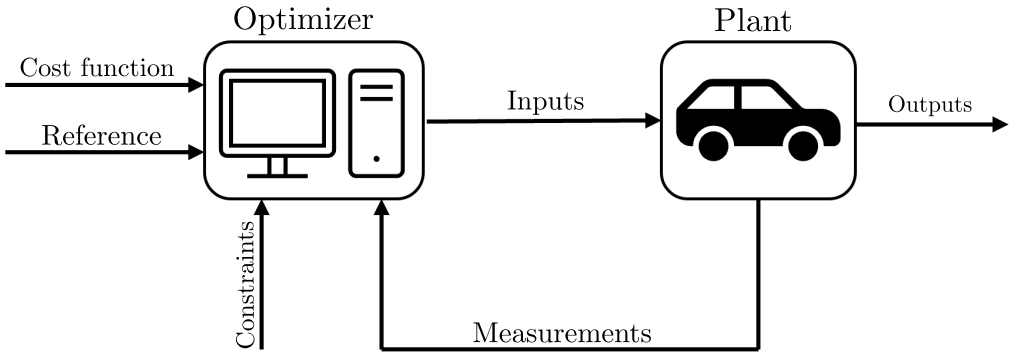


Figure 3.2: Flow diagram of MPC algorithm

In classical MPC, which indicates control problems involving linear time-invariant (LTI) systems, the system dynamics are described in the equations below without

considering any uncertainties.

$$x(k+1) = Ax(k) + Bu(k) \tag{3.5.1a}$$

$$y(k) = Cx(k) \tag{3.5.1b}$$

Where $x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, y(k) \in \mathcal{Y}$ are the system state, control input and system output, respectively. The index $k = 0, 1, ..., N-1$ is indicating the discrete time, where N is the controller horizon. The controller is required to minimize $\sum_{k=0}^{N-1} g(x(k), u(k))$ subject to the constraints. Also, it is assumed that the state of the model in equation 3.5.1 is stable and observable to ensure that the optimal cost value is properly defined [43].

3.6 Neural Networks and Machine Learning

Artificial Intelligent (AI) related EMSs are able to solve complicated problems and do intelligent data processing and provide intelligent control. Artificial Neural Networks (ANN), Deep Learning (DL), and Q-Learning (QL) are examples of this kind of EMSs [44]. An artificial neural network (ANN) is a computing system made up of several basic, highly linked processing units that process data by changing their state in response to external inputs. By changing weights to minimize the error between the actual and projected output patterns of a training set, neural networks may be trained to learn a highly non-linear input/output relationship [45].

Some researchers used neural networks to predict and recognize the driving patterns based on typical driving patterns and then employed them in other EMSs to obtain an intelligent EMS [46]. In other cases, [47] the neural network architecture

has been enhanced to accommodate additional multi-objectives such as fuel efficiency and torque distribution optimization. Recurrent Neural Network (RNN) algorithms are also employed for HEV EMS or batteries state-of-charge (SOC) estimation due to their ability to model a time collection. In [48], an NN-controller has been proposed for the Toyota Prius based on RNN using online and offline training.

Another learning-based strategy that has drawn researchers' attention is the reinforcement-Learning (RL) algorithm-based control strategy. The RL algorithm is a method for learning from interactions. Through a numerical reward or punishment signal, the goal to make the best decision possible can be achieved by trial and error [49].

In this work, using Long Short-Term Memory (LSTM) network, which is a deep RNN-controller is the main approach, and it will be discussed more in the next chapters.

3.7 Comparison

In conclusion of this chapter and the literature review on a few of the popular EMSs, a brief comparison along with advantages and disadvantages of the aforementioned strategies will be presented. Figure 3.3 illustrates the comparison between rule-based (RB), learning-based (LB) (including AI-related strategies), equivalent fuel consumption minimization (ECMS), model predictive control (MPC), and dynamic programming (DP) in respect of optimality, real-time implementation capability, and control horizon. Tran et al. in [28] investigated a comparative analysis of these strategies.

The main focus in the rest of the chapters will be on rule-based and AI-related energy management approaches.

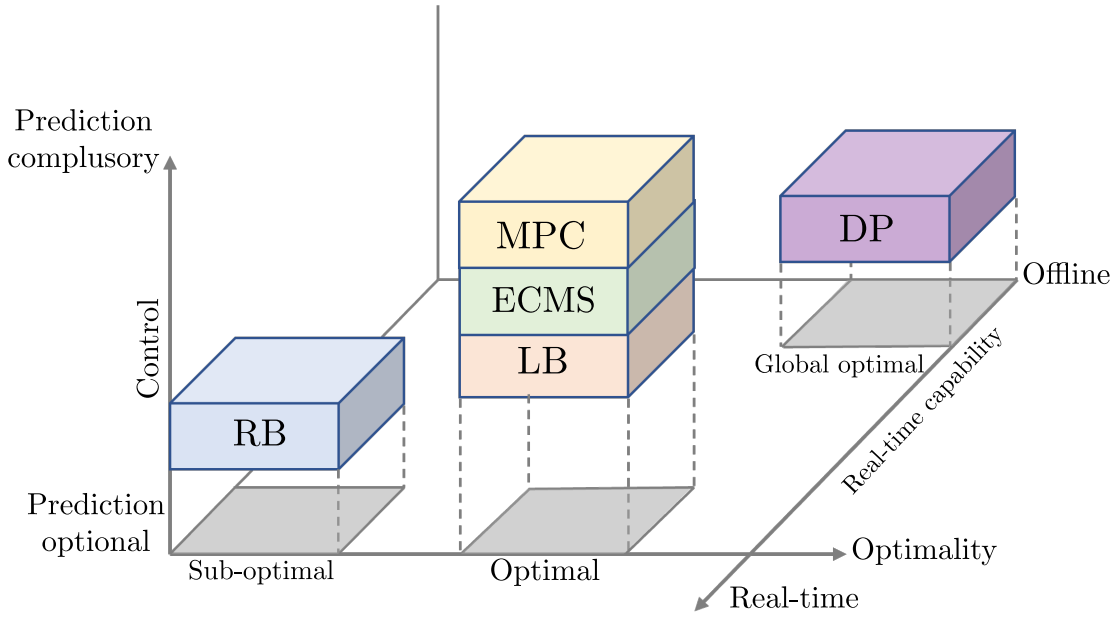


Figure 3.3: Comparison of studied EMSs. (adapted from [28])

a

Chapter 4

Hybrid Electric Vehicle Modeling

4.1 Introduction

The first step in developing any control strategy or energy management system for an HEV is developing its model based on available data and engineering assumptions. Also, detailed design is one of the important steps in the V-model of software developments after requirement analysis, and it makes the basis for integration, testing, and verification of the software [50]. Building a high-fidelity model for the selected HEV gives us a more clear idea of the performance and behavior of the vehicle and each component under different driving situations.

The targeted modeled HEV is Toyota Prius model year 2010- 3rd generation of Prius models, which has a series-parallel powertrain configuration with a power-split device to control the energy flow between its Atkinson-cycle engine and two permanent magnet motors. The model has been developed in MATLAB/Simulink. The goal is to create a system-level model that can simulate power flow between components in a real vehicle.

The model structure, as shown in figure 4.1 consists of three main blocks: Driver, system-level controller, and vehicle model. As it can be seen, the drive cycle information, which is the vehicle velocity $v(t)$ from t_{start} to t_{end} is provided for the driver. This approach gives the ability to compare the model performance under various conditions and drive cycles. Figure 4.2 shows a sample drive cycle of urban dynamometer driving schedule (UDDS). The driver uses the input from the driving cycle to calculate the required torque needed from the vehicle to follow the drive cycle velocity. The driver requested torque is then given to the controller to make a decision on how to split the torque and power between sources of energy and thus make commands for powertrain components. In the last step, the torque and power command fed to the drivetrain and then simulated the vehicle acceleration/deceleration through the wheel model.

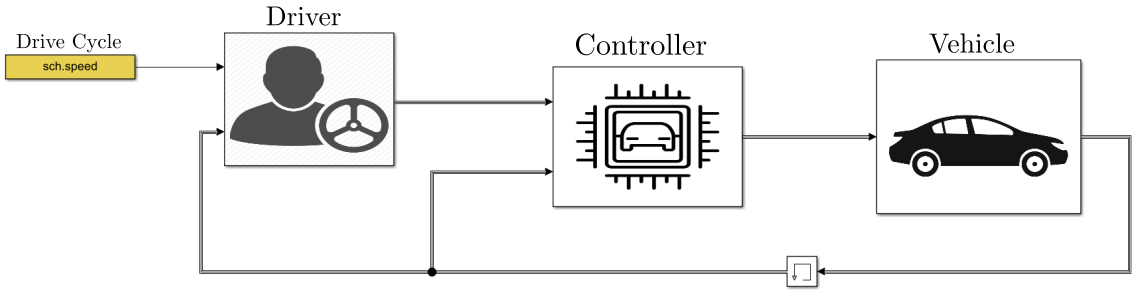


Figure 4.1: Model structure in Simulink: driver, controller, vehicle blocks

In the development of the model, downloadable dynamometer database testing results for the 2010 Toyota Prius are used, which is available from the Argonne National Laboratory database, [51] to improve the fidelity of the controller and vehicle

model and to obtain some parameters. Moreover, the data for the vehicle model is obtained from technical reports, publications, and vehicle specifications documents.

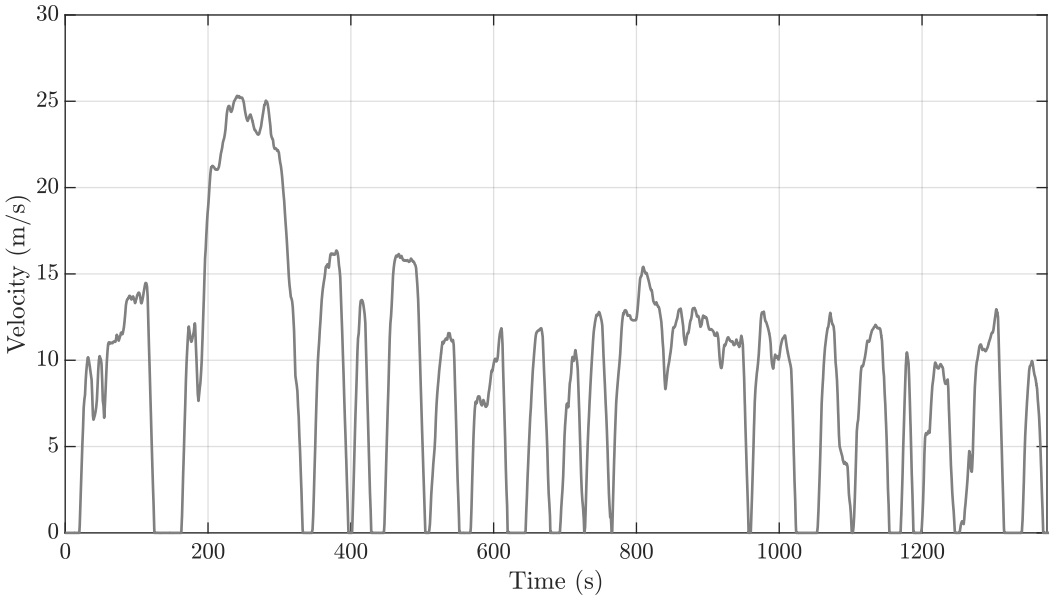


Figure 4.2: Urban Dynamometer Driving Schedule (UDDS)

4.2 Vehicle Model

The selected vehicle, as mentioned before, has an Atkinson-cycle internal combustion engine (ICE) which propels the vehicle by converting the chemical energy in fuel (gasoline) to mechanical energy, and it is connected to motor/generator 1 (MG1) through the planetary gearset 1 in the power-split device (PSD). Excessive engine power goes to MG1, which acts mostly as a generator and charges the battery pack. Also, MG1 helps in the start/stop of the engine by acting as a motor. Motor/ generator 2 (MG2) is connected to the planetary gearset 2 of the PSD, and it propels the vehicle by using battery energy and converts it to mechanical energy. Due to the

higher power ratings of MG2, it mostly acts as a motor, but it is also responsible for regenerative braking. Energy storage system (ESS) is a fan-cooled high-voltage Nickle-metal Hydride (NI-MH) battery pack that can be charged or discharges by motor/generators through power electronic devices (power inverter/converter). Also, electrical accessories such as headlights and air conditioner (AC) use battery power for their operation. The schematic of the drivetrain and PSD has shown in figure 4.3. Parameters of drivetrain components are listed in Table 4.1 [52–54]. Also, the Simulink model layout for the vehicle is captured in figure 4.4 and each component of the model is going to be explained in detail in the next sections.

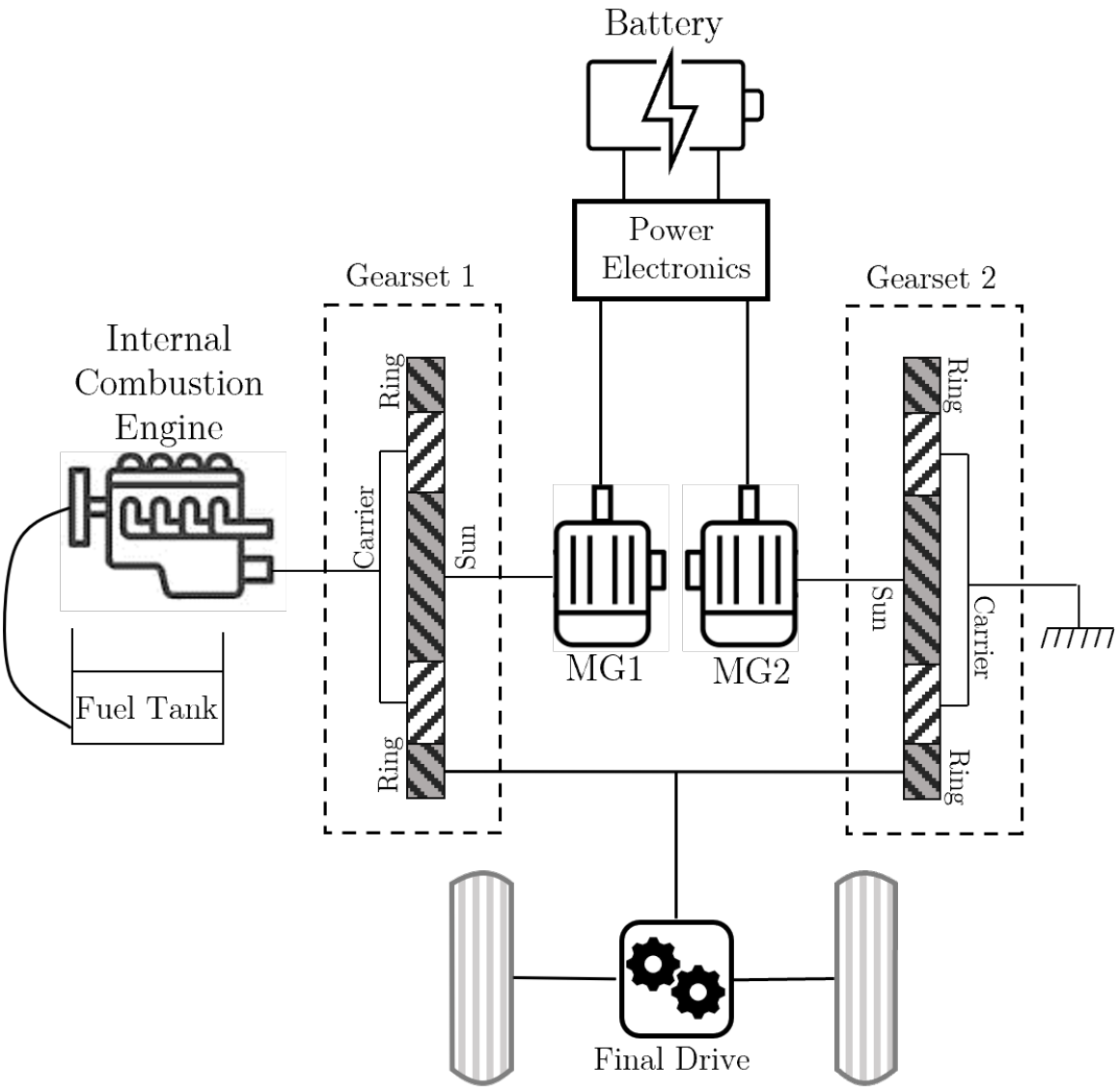


Figure 4.3: Drivetrain and power-split device configuration (figure not drawn to scale)

Component	Parameters	Symbol	Value
Engine (ICE)	Max. Power	$P_{ICE,max}$	73 kW
	Max. Torque	$T_{ICE,max}$	142 $N.m$
	Moment of inertia	J_{ICE}	0.18 $kg.m^2$
Motor/Generator 1 (MG1)	Max. Power	$P_{MG1,max}$	42 kW
	Max. Torque	$T_{MG1,max}$	45 $N.m$
	Max. Speed	$\omega_{MG1,max}$	10000 rpm
	Moment of inertia	J_{MG1}	0.02 $kg.m^2$
Motor/Generator 2 (MG2)	Max. Power	$P_{MG2,max}$	60 kW
	Max. Torque	$T_{MG2,max}$	207 $N.m$
	Max. Speed	$\omega_{MG2,max}$	13500 rpm
	Moment of inertia	J_{MG2}	0.05 $kg.m^2$
Battery	Rated Capacity	C_{batt}	6.5 Ah
	Nominal voltage	$V_{batt,nom}$	201.6 V
	Rated Power	$P_{batt,max}$	27 kW
Power-split device	Gearset 1 ratio	n_1	78:30
	Gearset 2 ratio	n_2	58:22
	Efficiency	η_{psd}	0.95
Final Drive	Gear ratio	n_{fd}	3.268
	Moment of inertia	J_{fd}	0.1 $kg.m^2$
Wheel	Radius	r_{wh}	0.31 m
	Efficiency	η_{wh}	0.98
	Moment of inertia	J_{wh}	0.8 $kg.m^2$

Table 4.1: Vehicle components parameters

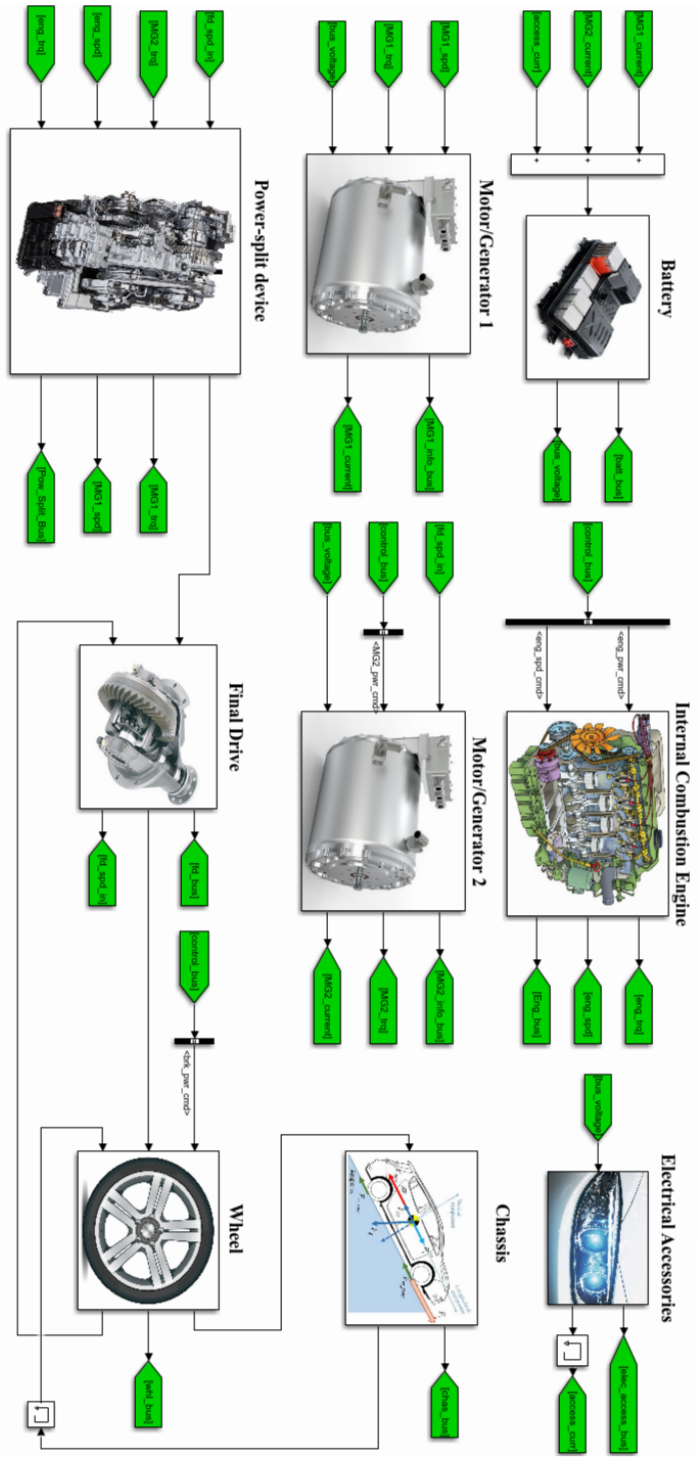


Figure 4.4: Vehicle components layout in Simulink including MG1, MG2, ICE, battery, electrical accessories, final drive, chassis, wheel, and PSD

4.2.1 Vehicle Dynamics

In order to have sufficient model accuracy, a longitudinal vehicle dynamics mode is necessary. In this regard, the chassis (vehicle body) model considers aerodynamic, and road grade losses and the wheel model accounts for rolling resistance losses. Table 4.2shows a summary of required parameters for the calculation of mentioned losses.

Parameter	Symbol	Value
Total vehicle mass	m	1460 kg
Equivalent vehicle mass	m_{eq}	1565kg
Frontal area	A	1.746 m^2
Air drag coefficient	c_d	0.25
Air density	ρ	1.225 kg/m^2
Rolling resistance	μ_1	0.002
	μ_2	0.0002
Gravity constant	g	9.81 m/s^2

Table 4.2: Vehicle specifications and longitudinal dynamics parameters

For road grade losses, the equation for gravitational drag force can be written as

$$F_{grade} = mg\sin(\theta) \tag{4.2.1}$$

where θ is the road grade. For rolling resistance force, the following equation is considered

$$F_{roll} = (\mu_1 + \mu_2.v)mg\cos(\theta) \tag{4.2.2}$$

where v is the vehicle speed. And lastly, the aerodynamic drag force can be calculated

from

$$F_{aero} = \frac{1}{2}\rho c_d A v^2 \tag{4.2.3}$$

Thus, the total loss force and the wheel force which is propelling the vehicle are equal to

$$F_{tot,loss} = F_{aero} + F_{roll} + F_{grade} \tag{4.2.4}$$

$$F_{wh} = m_{eq}\frac{dv}{dt} - F_{tot,loss} \tag{4.2.5}$$

where the equivalent vehicle mass is the sum of vehicle mass and total moment of inertia of rotational components, and it can be stated as

$$\begin{aligned} J_{\text{total}} &= \left(n_1^2 J_{MG1} + \left(\frac{n_1}{1+n_1} \right)^2 J_{ICE} + n_2^2 J_{MG2} \right) n_{fd}^2 + J_{fd} + 4J_{wh} \\ m_{eq} &= m + \frac{J_{\text{total}}}{r_{wh}^2} \end{aligned} \tag{4.2.6}$$

Figure 4.5 shows a free-body diagram of the longitudinal dynamics of a vehicle on a graded surface.

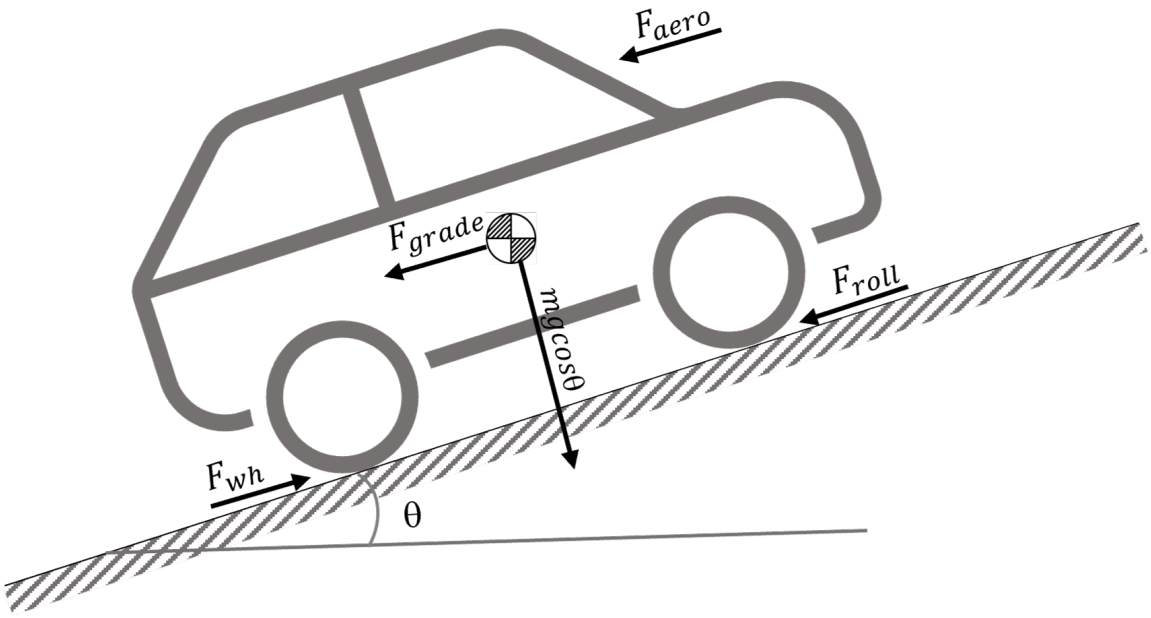


Figure 4.5: Vehicle longitudinal dynamics free-body diagram

4.2.2 Internal Combustion Engine Model

The power requirement from the system-level controller is sent into the engine model. The brake-specific fuel consumption (BSFC) map (adopted from [55]) has been used inside the model to obtain engine speed and torque needs in the form of a 2-D lookup table. It has also been used to determine the minimum engine idle speed, which is at 1000 rpm and engine maximum speed, and saturates the engine torque based on the maximum torque curve at each speed. In this model, we are neglecting the dynamics of the ICE and leaving out all the losses associated with it. Engine mechanical power P_{ICE} and fuel power $P_{ICE,fuel}$ can be written as

$$P_{ICE} = T_{ICE} \cdot \omega_{ICE} \tag{4.2.7}$$

$$P_{ICE,fuel} = \frac{P_{ICE}}{f_1\left(T_{ICE}, \omega_{ICE}\right)} \tag{4.2.8}$$

where T_{ICE}, ω_{ICE} and f_1 are engine torque, speed and the BSFC map (2-D lookup table), respectively.

4.2.3 Motor/Generator Model

Permanent magnet synchronous motor/generators rated at 650Vdc (MG1 and MG2) are built in this vehicle. The motor can operate in either propulsion or regenerative mode. The electric motor acts as a generator during regenerative braking, generating negative torque to assist vehicle deceleration while charging the battery pack. Combined motor-inverter efficiency maps (adopted from [53]) are used to simulate both motor/generators in the form of 2-D lookup tables. The motor torque is saturated by torque limits as a function of motor speed. Motor speed from the transmission model and motor torque from the system-level controller are used as inputs to the model. Electrical power for MG1 and MG2 can be stated as

$$P_{MG1,elec} = P_{MG1} \cdot f_2\left(T_{MG1}, \omega_{MG1}\right)^{-\operatorname{sgn}(P_{MG1})} \tag{4.2.9}$$

$$P_{MG1} = T_{MG1} \cdot \omega_{MG1} \tag{4.2.10}$$

$$P_{MG2,elec} = P_{MG2} \cdot f_3\left(T_{MG2}, \omega_{MG2}\right)^{-\operatorname{sgn}(P_{MG2})} \tag{4.2.11}$$

$$P_{MG2} = T_{MG2} \cdot \omega_{MG2} \tag{4.2.12}$$

where $f2$ and $f3$ are motor-inverter efficiency maps for MG1 and MG2, respectively.

4.2.4 Power-split Device Model

The power-split device or commonly known as the electronic continuously variable transmission (eCVT), consists of two planetary gearsets in this vehicle. The PSD is depicted schematically in figure 4.3. The sun gear is attached to each electric motor in both gear sets (MG1 and MG2). Two ring gears are linked to each other before being coupled with the final drive gearset (differential), which delivers power to the wheels. The first carrier link is connected to the engine, while the second carrier link is grounded.

It is worth noting that the powertrain components integrated by the planetary gearsets have two degrees of freedom. If two speeds or torques are determined, the other two torques and speeds can be determined by the equations.

$$\omega_{MG1} + n_1\omega_{out} = (1 + n_1)\,\omega_{ICE} \tag{4.2.13}$$

$$\omega_{MG2} = -n_2\omega_{out} \tag{4.2.14}$$

$$T_{out} = \frac{n_1}{1 + n_1}T_{ICE} + n_2 T_{MG2} \tag{4.2.15}$$

$$T_{MG1} = -\frac{1}{1 + n_1}T_{ICE} \tag{4.2.16}$$

where n_1 and n_2 as listed in table 4.1 are representing the gear ratio between Ring and Sun gears in gearset 1 and gearset 2, respectively.

4.2.5 Battery Model

The OCV-R equivalent circuit model is used to simulate the battery pack. The required current (I) from powertrain components, including the electric motor, generator, and electrical accessories, is the battery model input, as shown in the following equation. A constant electrical accessories power of $P_{acc} = 500W$ is considered.

$$I = I_{MG1} + I_{MG2} + \frac{P_{acc}}{V_{batt}} \tag{4.2.17}$$

The Coulomb Counting technique is used to estimate the battery state-of-charge (SOC) from input current and battery rated capacity C_{batt}. SOC_0 is the initial SOC value.

$$\dot{SOC} = -\frac{1}{C_{batt}} I(t) \tag{4.2.18}$$

$$SOC = SOC_0 - \frac{1}{C_{batt}} \int_0^t I(t)dt \tag{4.2.19}$$

By using test data in lookup tables, the estimated SOC is then utilized to obtain open-circuit voltage (OCV) and internal resistance (R) during charge and discharge.

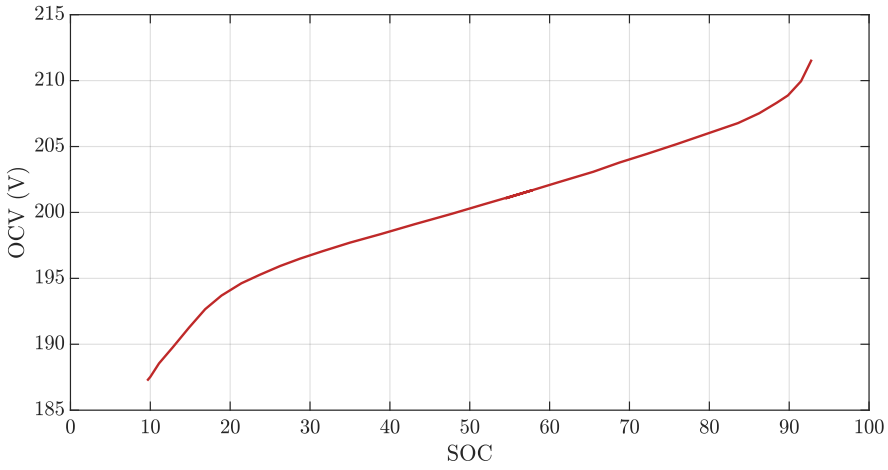


Figure 4.6: Open circuit voltage versus battery state of charge

Equation 4.2.20 is used to calculate the battery terminal voltage (V_T), which is then sent into electric motor models as an input.

$$V_T = OCV(SOC) - I \times R(SOC) \tag{4.2.20}$$

$$P_{batt} = V_T \times I \tag{4.2.21}$$

Note that $I \in \mathbb{R}$ and positive sign for I happens during battery discharge, and a negative sign is during the battery charging intervals.

4.2.6 Wheel and Final Drive Models

For the final drive model, as listed in table 4.1, the overall ratio in differential (counter-drive and final drive gears) is $n_{fd} = 0.3268$, and a 1-D look-up table is used to obtain the efficiency of the final drive based on torque input from the power-split model. Depending on the state of the vehicle (propulsion or braking), the input torque will

be either multiplied or divided by the efficiency.

For the wheel model, a constant efficiency of $\eta_{wh} = 0.98$ is used (listed in table 4.1) because of the slip. The output force of the wheel F_{wh} is calculated by output torque from final drive T_{fd}, braking torque command T_{brk}, and torque loss from rolling resistance (discussed in section 4.2.1).

$$F_{wh} = \frac{T_{fd} + T_{brk} - T_{loss}}{r_{wh}} \tag{4.2.22}$$

4.3 Driver Model

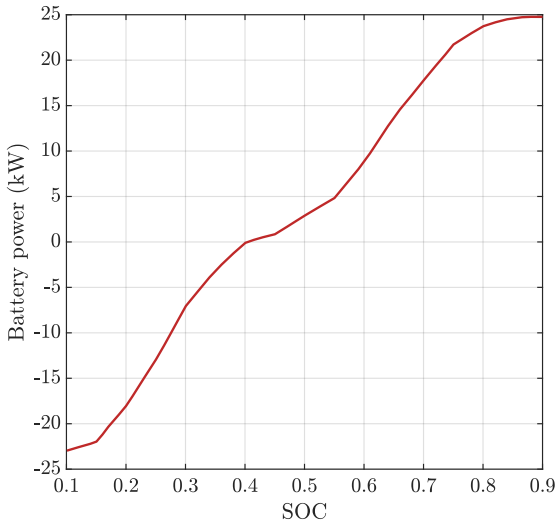
The driver model goal is to simulate the vehicle's required speed and torque based on the driving cycle chosen for the simulation. Vehicle dynamics calculations (4.2.1) define the required torque and power of the vehicle for a particular speed. Although, determining the torque and power requests from this open-loop approach is insufficient given the model uncertainty and powertrain limitations. Thus, to adjust the requests based on the driving cycle, a PI controller is used on the vehicle speed feedback and drive cycle speed difference. The PI controller parameters are obtained by trial and error to minimize the RMS error of driver requested speed and drive cycle speed.

4.4 Controller

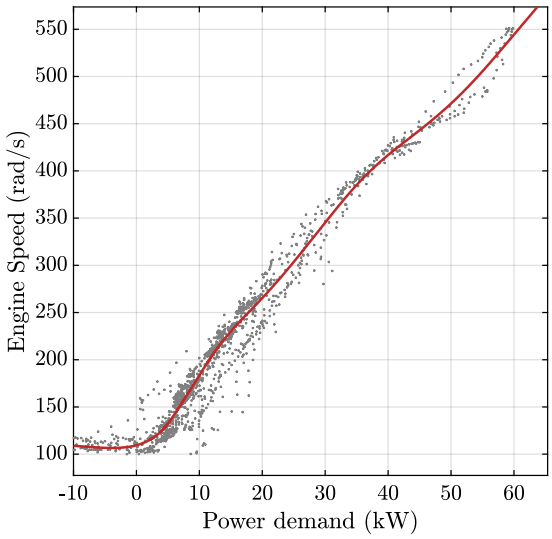
The system-level controller of the vehicle consists of high-level and rule-based algorithms, including engine on/off control, power-splitting strategy, energy management, and SOC balancing. The algorithm should also consider the vehicle components' limitations to guarantee that the final decision is feasible. The controller is also created

using MATLAB/Simulink blocks and Simulink Stateflow. The controller is responsible for generating the required control commands for the powertrain components from the driver's torque and power requests. In this work, the vehicle controller strategies have been developed based on the dynamometer test data and previous studies on Toyota Prius 2010 [51, 56, 57].

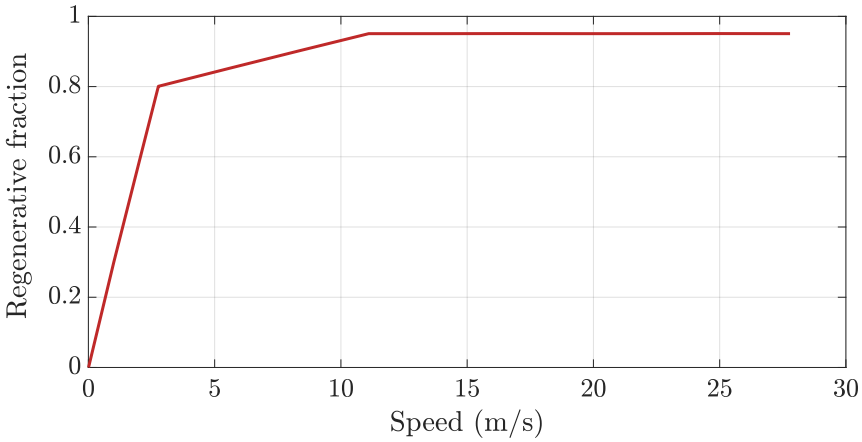
The first important thing to notice is that whether the requested power/torque is positive or negative. If it is positive, then the vehicle is in propulsion mode, and the powertrain needs to supply the power, and if it is negative vehicle is in the braking mode, and the powertrain needs to capture power. The next step is to define the engine-on/off control. When the requested power is lower than a threshold and the battery SOC is not considered low, the vehicle is controlled to run in a pure electric mode, and the engine is off. Due to having lower battery capacity in HEVs than EVs, the battery will be rapidly depleted in this mode, and the engine should be turned on at the right time. The threshold for the turning engine on/off is dependant on SOC (SOC_{th}), vehicle requested speed (v_{th}), and power (P_{th}) based on the actual data and model outputs. Table 4.3summarizes the statements of each control mode for engine-on/off strategy.



(a) Battery power vs. SOC



(b) Engine speed vs. Power demand



(c) Regenerative fraction vs. vehicle speed

Figure 4.7: Lookup tables used in control modes 1, 3, and 4.
Referred to as (a) $g_1(SOC)$, (b) $g_2(P_{dmd})$, and (c) $g_3(v)$

In mode 1, the engine speed is obtained based on the demanded power, as shown in figure 4.7b. This plot is fitted on vehicle power (product of vehicle speed and traction force) versus engine speed from dynamometer test data on the steady state

	Control Mode	Engine Mode	Condition
Propulsion	1	ON	$P_{dmd} \geq P_{th} \,\|\|\, v \geq v_{th} \,\|\|\, SOC \leq SOC_{th}$
	2	OFF	$P_{dmd} < P_{th}$ && $v < v_{th}$ && $SOC > SOC_{th}$
Braking	3	ON	$v \geq v_{th}$
	4	OFF	$v < v_{th}$

Table 4.3: Engine-on/off control modes and statements

drive cycle. The steady state drive cycle has been used mostly in developing the vehicle model and controller due to its characteristic of gradually accelerating and decelerating.

After determining the engine-on/off strategy, another important issue in HEV control is "SOC Balancing". Based on the actual test data of the vehicle, the battery SOC varies in a small range rather than utilizing the battery energy in a wide range of SOC. Of course, this approach is going to affect the energy management strategy by decreasing the system efficiency, but on the other hand, it is going to maintain the battery health and life by not using battery energy aggressively. In this regard, the desired battery power is related to the SOC and has been obtained from testing results [56]. Figure 4.7a demonstrates the relation between battery power and SOC used in the control strategy.

Based on each mode discussed above, a set of control inputs are defined for powertrain components as a power-splitting strategy. Control inputs for modes 1 and 2

in propulsion are listed in table 4.4.

Mode 1	Battery Power $P_{batt} = g_1(SOC)$
	Engine Speed $\omega_{ICE} = g_2(P_{dmd})$
	Engine Torque $T_{ICE} = (P_{dmd} - P_{batt})/\omega_{ICE}$
	MG1 Torque Obtained from 4.2.16
	MG1 Speed Obtained from 4.2.13
	MG2 Torque Obtained from 4.2.15
	MG2 Speed Obtained from 4.2.14
	Brake Power $P_{brk} = 0$
Mode 2	Engine Speed $\omega_{ICE} = 0$
	Engine Torque $T_{ICE} = 0$
	MG1 Torque $T_{MG1} = 0$
	MG1 Speed $\omega_{MG1} = 0$
	MG2 Torque $T_{MG2} = P_{dmd}/\omega_{MG2}$
	MG2 Speed Obtained from 4.2.14
	Brake Power $P_{brk} = 0$

Table 4.4: Control modes in propulsion

$g_1(SOC)$ and $g_2(P_{dmd})$ are the discussed lookup tables shown in figure 4.7.

As mentioned previously, MG2 is responsible for regenerative braking in assisting the vehicle's deceleration by producing negative torque during the braking. The amount of regenerative braking power is related to the vehicle speed, and it is calculated using a 1-D lookup table of regenerative fraction versus vehicle speed shown in figure 4.7c and referred to as $g_3(v)$. Control inputs for modes 3 and 4 during braking

are stated in table 4.5.

Mode 3	Engine Power $P_{ICE} = 1.17kW$, idle power
	Engine Speed $\omega_{ICE} = 1000rpm$ idle speed
	Engine Torque: $T_{ICE} = P_{ICE}/\omega_{ICE}$
	MG1 Torque Obtained from 4.2.16
	MG1 Speed Obtained from 4.2.13
	MG2 Torque $T_{MG2} = (g_3(v).P_{dmd})/\omega_{MG2}$
	MG2 Speed Obtained from 4.2.14
	Brake Power $P_{brk} = P_{dmd} - P_{MG2}$
Mode 4	Engine Speed $\omega_{ICE} = 0$
	Engine Torque $T_{ICE} = 0$
	MG1 Torque $T_{MG1} = 0$
	MG1 Speed Obtained from 4.2.13
	MG2 Torque $T_{MG2} = (g_3(v).P_{dmd})/\omega_{MG2}$
	MG2 Speed Obtained from 4.2.14
	Brake Power $P_{brk} = P_{dmd} - P_{MG2}$

Table 4.5: Control modes in braking

4.5 Evaluations

The final step in developing the vehicle model is to evaluate its performance under various driving conditions and compare it with available test data to ensure that model has sufficient accuracy for further developing the controller and embedded EMS. The available dynamometer test data available [51] is limited to vehicle outputs (such

as SOC, ICE speed, fuel consumption, etc.) under four driving cycles of UDDS, Highway, US06 (aggressive highway driving), and steady state.

The main driving cycle used to develop the model parameters and improve accuracy is the steady state drive cycle. It covers most of the speed ranges and constant acceleration/deceleration rates. The results for vehicle linear speed, battery SOC, ICE speed, and fuel flow rate has shown in the four following figures 4.8, 4.9, 4.10, 4.11.

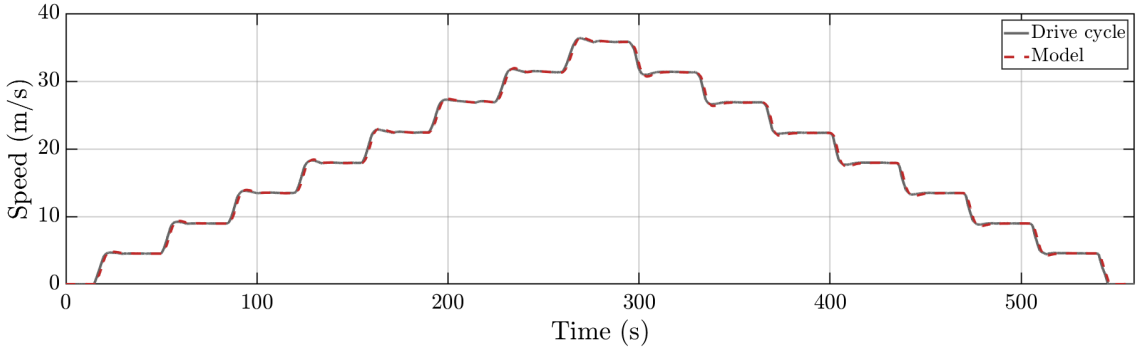


Figure 4.8: Model linear speed in comparison with reference driving cycle speed. Overall RMS error is 0.243 m/s.

It can be seen that the model outputs are following the reference data with acceptable root mean square (RMS) errors. Most of the deviations are due to simplified assumptions in components dynamics modeling (ICE, battery, transmission, etc.). In particular, the model engine speed inconsistency in following test data is due to the lack of engine thermal modeling, which affects the idle speed at braking periods with high speeds that the engine is working at idling conditions [58]. Also, the difference in model and test SOC profiles is due to the impact of error accumulation over time in the SOC estimation method (Coulomb-counting method). In general, model

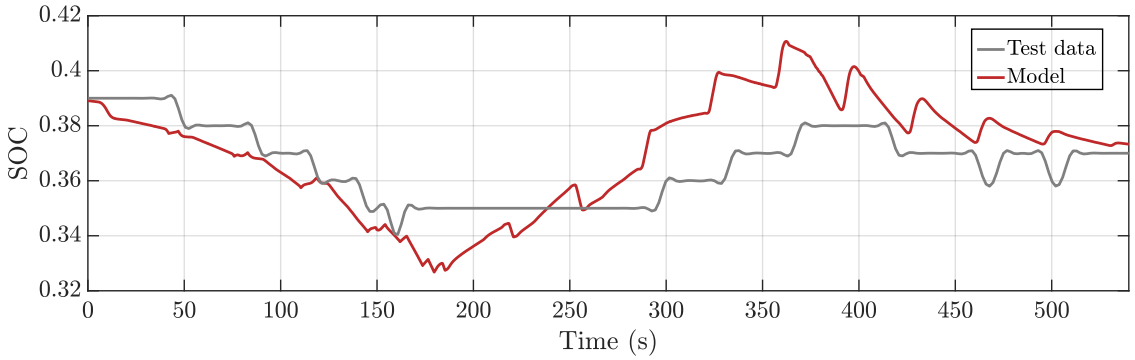


Figure 4.9: Model SOC profile in comparison with SOC profile in test data. Overall RMS error is 0.0136.

output signals are following the trend of test data signals which shows promising correspondence with actual vehicle experiments.

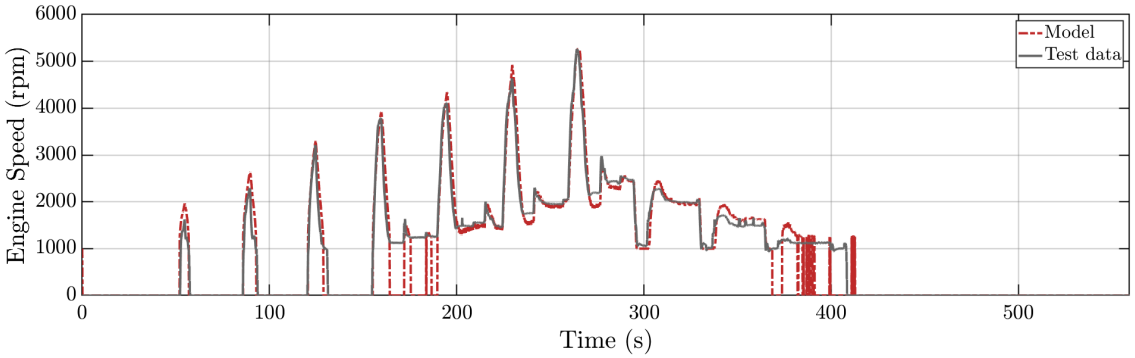


Figure 4.10: Model engine speed profile in comparison with test data engine speed. Overall RMS error is 375.18 rpm.

Furthermore, the model has been evaluated with UDDS driving cycles for the same output parameters to capture the effect of different driving conditions. The results are shown in the following figures 4.12, 4.13, 4.14, 4.15. Same observations as steady state results can be made about UDDS model outputs; however, the RMS

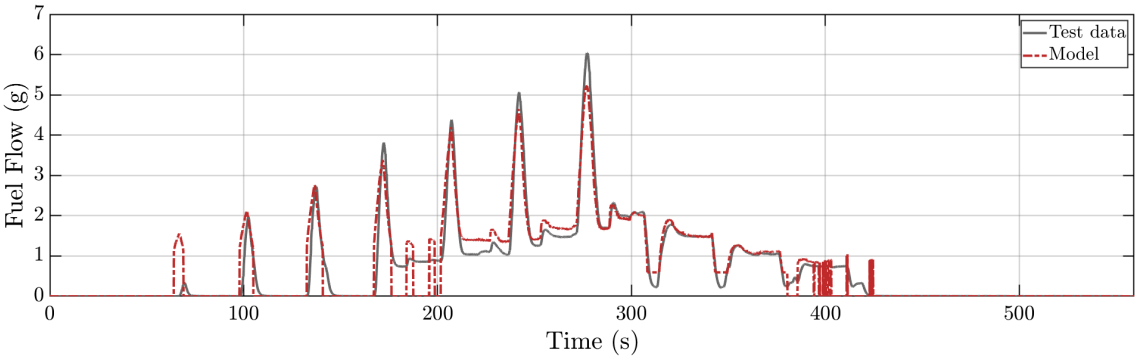


Figure 4.11: Model fuel consumption rate profile in comparison with test data fuel consumption rate. Overall RMS error is 0.622 g/s.

value for each case is different.

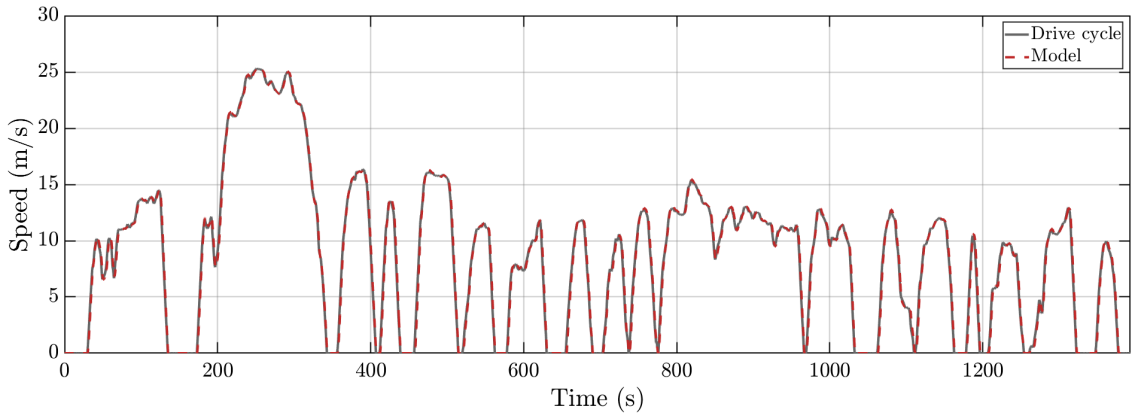


Figure 4.12: Model linear speed in comparison with reference driving cycle speed. Overall RMS error is 0.428 m/s.

Table 4.6 summarizes the fuel consumption and final SOC value for all four drive cycles in both model and test data. It is essential to evaluate the overall fuel consumption output of the model because, ultimately, the model will be used to develop EMS and compare the fuel economy. Also, the fuel consumption is highly dependant on the

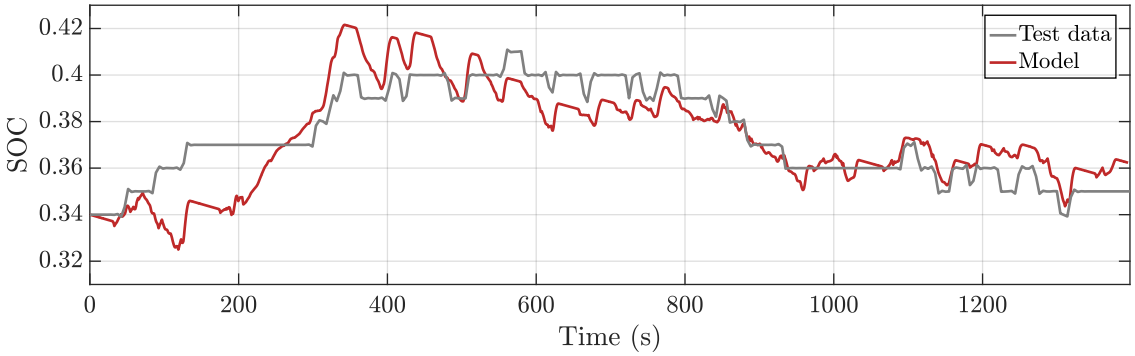


Figure 4.13: Model SOC profile in comparison with SOC profile in test data. Overall RMS error is 0.0123.

SOC, and for having a fair comparison between fuel consumption, it is important to capture approximately the same SOC at the end of each driving cycle. Results show that both fuel consumption and final SOC are very close to actual vehicle outputs. As a result, the accuracy of the developed HEV model is acceptable for developing the proposed energy management strategy.

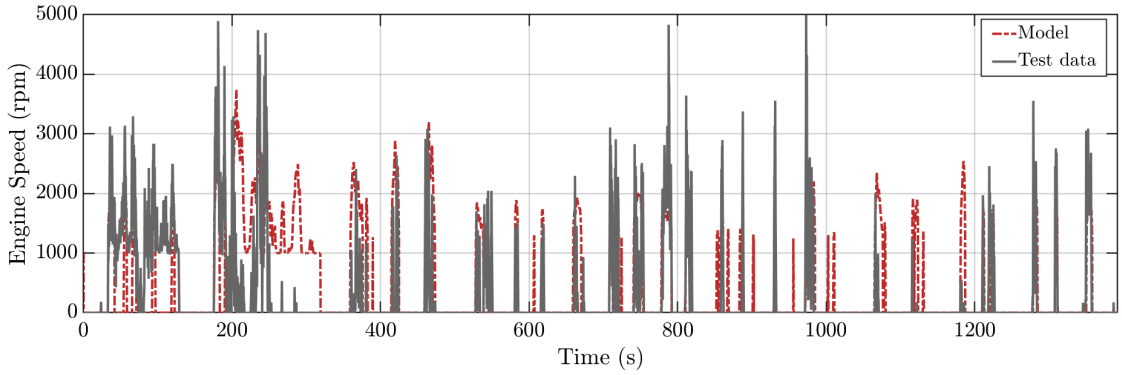


Figure 4.14: Model engine speed profile in comparison with test data engine speed. Overall RMS error is 405.99 rpm.

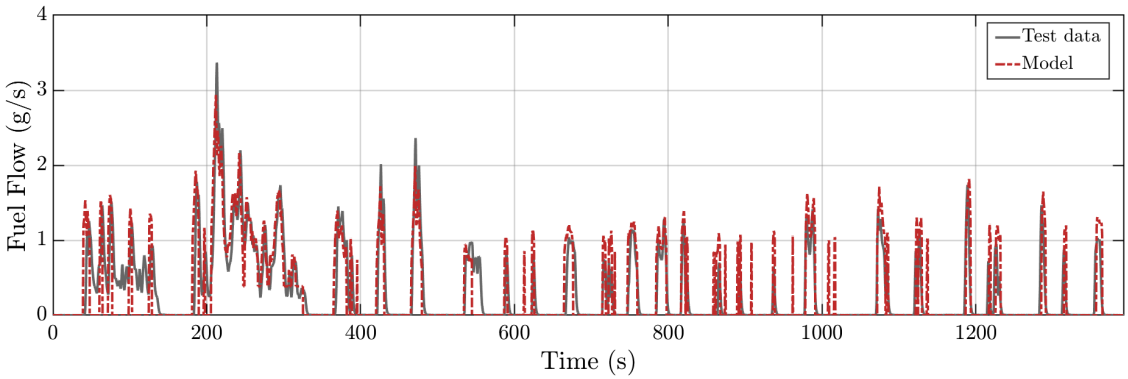


Figure 4.15: Model fuel consumption rate profile in comparison with test data fuel consumption rate. Overall RMS error is 0.423 g/s.

Drive cycle	Fuel consumption (g)			End SOC (%)		
	Test	Model	Error	Test	Model	Diff.
UDDS	403.5	402.3	-0.29%	35	36.23	1.23
Highway	1227	1183	-3.58%	40	43.69	3.69
US06	1459	1399	-4.11%	43	44.66	1.66
Steady State	388.9	388.4	-0.12%	37	37.24	0.24

Table 4.6: Fuel consumption and final SOC of model and test data comparison in different drive cycles
used

Chapter 5

Offline LSTM Multi-Parameter Prediction Model

5.1 Introduction

In section 3.6, a brief overview of intelligent energy management strategies using Neural Networks and Learning-based algorithms has been presented. In this work, a deep recurrent neural network (RNN) has chosen to estimate or predict the power distribution based on a few driving features such as vehicle speed and battery SOC.

For historical time series problems such as SOC estimation [59, 60], speech recognition [61, 62], and time-series predictions [63, 64], Recurrent Neural networks have demonstrated significant feature extraction and prediction capabilities. The EMS problem itself has consisted of time series data which can be formulated as time series prediction problem.

The recurrent neural network is a closed-loop neural network that leverages past information. Sending the network output, or an intermediate state, as an input

to a neural network can make it recurrent. Figure 5.1shows an RNN architecture being unfolded in time. Whenever short-term dependencies are required in output data, this conventional form of RNN is suitable; however, it may not function as effectively for long-term dependencies due to issues during training [65]. The LSTM, the bidirectional LSTM (BiLSTM), and the gated recurrent unit (GRU) are all RNN variants that were designed to address the aforementioned issues [66].

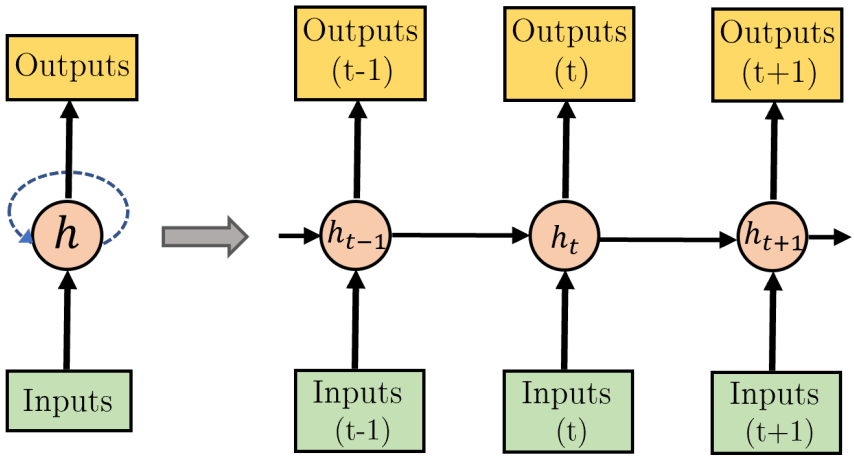


Figure 5.1: Architecture of RNN (left side) and RNN unfolded in time (right side), where t is the current time, and h_t is the hidden layer of LSTM RNN at the current time.

In this chapter, the chosen RNN network, which is the Long Short-Term Memory (LSTM) network, data processing, and methodology, will be discussed. A sensitivity analysis is done to improve the performance of the offline LSTM multi-parameter prediction model for further utilization of the model in an online EMS platform. This model can be used for both prediction and estimation. In this work, only the estimation part was explored where the controlled variables are being estimated for the next time step. The model also has the capability to predict the variables for longer time steps in the future which can be explored in future work. Due to this

extended capability, the model will be referred to as prediction model rather than estimation model for generalization purposes.

5.2 Data Description and Pre-Processing

5.2.1 Data Description

As discussed in the section 4.4, some multiple signals and data impact the supervisory control algorithm of the vehicle. Input information for the rule-based EMS implemented previously was based on battery SOC, chassis, or vehicle speed, power, or torque demand from the driver. The outputs of the controller, as shown in tables 4.4 and 4.5, are powertrain components' input signals. Based on power-split equations (4.2.13 - 4.2.16) and the control algorithm itself, it can be concluded that there are only two degrees of freedom, meaning if two of the independent powertrain components inputs (T_{ICE}, ω_{MG1}, T_{MG1}, ω_{ICE}, T_{MG2}, ω_{MG2}, P_{batt}) are determined, the rest can be calculated through equations of power-split or energy balance equation.

Inputs and outputs of the proposed model are chosen to replace the original supervisory control algorithm with minor changes instead of extracting too many features (such as max and min speed and acceleration, trip length, etc.) from drive cycles in comparison to previous works [67, 68], the LSTM RNN will extract relevant information and store them in cell state, which decreases the artificial interference while extracting driving cycle data [69].

In conclusion, similar to the rule-based supervisory control, the inputs to the model are vehicle speed (v), acceleration (a), battery SOC (SOC), and demanded torque (T_{dmd}). Two independent parameters of engine speed (ω_{ICE}) and engine torque

Drive Cycle	Distance (km)	Time (s)	Maximum Speed (m/s)	Average Speed (m/s)
HWFET	33	1565	26.78	21.09
UDDS	12	1390	25.32	8.61
WLTC class 3b	23.26	1800	36.47	12.92

Table 5.1: Training Drive Cycles Characteristics

(T_{ICE}) have been chosen as the network output.

5.2.2 Data Pre-Processing

Data for training, testing, and validation of the model are derived from the HEV vehicle model (Chapter 4) because the amount of data from dynamometer testing [51] are limited to only a set of data for four drive cycles. However, the model fidelity is already evaluated with the vehicle testing data to ensure that information given to the network has an adequate quality for training.

Choosing a training/testing dataset that involves different road types, driving styles, and control modes has significant importance. Thus, three drive cycles of UDDS, Highway Fuel Economy driving schedule (HWFET), and Worldwide Harmonized Light-duty vehicles Test Cycle (WLTC) class 3b, has been chosen for this matter. These drive cycles are able to capture urban, extra-urban, and highway road types. Furthermore, to learn different modes of rule-based control algorithms, all four control modes are being captured fairly. Table 5.1 shows the detailed information of each drive cycle.

It was important to capture an urban schedule with low average speed and frequent

start and stops (UDDS), along with a highway cycle with less than 17% brakes along the way (HWFET). The selection of the third drive cycle (WLTC) was based on a lack of very high speeds in the other two drive cycles (directly affecting fuel consumption). In this regard, the WLTC class 3b has been chosen, which is a recent replacement of the New European Drive Cycle (NEDC), and it is representative of the real and modern driving conditions [70]. Figure 5.2 shows the vehicle speed for this drive cycle.

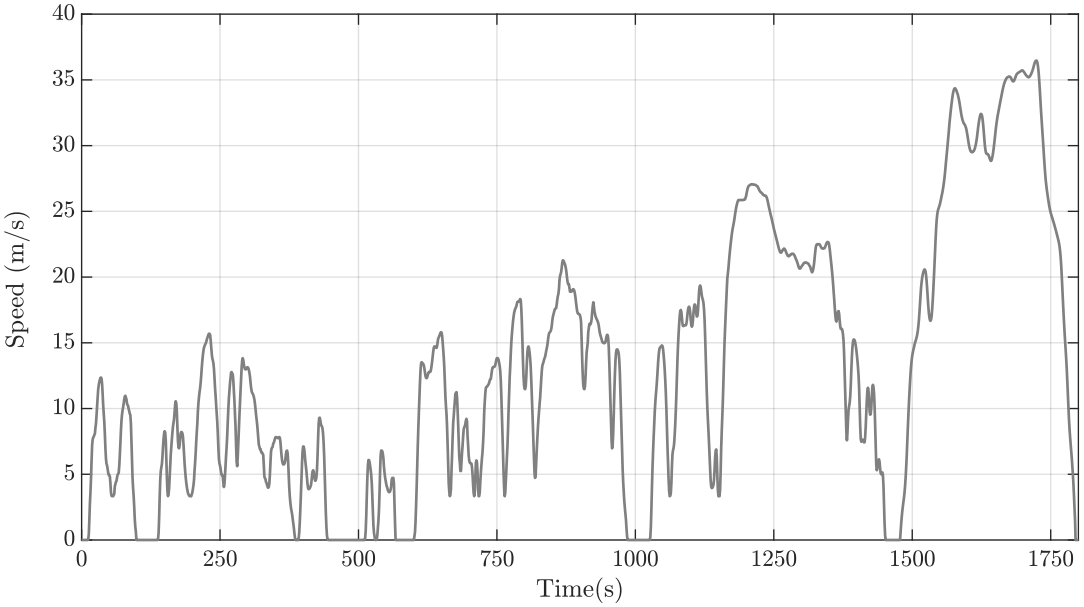


Figure 5.2: WLTC class 3b driving schedule

Battery SOC affects the engine power production (which impacts T_{ICE}) and ultimately on fuel consumption. Moreover, it is one of the inputs that can be changed and played with compared to the other three inputs. Thus, in the dataset, for each drive cycle, we changed the battery SOC from 0.30 to 0.70. Because the sensitivity of the battery SOC is 1% in dynamometer testing data, we chose to change the battery SOC in steps of 1% and randomly sample data from total available data as testing and validation datasets.

The input data's sampling time (time step) has been set to $\Delta t = 1s$ because the HEV's system reaction is not too quick, and the model complexity and adaptability must be balanced. However, the impact of changing time window size in training and prediction will be assessed thoroughly in the following sections. Hence, in total, 194,955 seconds of training/testing/validation data are available for the network.

The final step in data pre-processing is the normalization of data. The normalization is important for training and testing the network to eliminate the effect of data dimensions. There are a few common normalization methods used in previous works, which in this work, two popular methods had been tested. The first method scales the input and output data to 0-1, called maximum and minimum normalization, [71] shown in the following equation.

$$\widehat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{5.2.1}$$

where x_{min} and x_{max} for each data type determined based on the vehicle and powertrain components specifications.

The second method uses the average and standard deviation of data to create a standard score normalization with zero mean and unit variance [72, 73].

$$\widehat{x} = \frac{x - \mu}{\sigma} \tag{5.2.2}$$

where μ is the mean of the data and σ is the standard deviation of the data.

The second method with standard score normalization had been chosen due to lower errors with the same network parameters. The problem with the max and min normalization was that the engine speed and torque values in zero or near-zero values

were not being predicted accurately.

Finally, the procedure of obtaining data from the HEV model and implemented a rule-based controller, storing it into a database, normalization, and passing it on to the network training module has been illustrated in figure 5.3.

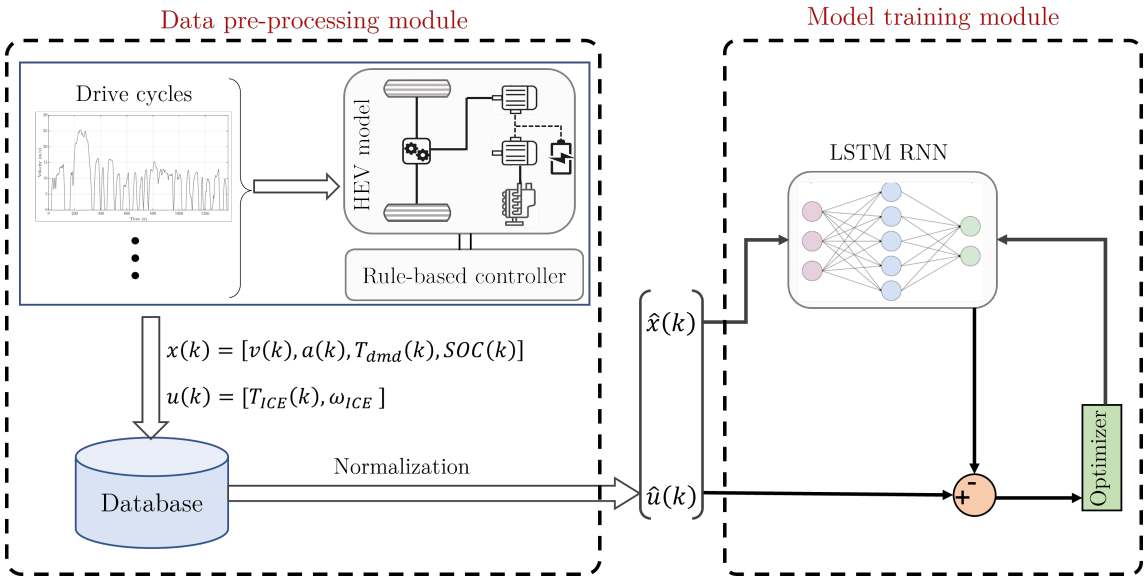


Figure 5.3: Data pre-processing and model training modules and their connections

For more clarification on the distribution of data in each training, testing and validation dataset, the break down of each set with number of initial SOCs used (either generated randomly or picked in a range) is shown in Table 5.2.

Dataset	Drive Cycles	No. of initial SOCs used	Time length (s)	Percentage (%)
Total	UDDS HWFET WLTP	41	194955	100
Training		23	166425	85.4
Testing		6	9510	4.9
Validation		12	19020	9.7

Table 5.2: Training, testing, and validation dataset breakdown based on size and initial SOCs

5.3 Methodology

To apply the LSTM network model on HEV to predict engine parameters during the vehicle operation, a few issues will be dealt with. The first issue is to ensure that the network converges to the right network weights. Another issue is that in most of the literature using LSTM network for battery or powertrain parameters prediction, the output of the regression model is a single output (e.g., battery SOC, engine torque, velocity, etc.) [59,60,69,74]. So the challenge in this model is to train and optimize the model on two independent and different in dimension parameters. Another concern for a real-time implementation is the size of the trained network, which should not be too large in order to avoid time delays in online prediction and control of the powertrain components.

Numerous network structures and training parameters have been explored to address the above-mentioned challenges, which will be discussed in the following sections. The developed model used a many-to-many (m-n) LSTM architecture, as can

be seen in figure 5.4 for the multi-parameter prediction issue mentioned earlier.

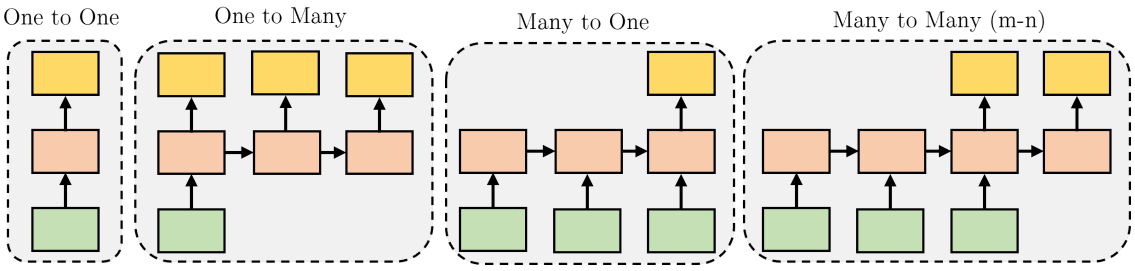


Figure 5.4: Different mapping types of LSTM [75]. Input vectors are shown in green, output vectors are in yellow, and RNN's states are in red rectangles. Arrows represents mathematical functions.

5.3.1 LSTM Networks

LSTM Networks, as mentioned, are a special form of RNN with the capability of learning long-term dependencies. Hochreiter & Schmidhuber [76] introduced them in 1997 and many researchers improved and popularized them in future works. All RNNs are made up of series of repeated NN modules. This repeating module can have a relatively basic structure, such as a single *tanh* layer in conventional RNNs. LSTMs have a chain-like structure as well, but the repeating module is more complicated. As it can be seen in figure 5.5 each module is consisted of four NN layers, each of which interacts in a unique way [77].

Using a memory cell c_k, the LSTM can store and transmit information from previous and present states for future use. The data flow through the cell state (the upper horizontal stream in figure 5.5) is regulated by gates. LSTM has three gates of input, output, and forget, which can selectively allow information to pass through each cell. A sigmoid neural net layer plus a point-wise multiplication operation make

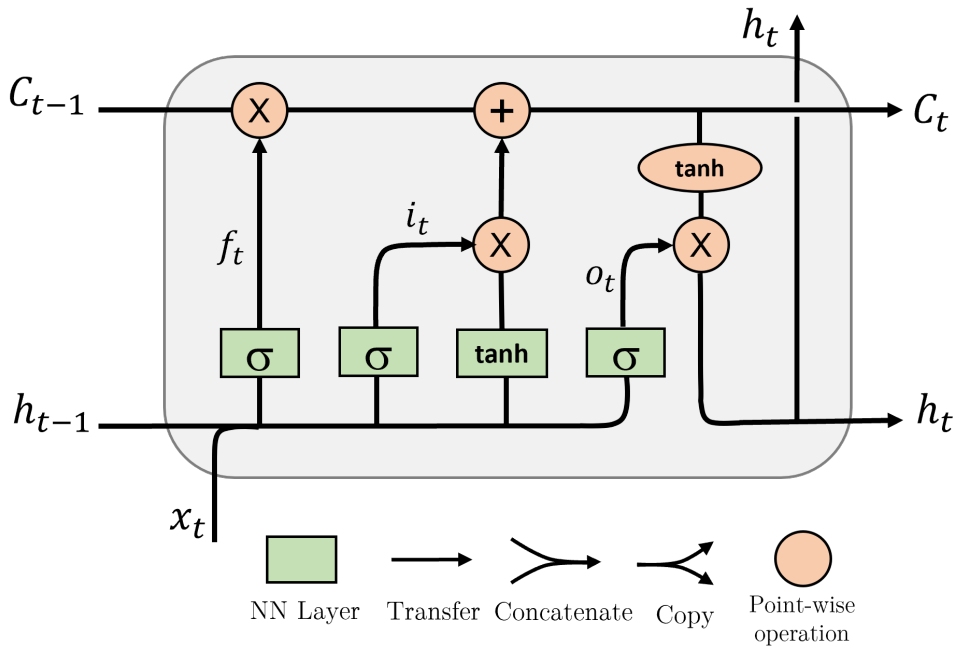


Figure 5.5: Schematic of LSTM layer module [77]

up each gate. A step-by-step explanation for each gate, starting from the left side of the LSTM layer figure is given below.

***Forget gate*:** The decision to save or discard information is made by this layer which is the first step of LSTM. Inputs to the gate are output from previous time-step (h_{t-1}), and input at current time-step (x_t). The output of the sigmoid function will be a value $\in$ [0 1], representing fully discard or fully save the data, respectively. In the case of this work problem, the discarded information can be noises and irrelevant parameters.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5.3.1}$$

where W_f and b_f are the weight and bias for the forget gate, respectively.

***Input gate*:** This layer, in the next step, decide the values to be updated, and then a tanh activation function squeezes the input between [-1 1]. Input vectors to

this gate are the same as to forget gate. The input gate i_t and tanh layer g_t can be expressed as

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{5.3.2}$$

$$g_t = \tanh\left(W_g \cdot [h_{t-1}, x_t] + b_g\right) \tag{5.3.3}$$

where W_i and W_g represent the weights for input and previous output, respectively, and b_f and b_g are the corresponding biases.

***Output gate*:** This layer which is expressed as follows, determines what information is eventually outputted. The output is based on the filtered version of the cell state. Similar to the input gate, the sigmoid layer decides for output values, and then it is multiplied to the cell state, which went through tanh. Inputs of this gate are equal to the previous gates.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{5.3.4}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{5.3.5}$$

where h_t is the output of the LSTM layer at the current time-step, W_o and b_o are the weight and bias for the output gate, respectively.

The final step is to update the previous cell state (c_{t-1}), which, as it can be seen from the figure is calculated through forget and input gates.

$$C_t = f_t * C_{t-1} + i_t * g_t \tag{5.3.6}$$

5.3.2 Architecture of LSTM multi-parameter prediction

The LSTM multi-parameter prediction network has been trained using MATLAB Deep Learning Toolbox, which is easy to use and implement for this case as both the HEV model and data preparation have been done in the same software platform.

There are four major steps for constructing the model: Define, compile & fit, verify, and predict.

1. Define: In the first step, we need to define both the network and the portion of the training to testing data. The appropriate size of the testing dataset usually varies from approximately 10% to 30% of the training dataset. Defining the network structure, meaning determining the number of layers and number of hidden units or neurons in each layer, is of high importance and also complicated to achieve the optimal network structure. the simulation results of the model with different network architectures and size of the testing dataset are shown in the next section.

2. Compile & fit: After defining the network structure, we should specify various parameters for training to be compiled, such as optimizer, initial learn rate, time window or sequence length, weight parameters initialization, mini-batch size, etc. The NN training uses the backpropagation method, so an optimal number of epochs need to be determined to achieve lower training losses without facing overfitting or underfitting problem.

3. Verify: After completion of training, the network performance needs to be evaluated and analyzed against the effects of changing parameters mentioned in the second step on validation datasets. This will ensure the model's generalization

capability and leads to finding the model that best fits our requirements.

4. Predict: Finally, after choosing the well-fitted model, the model can be used to perform predictions on new sets of data.

Based on the above procedure, the LSTM multi-parameter prediction model for powertrain EMS can be designed as demonstrated in figure 5.6.

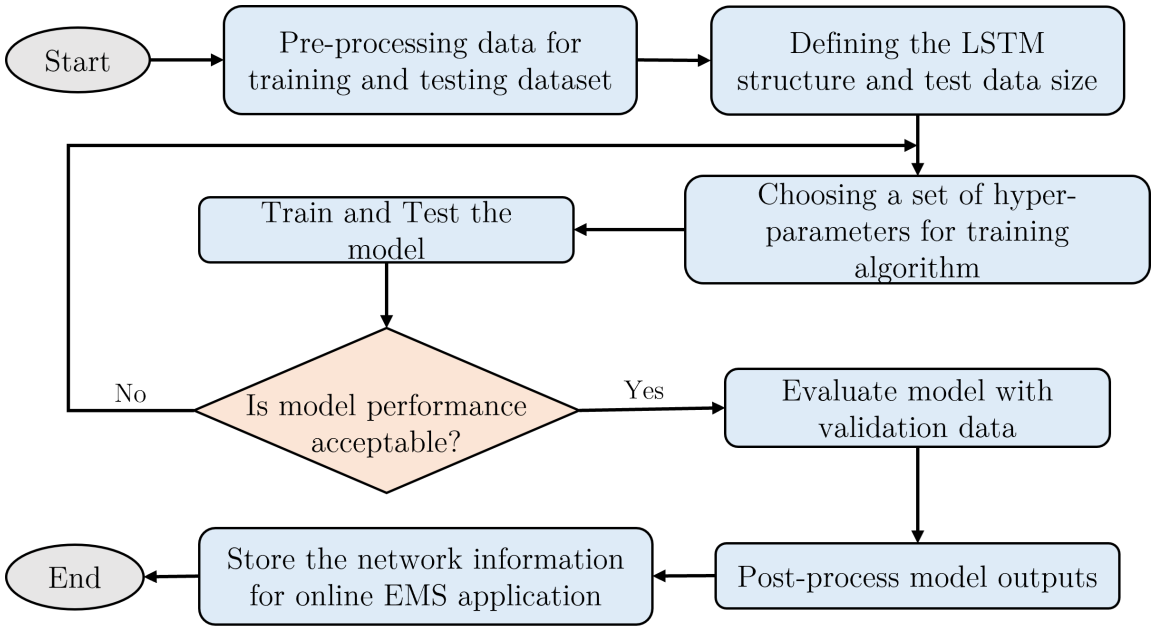


Figure 5.6: LSTM multi-parameter prediction model design flowchart

5.3.3 Evaluation Metrics

The goal in the training of the network is to minimize the error between predicted outputs ($T_{ICE}(k \mid \theta)$ and $\omega_{ICE}(k \mid \theta)$) and the reference controller data from HEV model (T_{ICE} and ω_{ICE}). Various performance metrics have been used in previous works, such as mean square error (MSE) [67, 78], mean absolute error (MAE), and

root mean square error (RMSE) [59, 64, 71], and mean relative error (MRE) [60]. In this work MAE of normalized data has been chosen as the main performance index; however, the RMSE of de-normalized data has also been calculated to give a better sense of error values in their own units and dimensions. MRE is not a suitable performance index in this work due to calculating large errors for data values near zero, which often happens in engine speed and torque outputs.

$$\begin{aligned} RMSE_{trq} &= \sqrt{\frac{\sum_{i=1}^{N} \left(T_{ICE}^{i}(i\Delta t \mid \theta) - T_{ICE}^{i}(i\Delta t)\right)^2}{N}} \\ RMSE_{spd} &= \sqrt{\frac{\sum_{i=1}^{N} \left(\omega_{ICE}^{i}(i\Delta t \mid \theta) - \omega_{ICE}^{i}(i\Delta t)\right)^2}{N}} \end{aligned} \tag{5.3.7}$$

Besides, because the network has two outputs of different units, it is important to evaluate the whole output dataset; thus, the MAE and RMSE of concatenated normalized outputs are also calculated and compared for each case.

$$\min_{\theta \in \Theta} MAE = \frac{1}{M} \sum_{i=1}^{M} |Y_i(i\Delta t \mid \theta) - Y_i(i\Delta t)| \tag{5.3.8}$$

where $M = 2N$ and Y_i are concatenated vectors of engine torque and speed vectors.

5.4 Simulation Analysis and Discussion

5.4.1 Comparison of different Network Structures and training dataset

s

The network structure consists of a sequence input layer, one or multiple hidden

sequence-to-sequence LSTM layers, activation layers, a fully connected layer, and an output regression layer. In general, hidden layers represent the relationship between past and future time series; they can also help the LSTM network create more complicated models and increase generalization ability. In this work, different structures have been chosen and simulated to find the optimal network structure. Table 5.3 shows different network structures used in this work to find the optimal number of hidden lstm layers and the number of hidden units or neurons in each layer. Also, the 'tanh' function had been chosen as the activation function for the hidden layers in this work to reduce the transfer error between layers. The network definition in MATLAB code can be found in Appendix A.

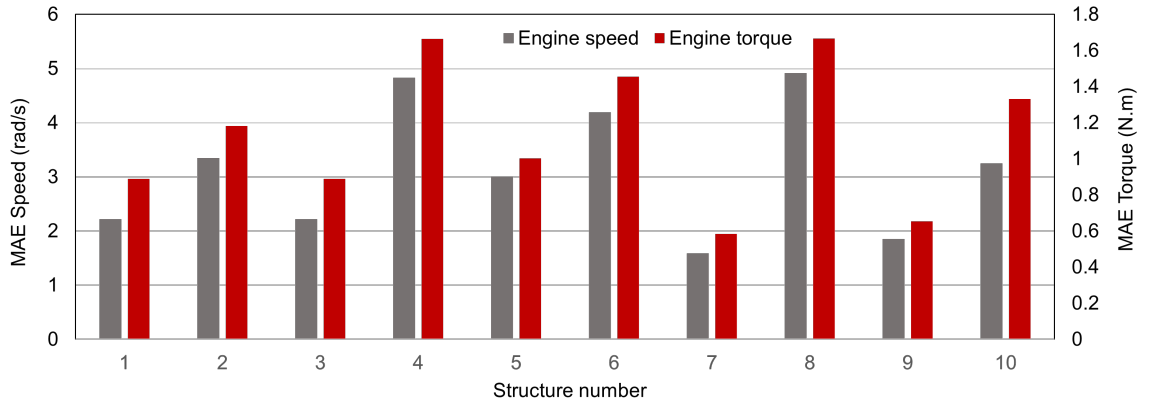
The funnel-shaped structures have been chosen to cover up to 5 layers and up to 120 hidden units in total. Also, various combinations of training and testing dataset proportions have been explored to discover the best-performed network since the link between the size of the testing dataset and network structure is difficult to detect. From the available data described in the section 5.2.2, 16 randomly selected datasets were set aside for validation purposes, meaning those datasets have never been used to train and test the network. From the remaining data for training, 10%, 20%, and 30% were randomly selected as testing datasets.

Structure number	Hidden Layer				
	1	2	3	4	5
1	30	0	0	0	0
2	50	0	0	0	0
3	100	0	0	0	0
4	10	10	0	0	0
5	30	20	0	0	0
6	30	10	10	0	0
7	80	30	10	0	0
8	10	10	10	10	0
9	50	30	20	10	0
10	50	30	20	10	10

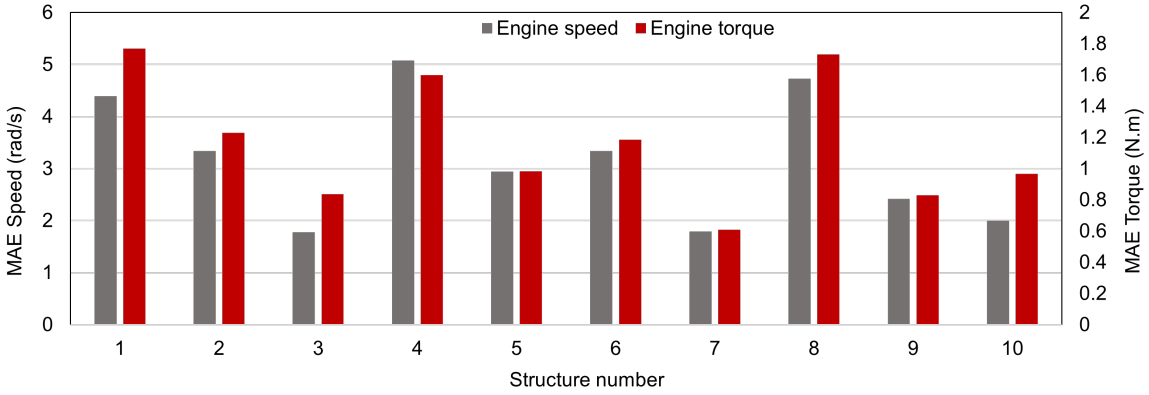
Table 5.3: Different network structures of hidden layers and hidden units

Simulation results for each network structure under three different sizes of the testing dataset are shown in figure 5.7. The performance of the LSTM multi-parameter prediction model has been evaluated based on MAE of model outputs T_{ICE} and ω_{ICE}. It can be seen that there is not a unique trend between performance with different structures and testing dataset sizes. However, it is obvious that in all three test sizes, structure number 7 has shown the best performance in both T_{ICE} and ω_{ICE} and increasing the number of hidden layers to more than three layers does not necessarily improve the model performance. Also, it can be concluded that in most of the structures, the model performance with a 10% testing dataset is better than the other two.

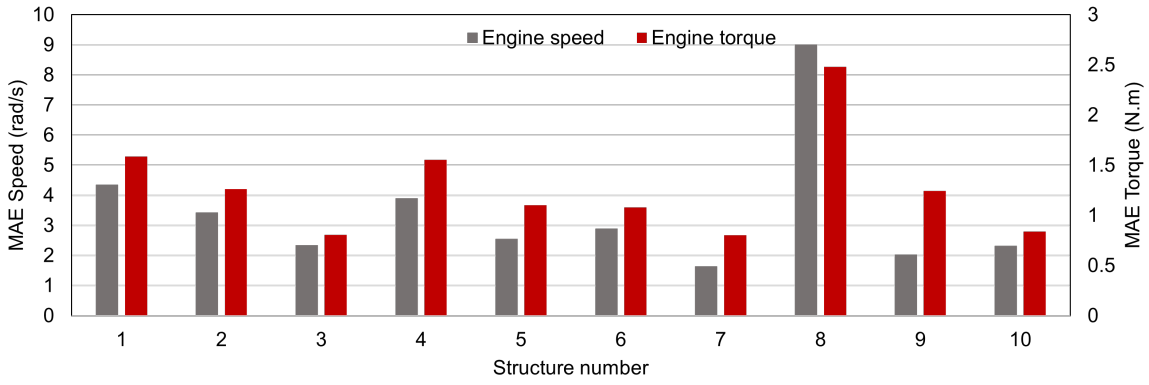
Thus, structure number 7, with three hidden layers of 80, 30, and 10 hidden units and 10% testing dataset size, has been selected for the rest of the work. The MAE of T_{ICE} and ω_{ICE} for that specific model in validation dataset is 0.58 N.m and 1.59 rad/s, respectively.



(a) with 10% of the testing dataset



(b) with 20% of the testing dataset



(c) with 30% of the testing dataset

Figure 5.7: Performance of LSTM multi-parameter model under various network structures and testing dataset sizes on the validation dataset.

5.4.2 Selection and Debugging of Hyper-parameters

There are multiple training-related and model-related parameters that can be optimized towards decreasing the training and validation losses. In the following sections, the selection procedure of a few important hyper-parameters will be discussed. To keep consistency between results for comparison, we only sample the testing dataset 'once' randomly from 10% of the whole training dataset. As discussed in the previous subsection, the chosen network with structure number 7 has been used to determine the hyper-parameters in the following sections.

5.4.3 Selection of training options

One of the important model-related hyper-parameters to determine is the optimizer or solver for the training network. Optimizers that can be used in MATLAB Deep Learning Toolbox are SGDM (stochastic gradient descent with momentum) , RMSProp (root mean square propagation), and Adam (adaptive moment estimation) [79]. Adam, a method for efficient stochastic optimization, is the most used optimizer in deep learning and machine learning applications [80]. It combines the benefits of two optimization approaches, AdaCrad and RMSProp, to improve the performance of the LSTM model.

According to the analysis done in [60], Adam optimizer offers outstanding features of having a lower need for computer memory processing, having the ability to compute multiple adaptive learning rates for various factors, and more significantly being more suitable to a huge dataset and a high dimensional space, in comparison of mentioned optimizers. So, the Adam optimizer has been chosen in this work due to numerous parameters that will be obtained and used throughout the LSTM-oriented modeling

process.

Learning rate is another training-related parameter that needs to be determined. It has a significant influence on the consistency and efficiency of training times. It is common to have a high learning rate at the start of the training process and then see it drops over time as it approaches convergence.

In this regard, a step decay method has been chosen for setting up the learning rate, in which the learning rate decays every n number of epochs. This method can be set in MATLAB training options as piece-wise learn rate schedule'. Following this method, two other factors of learn rate drop period (n) and drop rate (η) should be determined as well. As a common practice, the initial learn rate has been set to 0.01, and various drop rates simulated as shown in figure 5.8. It can be seen that generally, by increasing the η, the performance of the network increases as well. So, $\eta = 0.9$ with $n = 100$ epochs has been chosen.

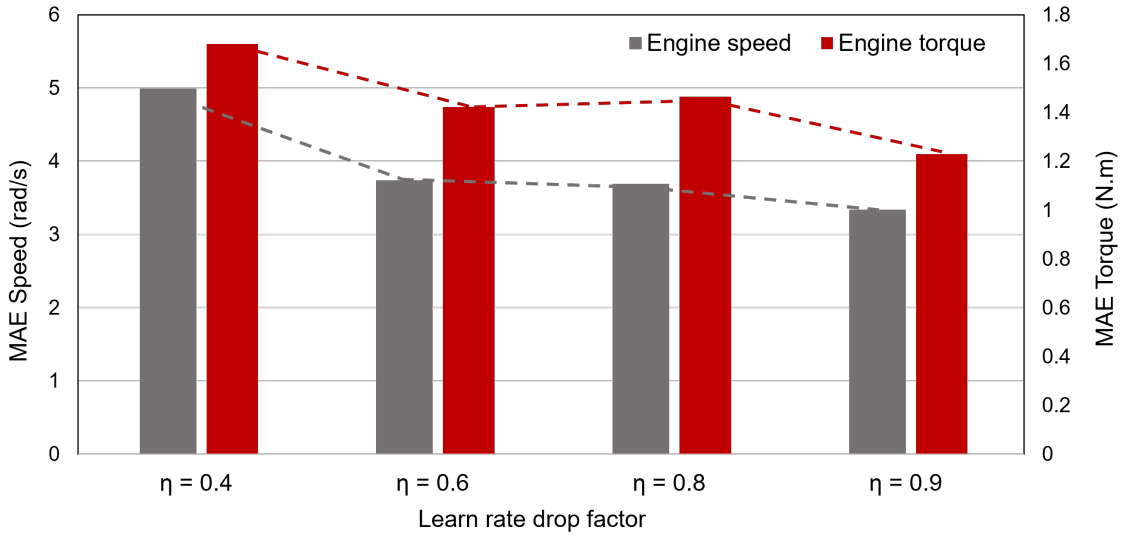


Figure 5.8: MAEs of predicted data based on different learn rate drop factors.

Mini-batch size is a critical model parameter that can be different for various

datasets. The batch size (BS) initially determines the gradient's direction of descent and updates the weights. Larger mini-batch sizes will decrease the number of iterations required to execute an epoch (full dataset) and enhance the same data processing performance. On the other hand, it can increase the operating time and memory usage. As a result, the mini-batch size should be adjusted to meet the prediction accuracy within a reasonable run time [60].

In this work, five mini-batch sizes of 4, 8, 16, 32, and 64 have been investigated, and the MAEs of predicted data with each BS are indicated in figure 5.9. It can be seen that the lowest validation errors for both T_{ICE} and ω_{ICE} happened in BS = 4 and then in BS = 16, and then the validation MAE increases as mini-batch size increases. Also, the convergence rate of validation loss was the fastest in BS = 4, as shown in figure 5.10. Hence, the mini-batch size of 4 has been selected for the model.

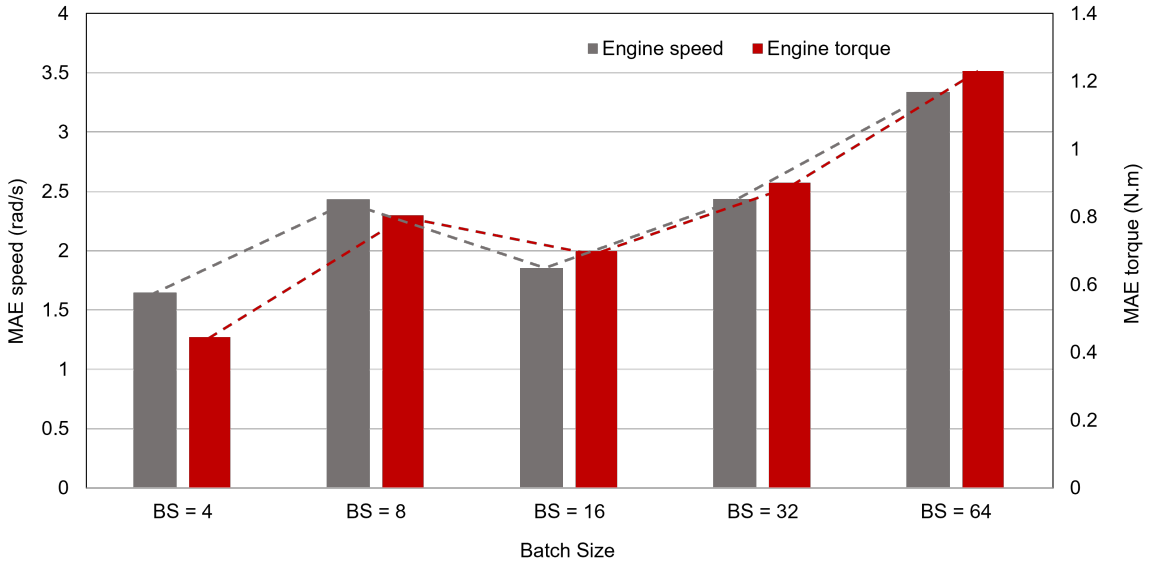


Figure 5.9: MAEs of predicted data based on different mini-batch sizes.

A maximum number of epochs for training was set to 500, as the training and validation loss converged well, and running under a higher number of epochs could

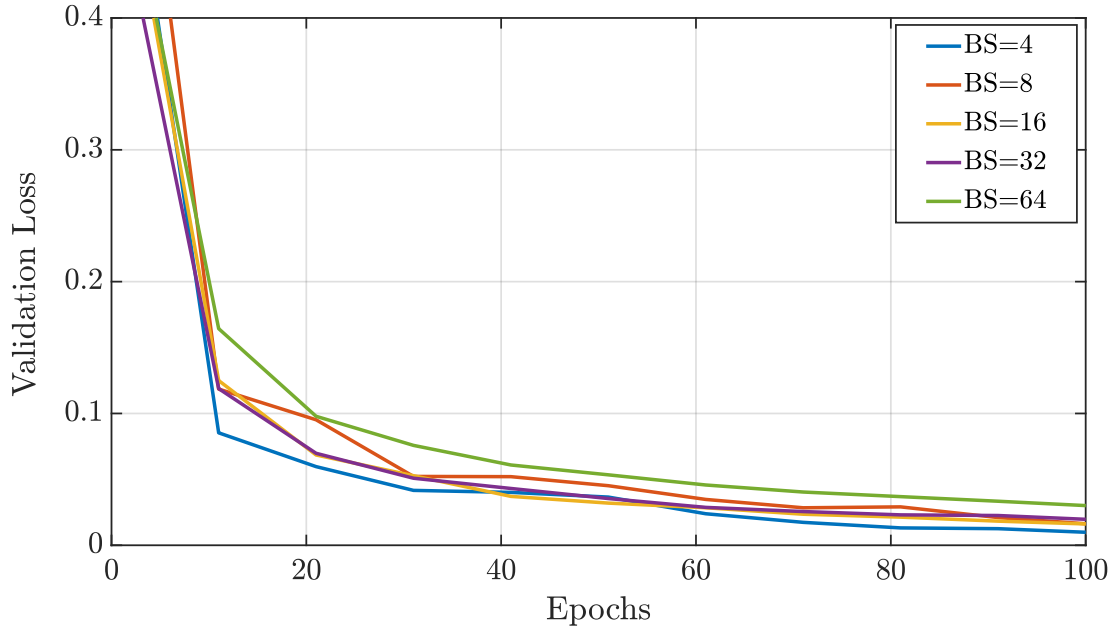


Figure 5.10: Validation losses with different mini-batch sizes.

cause over-fitting and much longer processing times.

Network weights and biases (W & b) are initialized randomly to break symmetry and ensure that various hidden units can learn different things. Since different initial weights (generating randomly) lead to different training and validation losses, for each network setting, the training process has been repeated three times, and then the average of losses and errors were reported.

Table 5.4 shows all the chosen training parameters for this model. Determining the values for sequence option parameters will be discussed in the next subsection.

5.4.4 Comparison of different time window sizes

For prediction of engine torque and engine speed in the near future, input features in the last segment, i.e., $[t_c - \Delta Z, t_c]$ are used when the driver is during the time period

Description	Value	Description	Value
Optimizer	ADAM	Mini batch size	4
Initial learn rate	0.01	Shuffle	never
Learn rate schedule	piecewise	Max epochs	500
Learn rate drop period	100	Sequence padding direction	right
Learn rate drop factor	0.9	Sequence padding value	0
Gradient Threshold	1	Sequence length	600

Table 5.4: Training parameters

$[t_c, t_c + \Delta t]$. t_c is the current time and the predictions are being made at $k\Delta t$, $k \in \mathcal{N}$ as discussed in section 5.2.2, $\Delta t = 1s$. The time window size of input features which is the focus of this section is ΔZ.

The effect of various time window sizes had been discussed in many papers relating to prediction either with LSTM network or other network types [59, 60, 67, 81, 82]. Features for estimation of outputs are extracted from the speed and acceleration profile, SOC, and torque demand in time window $[t_c - \Delta Z, t_c]$. In case of too small ΔZ, the time segment may not be large enough to hold relevant information, and in case of too large ΔZ, the time segment may include information that is no longer relevant. By changing the time window size in a proper range, the optimal ΔZ can be selected through a series of trials. The time window size can be changed through the 'sequence length' from training options, which can be a positive integer, longest, or shortest.

Sequence padding is a data processing that affects the output and training progress of the LSTM network. The training data sequences are sorted by their length, as can be seen in figure 5.11 to reduce the amount of padding or discarded data. Since

this model uses sequence-to-sequence LSTM layers, it is better to pad sequence data on the right to prevent padding in the very first time steps of the sequence, which can affect the predictions adversely. Also, a sample size of BS=8 (mini-batches) and $\Delta Z = 400$ (splits) along with created padding is illustrated on the sample sorted sequences graph to better understand the dataset and parameters.

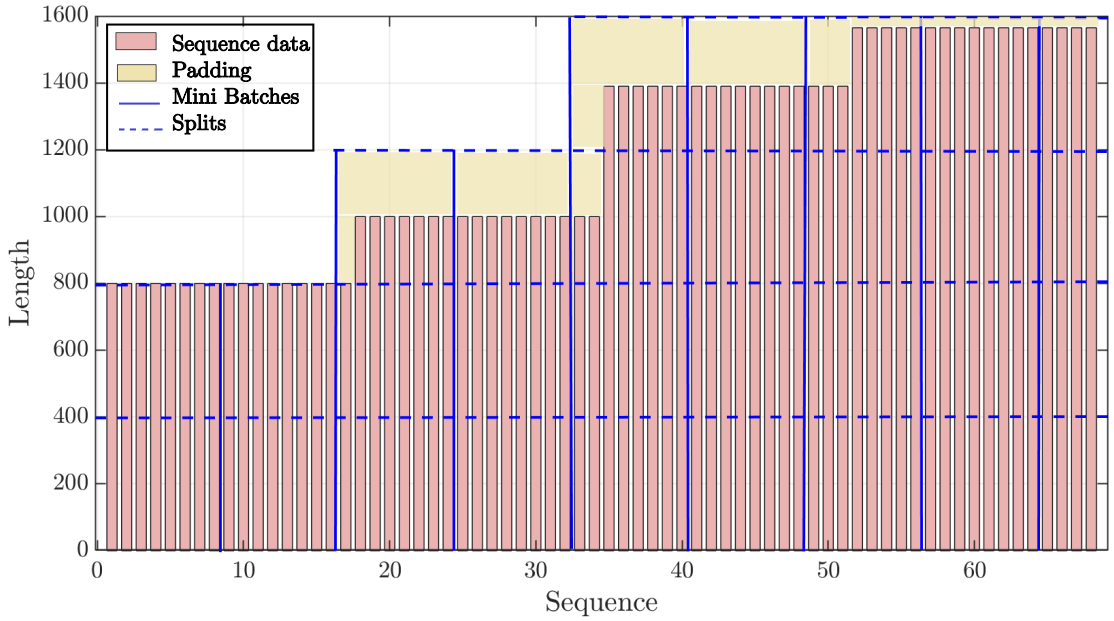


Figure 5.11: Sample of sequence data length, padding, mini-batches, and splits [adapted from [83]]

To investigate the effect of various time window sizes in training, seven time window sizes of 20, 50, 100, 200, 400, 600 seconds and the longest ΔZ, which in this case is the length of the longest drive cycle with a sequence length of 1565. The shortest ΔZ is 800 (S), which truncates at least half of the valuable training data. MAEs for engine speed and engine torque using different time window sizes are compared in figure 5.12. It can be concluded from the results that the network needs at least 100 (s) of historical data to achieve higher accuracy, and the optimum

time window size is $\Delta Z = 600(S)$. It can be assumed that the increase in MAEs for $\Delta Z = longest$ is due to the higher amount of padding on sequences with shorter lengths.

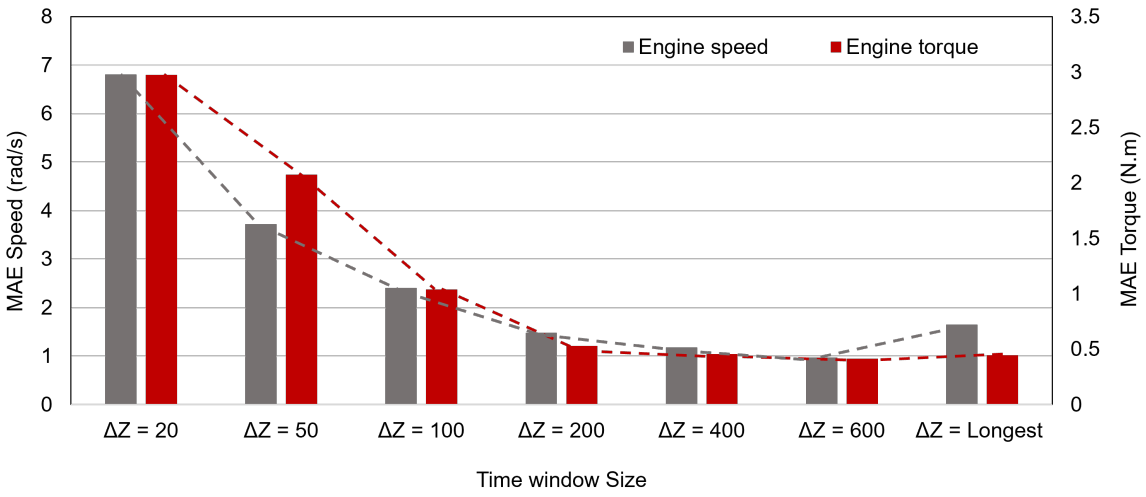


Figure 5.12: MAEs of predicted data based on different time window sizes.

5.5 Results and Conclusion

According to the simulation results and analysis for improving the performance of the LSTM multi-parameter prediction model, the final selected model has reached to 1.41 (rad/s) and 0.57 (N.m) RMSEs for ω_{ICE} and T_{ICE}, respectively. The validation and training loss for the first 100 epochs of the training is shown in figure 5.13. The final validation loss after 500 epochs reached 5.0033×10^{-4}. The validation loss will decrease further by increasing the number of epochs over 500. While increasing the epochs, the one should be aware of over fitting problem occurring in higher epochs. In such case, the training process needs to be stopped before validation loss starts to increase due to over fitting of weights on training dataset.

Besides the model's performance in the validation dataset, verifying the generalization capability and performance of the model with various drive cycles that have not been used in either training or testing is of high importance. In the last step of model verification, the network has been verified under US06 and LA92 (unified driving schedule) drive cycles which both have more aggressive driving behaviors (high acceleration and deceleration rates) with respect to UDDS and HWFET driving cycles.

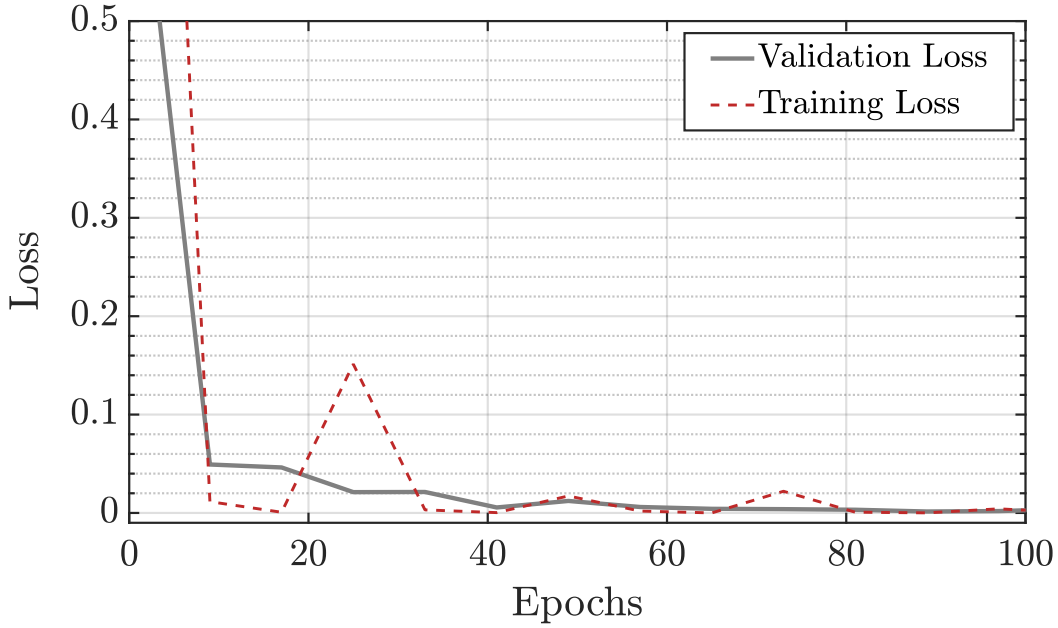
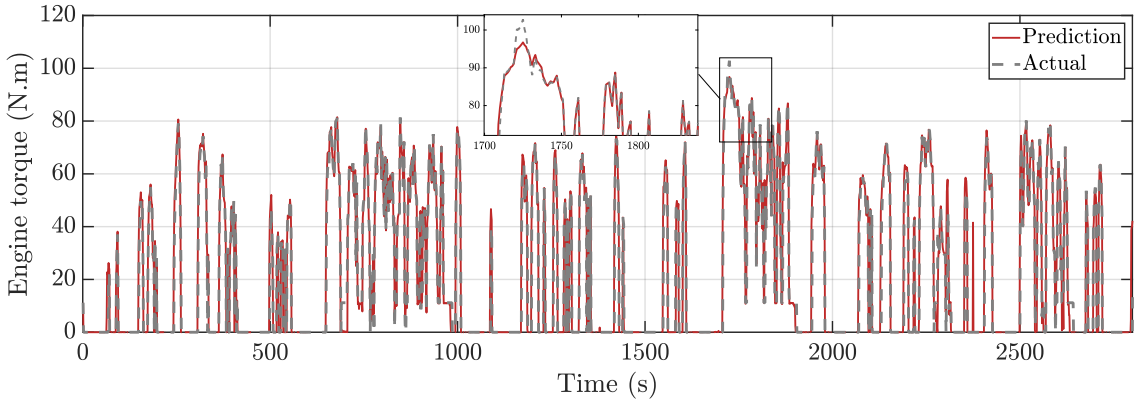
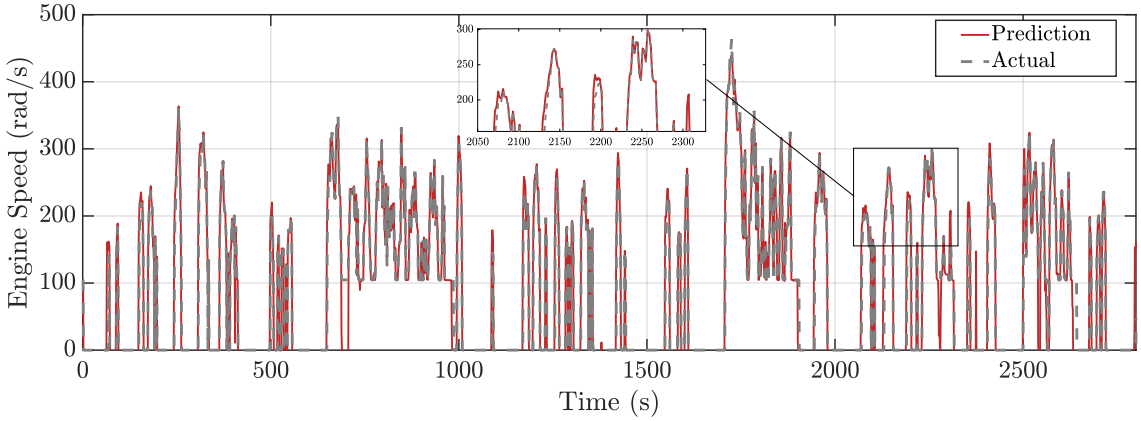


Figure 5.13: Training and validation loss of the LSTM prediction model.

For the generalization dataset , four of each US06 and LA92 with random initial SOCs have been chosen, and the actual output data from the HEV model is compared with LSTM prediction model outputs in figures 5.14 and 5.15. The total RMSE calculated for ω_{ICE} is 24.98 (rad/s), and for T_{ICE} is 6.72 (N.m), which are acceptable for the purpose of online EMS where this prediction model will be used.



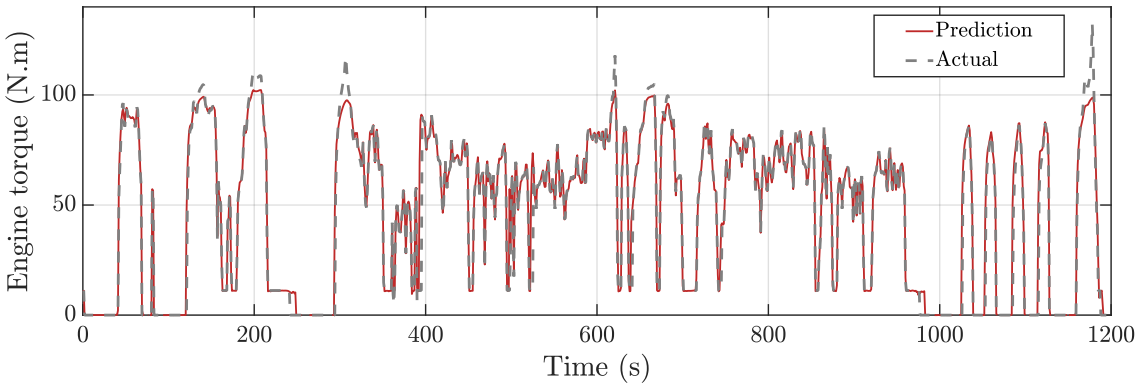
(a) Engine torque profile



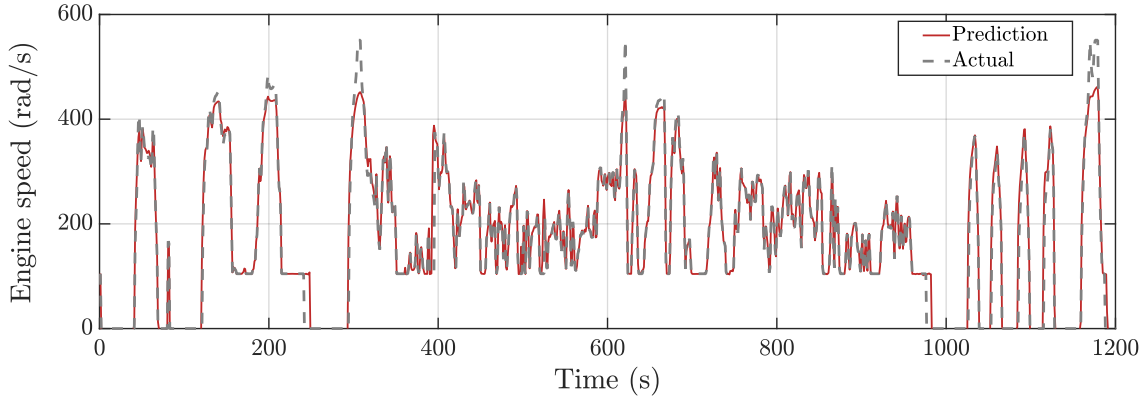
(b) Engine speed profile

Figure 5.14: LSTM prediction model outputs in comparison with actual data obtained from the HEV model under the LA92 drive cycle.

The predictions made in this chapter were using the whole drive cycle and input dataset at once, which was suitable for the purpose of training and improving the network performance. Moreover, the model's focus was to learn the driving trends and controller responses as effectively as possible. In the next chapter, the predictions will be made online as the goal is to perform an online EMS controller using the LSTM multi-parameter prediction model.



(a) Engine torque profile



(b) Engine speed profile

Figure 5.15: LSTM prediction model outputs in comparison with actual data obtained from the HEV model under the US06 drive cycle.

Chapter 6

Deep Recurrent Neural Network Based Energy Management Strategy

The fuel efficiency of HEVs can be improved by designing the EMS based on characteristics collected from previous driving data. Various studies have revealed that the driving cycle and driving style significantly impact fuel consumption and pollutants [84,85]. Neural network and machine learning algorithms have been employed to identify characteristics such as drive cycle recognition, traffic flow prediction, driving trend prediction from historical data, resulting in considerable improvements in fuel efficiency [67, 82, 86]. In [81], authors explained used existing on-board vehicle data to estimate road type and traffic congestion (RT&TC) and then used that knowledge to enhance vehicle power management. In a similar work [87], Murphey et al. used the machine-learning framework for RT&TC and driving trend estimation [67] to predict battery power and engine (ICE) speed to develop three online intelligent energy

controllers ultimately. In [88] and aforementioned studies, it can be observed that multiple NNs are being utilized to develop an online intelligent energy controller, and they are often trained with a high number of feature data from driving cycles and vehicle outputs.

This chapter investigates the process of implementing the developed offline LSTM prediction model (figure 6.1) in the previous chapter in an online format. The fuel consumption of the selected HEV using this controller is then compared to the previously developed rule-based EMS controller.

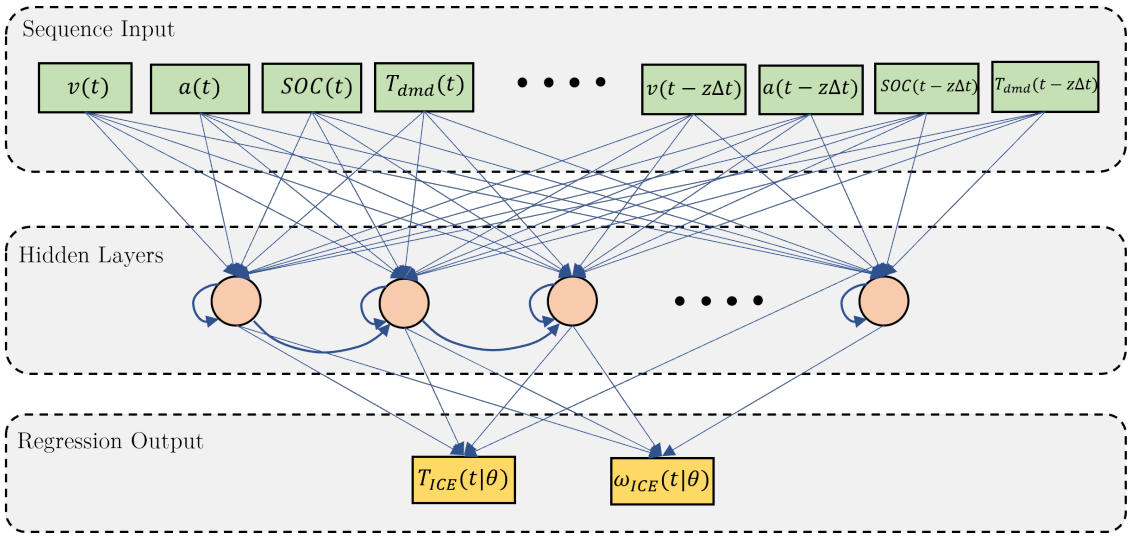


Figure 6.1: The structure of deep RNN used for energy management controller

6.1 Time series interpretation of EMS

The quasi-static model equations for HEV powertrain components were discussed in chapter 4. The problem of HEV energy management is a dynamic optimization

problem, [67] and the EMS can be expressed as a time series forecast in the discrete-time format:

$$u(t+1) = \Phi(x(t), x(t-1), \cdots, x(t-N), t) \tag{6.1.1}$$

where $u(t+1) = [\omega_{ICE}(t+1), T_{ICE}(t+1)]$ is the control variables array at sample time $t+1$ and $x(t) = [v(t), a(t), SOC(t), T_{dmd}(t)]$ is the state variables array. The nonlinear connection between the system states and the control variables is described by the function $\Phi()$.

To elaborate the equation 6.1.1, the relation between control variables and system states can be written as:

$$\begin{cases} T_{ICE}(t \mid \theta) = f(v(t), a(t), \mathrm{SOC}(t) \\ \qquad\qquad T_{dmd}(t), \ldots, v(t - z\Delta t), a(t - z\Delta t) \\ \qquad\qquad \mathrm{SOC}(t - z\Delta t), T_{dmd}(t - z\Delta t)) \\ \omega_{ICE}(t \mid \theta) = g(v(t), a(t), \mathrm{SOC}(t) \\ \qquad\qquad T_{dmd}(t), \ldots, v(t - z\Delta t), a(t - z\Delta t) \\ \qquad\qquad \mathrm{SOC}(t - z\Delta t), T_{dmd}(t - z\Delta t)) \end{cases} \tag{6.1.2}$$

where $T_{ICE}(t \mid \theta)$ and $\omega_{ICE}(t \mid \theta)$ are representing the predicted engine torque and speed at sample time t. Functions f and g are special functions which are being learned and updated by the developed LSTM deep RNN model. θ is representing weight parameters and activation functions in the LSTM network. Inputs of the special functions are the model inputs of $v(t), a(t), SOC(t), T_{dmd}(t)$ at current time and the historical data from previous time window showing. z is representing the time window size discussed in section 5.4.4 (described as ΔZ) and Δt is

the update time step which has set to 0.1 (s) same as in vehicle EMS modelling. $v(t-z\Delta t), a(t-z\Delta t), SOC(t-z\Delta t), T_{dmd}(t-z\Delta t)$ are denoting the input data at first time of historical time window.

Moreover, powertrain components (ICE, motor/generators, battery pack) constraints in the system must be considered while using the control variables. Inequalities should be applied to each component's minimum and maximum values.

$$\begin{cases} 0 \leq \omega_{ICE} \leq \omega_{ICE\_\max} \\ 0 \leq T_{ICE}(t) \leq T_{ICE\_\max}\left(\omega_{ICE}(t)\right) \\ \omega_{MG1\_\min} \leq \omega_{MG1} \leq \omega_{MG1\_\max} \\ T_{MG1\_\min}\left(\omega_{MG1}(t)\right) \leq T_{MG1}(t) \leq T_{MG1\_\max}\left(\omega_{MG1}(t)\right) \\ \omega_{MG2\_\min} \leq \omega_{MG2} \leq \omega_{MG2\_\max} \\ T_{MG2\_\min}\left(\omega_{MG2}(t)\right) \leq T_{MG2}(t) \leq T_{MG2\_\max}\left(\omega_{MG2}(t)\right) \end{cases} \tag{6.1.3}$$

Similar to the controller of the HEV (discussed in section 4.4), the rest of the control parameters can be obtained from power-split relations. The difference is that there are only two major modes of propulsion and braking, which are determined using the sign of T_{dmd} and the engine speed and torque will be obtained from the LSTM prediction model and given to the RNN-based controller. Table 6.1shows the equations used in the controller.

Control parameters	Propulsion	Braking
Engine Speed	$\omega_{ICE}(t \mid \theta)$	
Engine Torque	$T_{ICE}(t \mid \theta)$	
Engine Power	$P_{ICE} = T_{ICE}(t \mid \theta).\omega_{ICE}(t \mid \theta)$	
MG1 Torque	Obtained from 4.2.16	
MG1 Speed	Obtained from 4.2.13	
MG2 Torque	$(P_{dmd} - P_{MG1} - P_{ICE})/\omega_{MG2}$	$(g_3(v).P_{dmd})/\omega_{MG2}$
MG2 Speed	Obtained from 4.2.14	
Brake Power	$P_{brk} = 0$	$P_{brk} = P_{dmd} - P_{MG2}$

Table 6.1: Modes in RNN-based controller

6.2 Online implementation of RNN-based EMS

For the online implementation of the LSTM prediction model, the same Simulink model has been used for the driver and vehicle plant model, but the controller block has changed to the RNN-based controller. The trained optimal network based on analysis in the previous chapter has been loaded on the controller via a function. As mentioned in the previous chapter, for RNN-based controller, predictions are being made at each time step as the simulation runs based on previous time steps inputs.

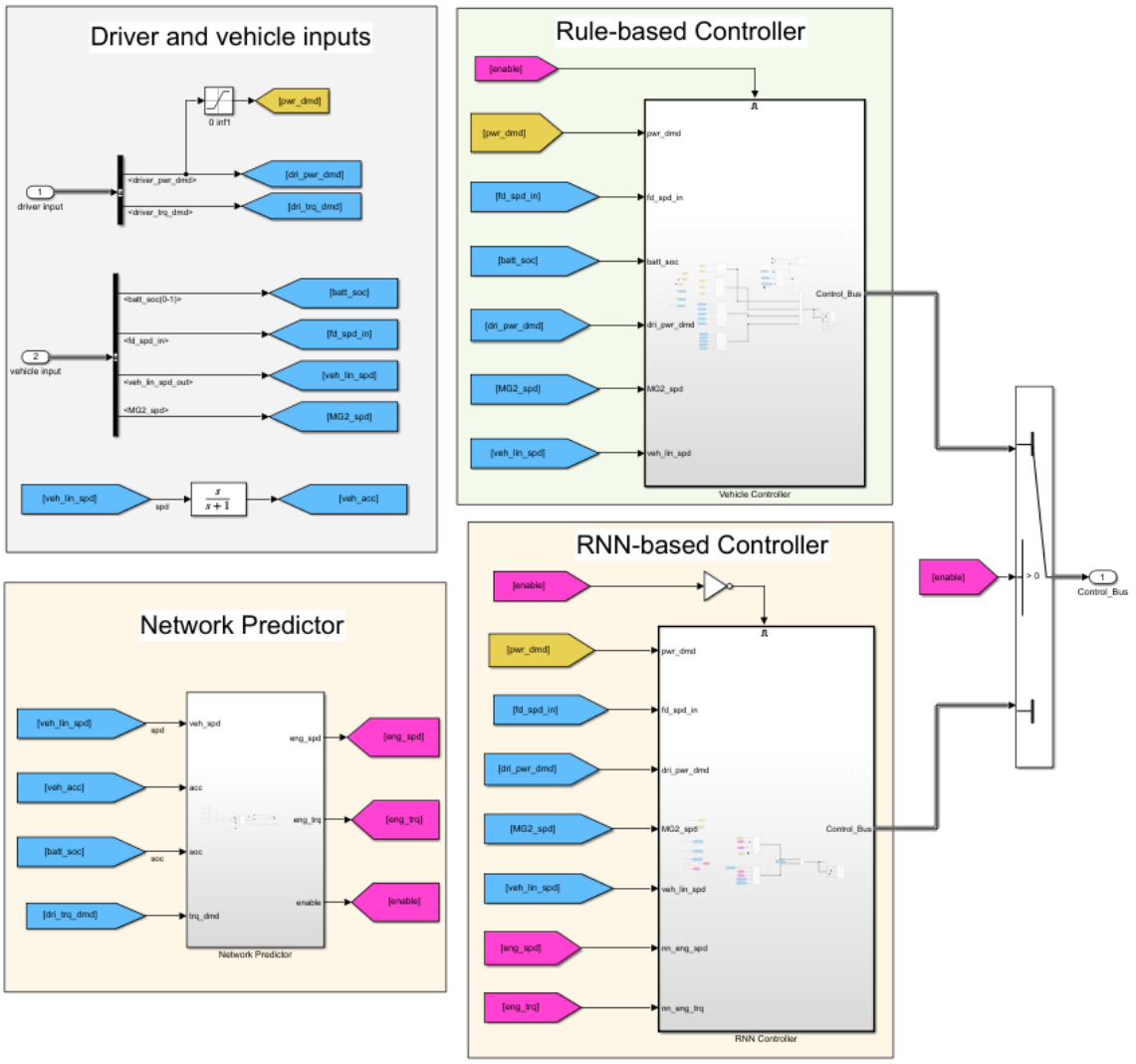


Figure 6.2: Layout of Vehicle controller modules in online implementation of RNN-based EMS

In order to maintain the fidelity of the HEV model, the prediction time step has been set to $\Delta t = 0.1(s)$. For the first sequence of input data from [0,$z\Delta t$], the vehicle uses the rule-based controller, and the network predictor module is updating the LSTM prediction model simultaneously. Based on the analysis, the updating time window has been set to one minute using $z = 600$ and Δt. Thus, from [60,

$60+k\Delta t]$, the network predictor module is estimating the $T_{ICE}(t \mid \theta)$ and $\omega_{ICE}(t \mid \theta)$ and constantly updating the network based on previous data. The outputs of this module are then being used in the RNN-based controller accordingly (see table 6.1). Figure 6.2shows various modules used in the HEV controller model to implement the RNN-based controller. The RNN-based controller, is an add-on to the previously implemented rule-based algorithm. The RNN used inside the Network Predictor module and utilized in RNN-based controller is the same as the final LSTM prediction model developed in Chapter 5.

6.3 Results and Discussion

Even though identifying characteristics of driving cycles and driving behaviors can improve the EMS performance, it is important to examine how the results of training an LSTM prediction model and implementing an RNN-based controller apply to the vehicle model. In this section, the developed RNN-based EMS will be applied to the HEV model, and the results will be compared to the rule-based EMS and dynamometer testing data.

For verification and generalization purposes, rather than previously mentioned drive cycles, two new drive cycles of NEDC (New European Drive Cycle) and Japanese JC08 drive cycle were also simulated with both rule-based and RNN-based EMS. The overall fuel consumption and end SOC for each drive cycle in both methods are then recorded.

Table 6.2 shows the comparison of using RNN-based and rule-based methods performance on the vehicle model. In the results of this table, the initial SOC for UDDS is 34%, HWFET 32%, Steady state 39%, and for the rest of the drive cycles

is 40%. Following equations were used to obtain the ΔFC and ΔSOC (FC stands for fuel consumption)

$$\Delta FC = \frac{FC_{RNN} - FC_{RB}}{FC_{RB}} \times 100 \tag{6.3.1}$$

$$\Delta SOC = SOC_{RNN_{end}} - SOC_{RB_{end}} \tag{6.3.2}$$

Drive Cycles	Fuel Consumption (g)			End SOC (%)		
	RNN-based	Rule-based	ΔFC	RNN-based	Rule-based	ΔSOC
UDDS	382.8	402.3	-4.85	34	36.23	-2.23
HWFET	1050	1183	-11.24	39.19	43.69	-4.5
US06	1343	1399	-4.00	43.84	44.66	-0.82
Steady state	375.6	388.4	-3.30	34.41	37.24	-2.83
LA92	628.8	641.2	-1.93	43.54	44.29	-0.75
NEDC	328.2	368.6	-10.96	39.69	44.1	-4.41
JC08	247.4	250.9	-1.39	37.29	38.48	-1.19
WLTC	812.9	852.4	-4.63	43.51	44.34	-0.83

Table 6.2: RNN-based vs. Rule-based EMS results

From the results in table 6.2, it can be seen that, as expected, the fuel consumption using RNN-based controller has decreased slightly. The fuel consumption reduction is more in drive cycles with higher average speeds (HWFET and NEDC), and it is

mostly due to the reduction in idling time of the engine. In a rule-based approach, only a certain speed and power threshold were responsible for changing the engine idling status in urban and highway driving situations, which was not completely efficient and robust. It can also be observed that the ending SOC in RNN-based EMS is lower than the rule-based approach, which is consistent with the decrease in fuel consumption amounts. However, the frequent charging and discharging of the battery, which happens more often in the RNN-based method, can degrade the battery performance and state of health over time.

Figure 6.3 better illustrates the difference in fuel consumption with both methods on tested drive cycles. Also, table 6.3 shows the comparison between vehicle fuel consumption reported in dynamometer test data and fuel consumption using an RNN-based controller. The results are in line with the conclusions drawn from table 6.2.

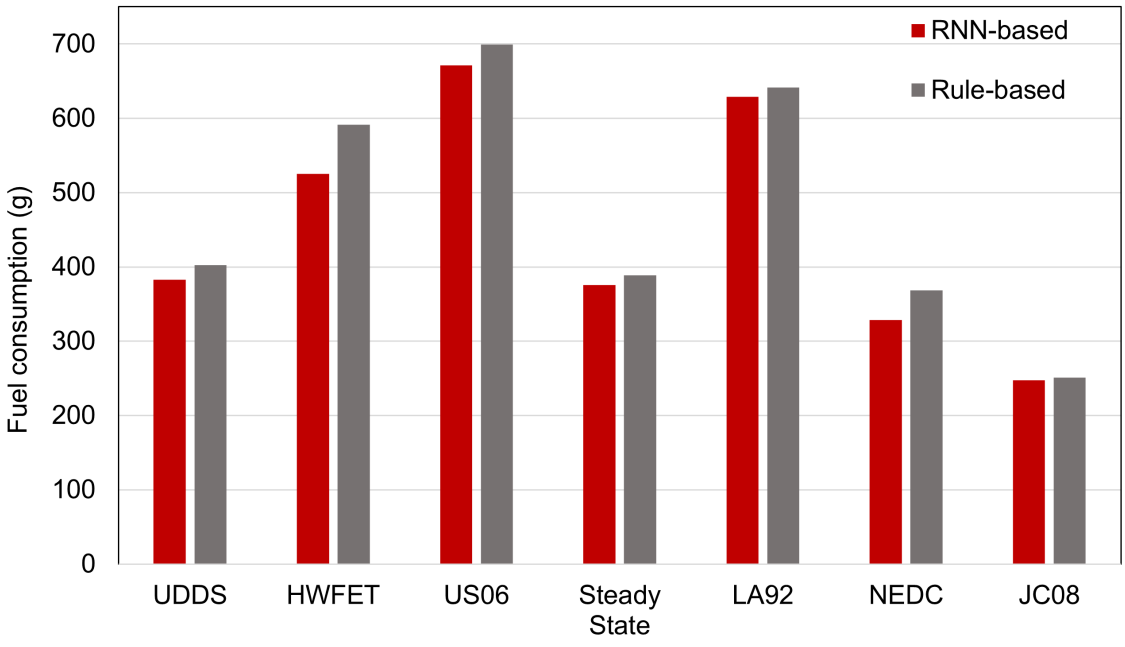


Figure 6.3: Results of fuel consumption obtained by rule-based and RNN-based methods on HEV vehicle model

Drive cycles	Fuel Consumption (g)			End SOC (%)	
	RNN-based	Test	ΔFC	RNN-based	Test
UDDS	382.8	403.5	-5.13	34	35
HWFET	1050	1227	-14.43	39.19	40
US06	1343	1459	-7.95	43.84	40
Steady State	375.6	388.9	-3.42	34.41	37

Table 6.3: RNN-based EMS vs. actual dynamometer test results

Following figures 6.4, 6.5, and 6.6 show battery SOC, engine power, and MG2 power profiles along with vehicle speed obtained from the HEV model using both rule-based and RNN-based EMS.

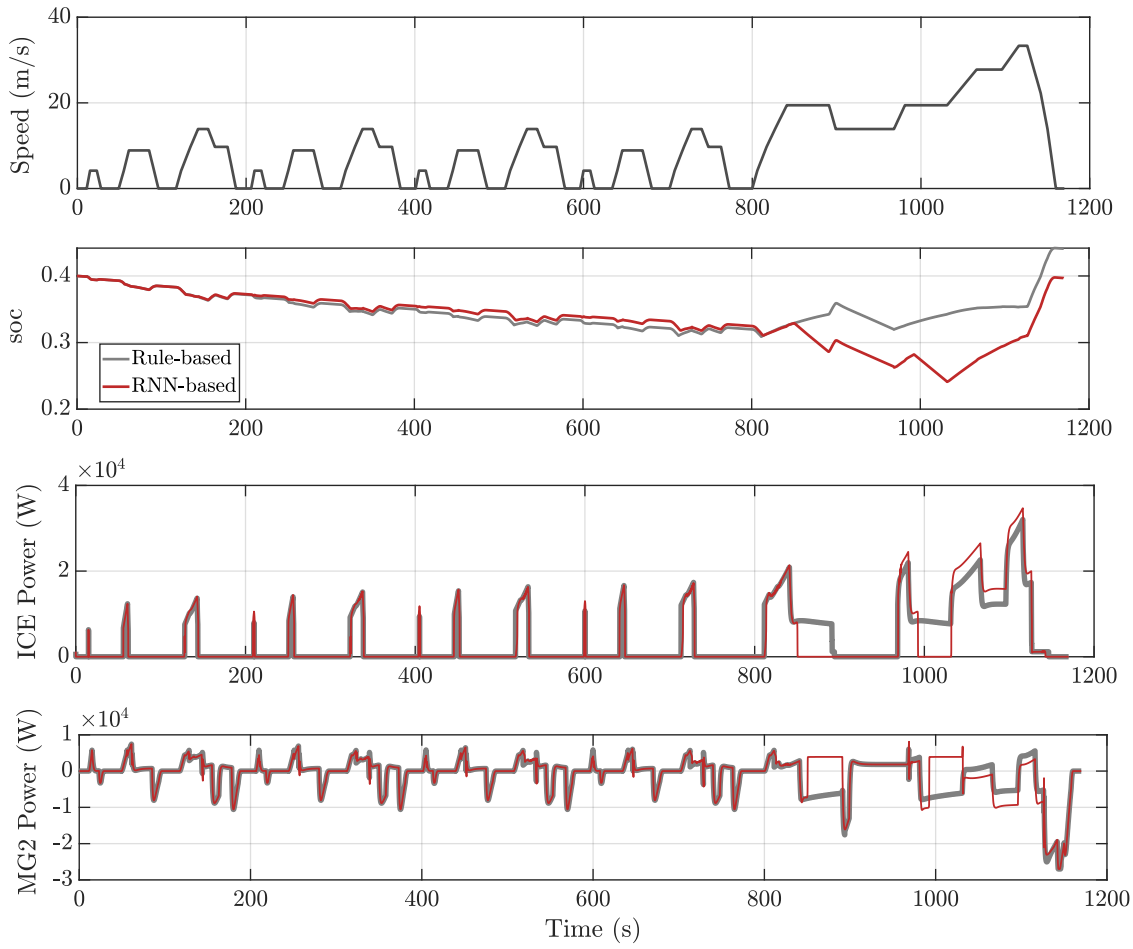


Figure 6.4: Results of battery SOC, mechanical power of Engine and MG2 obtained by rule-based and RNN-based methods on HEV vehicle model for NEDC drive cycle

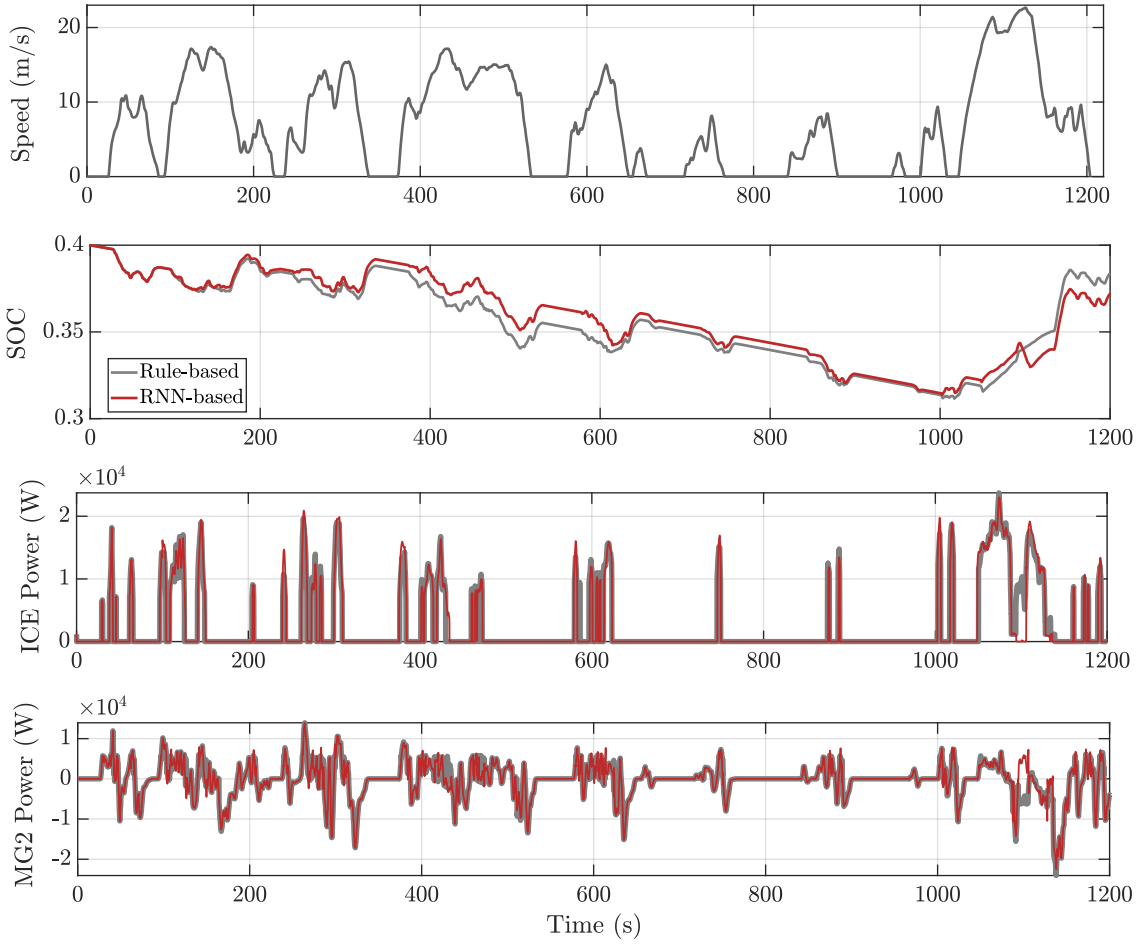


Figure 6.5: Results of battery SOC, mechanical power of Engine and MG2 obtained by rule-based and RNN-based methods on HEV vehicle model for JC08 drive cycle

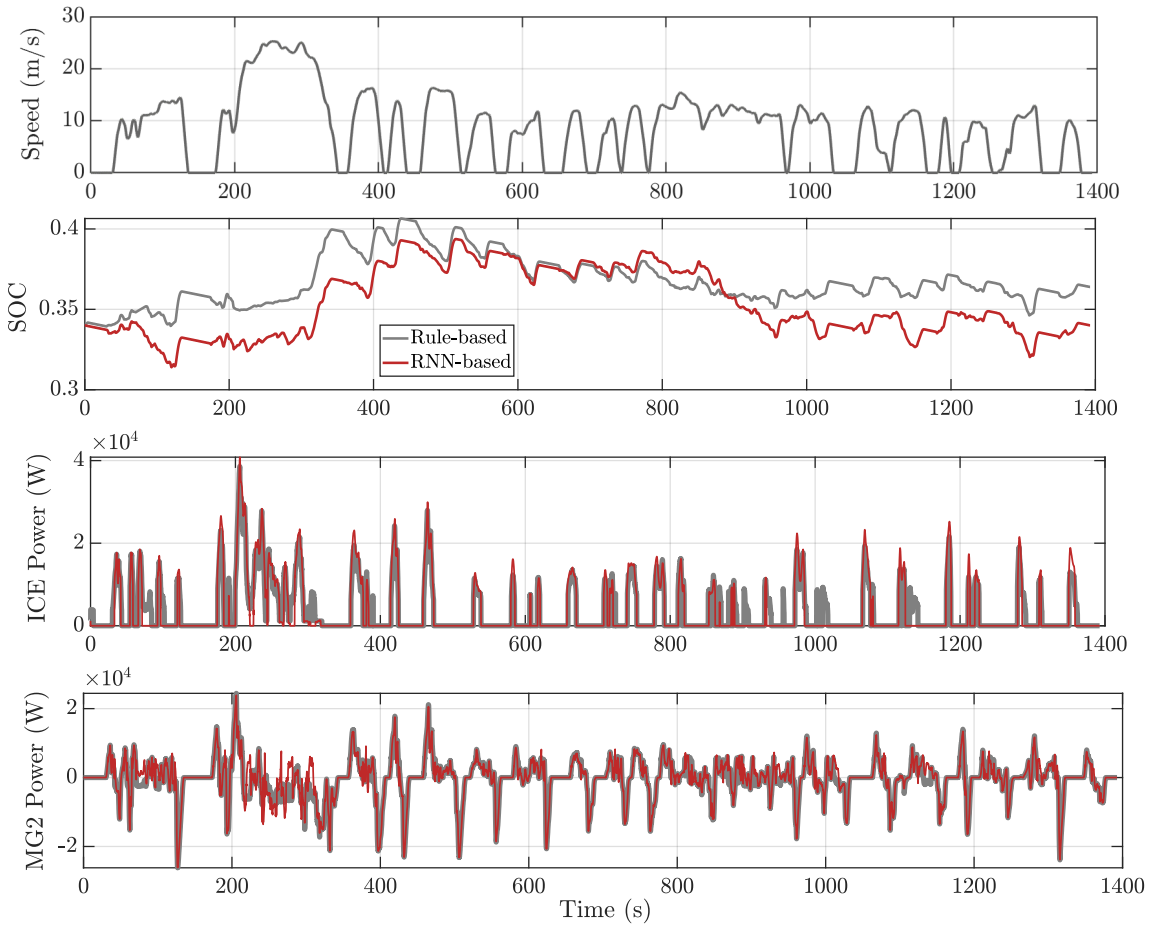


Figure 6.6: Results of battery SOC, mechanical power of Engine and MG2 obtained by rule-based and RNN-based methods on HEV vehicle model for UDDS drive cycle

Chapter 7

Conclusions and Future Work

In this chapter, an overview of the research findings is presented. Following that, the contributions of this thesis are outlined. The current research's potential future work is then described, with suggestions for improvements in several aspects.

7.1 Summary

The literature review chapters present a general overview of electrification and hybrid electric powertrains and the motivations and objectives of supervisory control. Then, state-of-the-art energy management strategies are reviewed in chapter three. Each EMS method is evaluated in terms of its benefits and drawbacks. Also, the underlying reasons for choosing a learning-based EMS are explained.

To begin the main work, an HEV model, in this case, Toyota Prius 2010, is created based on online available test data. Model parameters were obtained from various resources, reverse engineering techniques, and by calibrating the model outputs with respect to the actual dynamometer test data. The developed model consists of three

main subsystems: driver, system-level controller, and vehicle model. A rule-based EMS was developed for the system-level controller to imitate the actual vehicle behavior in power flow and energy consumption. Model evaluation results were found to be in accordance with the dynamometer data. For example, model and test data fuel consumption and fuel flow errors for the UDDS driving cycle are -0.29% and 0.423 g/s, respectively. As a result, the HEV model achieved an acceptable accuracy for capturing the model's performance with new energy management strategies and ensuring that they can be replicated in the actual vehicle.

In the next step, an LSTM multi-parameter prediction model is introduced and developed. First, an explanation for choosing recurrent neural networks over traditional neural networks and the reasons for developing the model with LSTM network are discussed. Data for training the network were chosen carefully to agree with the main goal of learning the driving behaviors and control modes with the network. During the data processing and training, an improvement in results was observed with the standard score normalization method over the min and max normalization approach. Based on the explained methodology, the network performance is improved through four iterative steps: define, compile & fit, verify, and predict. Simulations on ten different network structures varying from one hidden layer to five hidden layers with different training to testing proportion sizes were performed. Through multiple simulations and analyses, some of the important model hyper-parameters are tuned to improve the accuracy of predictions. For a better understanding of model accuracy enhancement, the first trained network during the development of this model had validation RMSEs of 8.93 (rad/s) and 2.92 (N.m) for engine speed and torque, respectively. After the analysis, the final selected network achieved 1.41 (rad/s) and

0.57 (N.m) validation RMSEs, which shows an average reduction of 80% in validation RMSEs.

In the final chapter, the deep RNN-based energy management strategy is defined and explained. The main goal is to employ the developed LSTM multi-parameter prediction model in the system-level controller of the HEV. Being a type of learning-based EMS, deep RNN-based EMS has the capability of real-time implementation. Due to its learning ability, it can adapt the control variables based on driving characteristics and historical data. In this regard, a time-series interpretation of EMS is given. Engine speed and torque are being predicted in the current time based on the network updated weights and biases from previous time window. Deep RNN-based controller kicks off when the first time window (one minute) passes by. Integrating network predictor functions in MATLAB with Simulink environment for online running the controller and model was the challenging part of this chapter. The results showed that the employment of deep RNN-based leads to an average 5.29% fuel consumption reduction.

7.2 Contributions

There are three major contributions to this research. This thesis proposes an LSTM multi-parameter prediction model, which takes the vehicle speed, acceleration, battery SOC, and torque demand to learn engine speed and engine torque of a power-split HEV model.

The proposed model is different from most of the previous work in terms of taking sequential and historical data as network input to capture the characteristics of both driver and controller behaviors and improve the prediction accuracy. Model inputs

are limited to four parameters to create an end-to-end framework with the ability to predict control variables without extracting features of drive cycles.

Moreover, being a multi-parameter model for predicting two of the power-split powertrain parameters with only one deep neural network makes this work novel. In previous relevant studies, proposed models either predict one of the control variables or use multiple networks to predict multiple variables.

The sensitivity analysis and debugging of model hyper-parameters and simulating various network structures in this work have not been investigated before to this extent for power-split EMS purposes. Many studies with similar analyses are in the SOC estimation field. Also, the achieved validation RMSE of model outputs are lower than the ones achieved in similar studies using deep RNN networks in HEV powertrain parameters prediction.

The deep RNN-based EMS based on the trained LSTM prediction model is implemented in real-time. Driver's intents are being fed to the trained network, and the network predictor module responds to the driver's intentions promptly. The fuel economy of the selected HEV is increased using an RNN-based controller by only learning the driving behaviors.

This work can be used as a modular framework for other studies developing the RNN-based controller. The LSTM multi-parameter prediction model can be trained using optimal EMS data such as dynamic programming or convex optimizations and then used as the deep RNN-based controller to further improve the fuel economy.

7.3 Future Work

7.3.1 Advanced Hybrid Electric Vehicle Modeling

The HEV model presented in chapter 4, uses quasi-static equations for most of the vehicle components. Achieving higher fidelity of the model is possible through consideration of the dynamics of each component. Taking 2D or 3D vehicle dynamics into account instead of longitudinal dynamics considered herein would improve the model accuracy to the real-world conditions. Moreover, developing component's thermal models can capture temperature effects in the vehicle model. In particular, battery pack and internal combustion engine models can be improved by adding temperature-dependent variables and their relations to the other subsystems.

7.3.2 Data Correlation Analysis

In the LSTM multi-parameter prediction model, four inputs are chosen among tens of HEV parameters to predict control variables. Selecting the appropriate parameters for model input ensures a better training effect and decreases the probability of over-fitting. This is because many of the parameters have strong or irrelevant connections. For this matter, correlation analysis on HEV parameters can be done to choose the most appropriate input parameters and eliminate the destructive ones. The Pearson correlation coefficient (PCC) is mostly used in relevant research studies.

7.3.3 Model and Training Hyper-parameters Analysis

A few important model-related and training-related hyper-parameters are discussed and analyzed in this work. However, there are still many parameters and aspects that

can be further investigated. Among training-related parameters, a higher number of epochs can be used and analyzed for validation RMSE and computational effort. Gradient threshold method and regularization can also be examined further.

One of the effective methods to improve the training performance and prevent overfitting is dropout or pre-dropout methods. When training data is not enough, usually overfitting happens. Dropout layers with different probabilities from (0 0.5] can be added to the network structure to preserve the most advantageous neurons and connections and stabilize the training process.

Lastly, the employment of the k-fold (5-fold or 10-fold) cross-validation method was one of the initial intentions in comparing different hyper-parameters in this work which could not be completed. Hence, a contribution can be made towards a more effective analysis of hyper-parameters using the k-fold cross-validation method in future.

7.3.4 Optimal Deep RNN-based EMS

The purpose of this work was to demonstrate the effect of using historical driving data in the energy management controller of an HEV vehicle, and the rule-based strategy was an efficient path to evaluate this. However, the LSTM model and RNN-based controller are designed in a way that data of various EMS methods can be easily developed by them. An offline optimization method such as dynamic programming or convex optimization can be carried out to achieve optimal/sub-optimal fuel consumption with the deep RNN-based EMS. Then, the LSTM multi-parameter prediction model can be trained using the optimized control variables. The online implementation of such an optimal trained network can lead to lower fuel consumption and

emissions while having the advantage of real-time implementation.

7.3.5 Model Verification with Driving Simulator

A valuable future work would be towards testing the deep RNN-based EMS in a virtual environment such as a driving simulator. The driving simulator will include the driver in the model verification process (driver-in-the-loop). Testing the EMS with a driver can reveal the model flaws in vehicle performance because every harsh acceleration or deceleration made by the controller can be detected. In this regard, a collaboration with the MARC Drive lab at McMaster Automotive Resource center can be made.

Appendix A

Codes and Functions

In this section, snapshots of some parts of the codes in MATLAB and Simulink is presented.

```
%% Network Structure

numResponses = size(YTrain{1},1);
featureDimension = size(XTrain{1},1);
numHiddenUnits1 = 80;
numHiddenUnits2 = 30;
numHiddenUnits3 = 10;
layers = [ ...
    %Sequence Layer
    sequenceInputLayer(featureDimension)
    %First hidde layer
    lstmLayer(numHiddenUnits1,'OutputMode','sequence')
    tanhLayer
    %Second hidde layer
    lstmLayer(numHiddenUnits2,'OutputMode','sequence')
    tanhLayer
    %Third hidden layer
    lstmLayer(numHiddenUnits3,'OutputMode','sequence')
    tanhLayer
    %Output layers
    fullyConnectedLayer(numResponses)
    regressionLayer];
```

Figure A.1: LSTM multi-parameter prediction network structure

```
options = trainingOptions('adam' , ...
    'MaxEpochs',500, ...
    'MiniBatchSize',4, ...
    'InitialLearnRate',0.01, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',100, ...
    'LearnRateDropFactor',0.9, ...
    'GradientThreshold',1, ...
    'Plots','training-progress',...
    'shuffle','never',...
    'Verbose',0 , ...
    'ValidationData',{XTest,YTest},...
    'ValidationFrequency',420,...
    'SequenceLength',600,...
    'SequencePaddingDirection','right',...
    'SequencePaddingValue', 0 ,...
    'ExecutionEnvironment','cpu');
```

Figure A.2: LSTM network training options

```
function yhat = my_update_predict_fcn(lastvalue)

oldnet = evalin('base', 'net');
[newnet, yhat] = predictAndUpdateState(oldnet, lastvalue);
assignin('base', 'net', newnet);

end
```

Figure A.3: Function used for updating the network state and prediction of parameters

```
function [engspd,engtrq,enable] = fcn(spd, acc, soc, trq, time)

X = [spd ;acc;soc;trq];

coder.extrinsic('my_update_predict_fcn');

if time<=60

    engspd=0;
    engtrq=0;
    enable=1;

else

    enable=0;
    YPred=[single(0);single(0)];
    YPred = my_update_predict_fcn(X);
    engspd=double(YPred(1));
    engtrq=double(YPred(2));
end

end
```

Figure A.4: Functions used in network predictor module of online controller

References

[1] J. D. Townsend and R. J. Calantone, "Evolution and transformation of innovation in the global automotive industry," *Journal of product innovation management*, vol. 31, no. 1, pp. 4–7, 2014. (Cited on p. 1.)

[2] "Mercedes AMG F1 engine achieves 50 percent thermal efficiency," [Accessed: 2021-09-21]. [Online]. Available: https://www.motorauthority.com/news/1112999_mercedes-amg-f1-engine-achieves-50-percent-thermal-efficiency (Cited on p. 2.)

[3] "Sources of Greenhouse Gas Emissions," [Accessed: 2021-09-21]. [Online]. Available: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions (Cited on p. 2.)

[4] M. Ehsani, Y. Gao, S. Longo, and K. M. Ebrahimi, *Modern electric, hybrid electric, and fuel cell vehicles*. CRC press, 2018. (Cited on pages 2, 9, 12, 13, and 15.)

[5] "IHS Markit Powertrain Production Forecast," [Accessed: 2021-09-21]. [Online]. Available: https://ihsmarkit.com/index.html (Cited on p. 2.)

[6] "Insights into the new energy vehicle market of 2030," [Accessed: 2021-09-21]. [Online]. Available: https://www.dsm.com/engineering-materials/en_US/blog/insights-into-the-new-energy-vehicle-market-2030.html (Cited on pages xi and 3.)

[7] B. Bilgin, J. W. Jiang, and A. Emadi, *Switched reluctance motor drives: fundamentals to applications*. CRC press, 2019. (Cited on p. 4.)

[8] K. P. Divakarla, S. G. Wirasingha, A. Emadi, and S. Razavi, "Artificial neural network based adaptive control for plug-in hybrid electric vehicles," *International Journal of Electric and Hybrid Vehicles*, vol. 11, no. 2, pp. 127–151, 2019. (Cited on p. 5.)

[9] A. Emadi, *Advanced electric drive vehicles*. CRC Press, 2014. (Cited on p. 8.)

[10] A. Emadi, Young Joo Lee, and K. Rajashekara, "Power Electronics and Motor Drives in Electric, Hybrid Electric, and Plug-In Hybrid Electric Vehicles," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2237–2245, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4493430 (Cited on p. 8.)

[11] 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. (Cited on p. 9.)

[12] B. Bilgin, P. Magne, P. Malysz, Y. Yang, V. Pantelic, M. Preindl, A. Korobkine, W. Jiang, M. Lawford, and A. Emadi, "Making the case for electrified transportation," *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1,

pp. 4–17, 2015. (Cited on p. 9.)

[13] S. Wirasingha, M. Khan, and O. Gross, "48-v electrification: Belt-driven starter generator systems," *Advanced Electric Drive Vehicles*, 2014. (Cited on pages xv and 10.)

[14] 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. (Cited on pages xv, 10, 12, and 18.)

[15] J. Liu and H. Peng, "A systematic design approach for two planetary gear split hybrid vehicles," *Vehicle System Dynamics*, vol. 48, no. 11, pp. 1395–1412, 2010. (Cited on p. 10.)

[16] P. G. Anselma, Y. Huo, J. Roeleveld, A. Emadi, and G. Belingardi, "Rapid optimal design of a multimode power split hybrid electric vehicle transmission," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, no. 3, pp. 740–762, 2019. (Cited on p. 10.)

[17] M. Sabri, K. A. Danapalasingam, and M. F. Rahmat, "A review on hybrid electric vehicles architecture and energy management strategies," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 1433–1442, 2016. (Cited on p. 12.)

[18] S. Mohr, J. Wang, G. Ellem, J. Ward, and D. Giurco, "Projection of world fossil fuels by country," *Fuel*, vol. 141, pp. 120–135, 2015. (Cited on p. 12.)

[19] "The epa automotive trends report," United States Environmental Protection Agency (EPA), Tech. Rep., 2020. (Cited on pages 12 and 14.)

[20] "CAFE Public Information Center," [Accessed: 2021-07-12]. [Online]. Available: https://ecfr.io/Title-49/Section-531.5 (Cited on p. 13.)

[21] "Automotive Trends Report from EPA," [Accessed: 2021-07-12]. [Online]. Available: https://www.epa.gov/automotive-trends/explore-automotive-trends-data (Cited on pages 13 and 14.)

[22] "vehicle emission standard," [Accessed: 2021-07-13]. [Online]. Available: https://en.wikipedia.org/wiki/Vehicle_emission_standard (Cited on p. 14.)

[23] "Epa and nhtsa set standards to reduce greenhouse gases and improve fuel economy for model years 2017-2025 cars and light trucks," United States Environmental Protection Agency (EPA), Tech. Rep., 2012. (Cited on p. 14.)

[24] "Stellantis intensifies electrification while targeting sustainable double-digit adjusted operating income margins in the mid-term," [Accessed: 2021-07-13]. [Online]. Available: https://www.stellantis.com/en/news/press-releases/2021/july/stellantis-intensifies-electrification-while-targeting-sustainable-double-digit-adjusted-operating (Cited on p. 14.)

[25] "General motors, the largest u.s. automaker, plans to be carbon neutral by 2040," [Accessed: 2021-07-13]. [Online]. Available: https://media.gm.com/media/us/en/gm/home.detail.html/content/Pages/news/us/en/2021/jan/0128-carbon.html (Cited on p. 14.)

[26] T. Katrasnik, S. Rodman, F. Trenc, A. Hribernik, and V. Medica, "Improvement of the dynamic characteristic of an automotive engine by a turbocharger assisted

by an electric motor," *J. Eng. Gas Turbines Power*, vol. 125, no. 2, pp. 590–595, 2003. (Cited on p. 15.)

[27] M. Valentine-Urbschat, W. Bernhart *et al.*, "Powertrain 2020–the future drives electric," *Roland Berger Strategy Consultants*, vol. 9, pp. 17–19, 2009. (Cited on p. 15.)

[28] D.-D. Tran, M. Vafaeipour, M. El Baghdadi, R. Barrero, J. Van Mierlo, and O. Hegazy, "Thorough state-of-the-art analysis of electric and hybrid vehicle powertrains: Topologies and integrated energy management strategies," *Renewable and Sustainable Energy Reviews*, vol. 119, p. 109596, 2020. (Cited on pages xi, 17, 18, 19, 24, and 25.)

[29] W. Enang and C. Bannister, "Modelling and control of hybrid electric vehicles (a comprehensive review)," *Renewable and Sustainable Energy Reviews*, vol. 74, pp. 1210–1239, 2017. (Cited on pages 18, 20, and 21.)

[30] S. F. Tie and C. W. Tan, "A review of energy sources and energy management system in electric vehicles," *Renewable and sustainable energy reviews*, vol. 20, pp. 82–102, 2013. (Cited on p. 19.)

[31] 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. (Cited on pages 19 and 21.)

[32] A. Sciarretta and L. Guzzella, "Control of hybrid electric vehicles," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 60–70, 2007. (Cited on p. 19.)

[33] G. Shi, Y. Jing, A. Xu, and J. Ma, "Study and simulation of based-fuzzy-logic parallel hybrid electric vehicles control strategy," in *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 1. IEEE, 2006, pp. 280–284. (Cited on p. 19.)

[34] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, "Model predictive control power management strategies for hevs: A review," *Journal of Power Sources*, vol. 341, pp. 91–106, 2017. (Cited on pages 19 and 22.)

[35] C.-C. Lin, H. Peng, J. W. Grizzle, and J.-M. Kang, "Power management strategy for a parallel hybrid electric truck," *IEEE transactions on control systems technology*, vol. 11, no. 6, pp. 839–849, 2003. (Cited on p. 21.)

[36] G. Paganelli, T. M. Guerra, S. Delprat, J.-J. Santin, M. Delhom, and E. Combes, "Simulation and assessment of power control strategies for a parallel hybrid car," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 214, no. 7, pp. 705–717, 2000. (Cited on p. 21.)

[37] B. Gu and G. Rizzoni, "An adaptive algorithm for hybrid electric vehicle energy management based on driving pattern recognition," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 47683, 2006, pp. 249–258. (Cited on p. 21.)

[38] 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. 4-5, pp. 509–524, 2005. (Cited on p. 21.)

[39] S. Onori, L. Serrao, and G. Rizzoni, "Adaptive equivalent consumption minimization strategy for hybrid electric vehicles," in *Dynamic Systems and Control Conference*, vol. 44175, 2010, pp. 499–505. (Cited on p. 22.)

[40] A. Sciarretta, L. Guzzella, and M. Back, "A real-time optimal control strategy for parallel hybrid vehicles with on-board estimation of the control parameters," *IFAC Proceedings Volumes*, vol. 37, no. 22, pp. 489–494, 2004. (Cited on p. 22.)

[41] J. Tang, L. Guo, B. Gao, Q. Liu, S. Yu, and H. Chen, "Energy management of a parallel hybrid electric vehicle with cvt using model predictive control," in *2016 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 4396–4401. (Cited on p. 22.)

[42] D. F. Opila, X. Wang, R. McGee, and J. Grizzle, "Real-time implementation and hardware testing of a hybrid vehicle energy management controller based on stochastic dynamic programming," *Journal of dynamic systems, measurement, and control*, vol. 135, no. 2, p. 021002, 2013. (Cited on p. 22.)

[43] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, p. 38, 2016. (Cited on p. 23.)

[44] V. A. Katkar and P. Goswami, "Review on energy management systems for hybrid e vehicles," in *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. IEEE, 2020, pp. 1–6. (Cited on p. 23.)

[45] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*. PWS Publishing Co., 1997. (Cited on p. 23.)

[46] T. Yi, Z. Xin, Z. Liang, and Z. Xinn, "Intelligent energy management based on driving cycle identification using fuzzy neural network," in *2009 Second International Symposium on Computational Intelligence and Design*, vol. 2. IEEE, 2009, pp. 501–504. (Cited on p. 23.)

[47] M. Suzuki, S. Yamaguchi, T. Araki, P. Raksincharoensak, M. Yoshizawa, and M. Nagai, "Fuel economy improvement strategy for light duty hybrid truck based on fuel consumption computational model using neural network," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 719–10 725, 2008. (Cited on p. 23.)

[48] D. V. Prokhorov, "Toyota prius hev neurocontrol and diagnostics," *Neural Networks*, vol. 21, no. 2-3, pp. 458–465, 2008. (Cited on p. 24.)

[49] R. Xiong, J. Cao, and Q. Yu, "Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle," *Applied energy*, vol. 211, pp. 538–548, 2018. (Cited on p. 24.)

[50] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," in *INCOSE international symposium*, vol. 1, no. 1. Wiley Online Library, 1991, pp. 57–65. (Cited on p. 26.)

[51] "Argonne National Laboratory Hybrid Electric Vehicle Testing," [Accessed: 2021-08-22]. [Online]. Available: https://www.anl.gov/es/hybrid-electric-vehicle-testing (Cited on pages 27, 41, 45, and 54.)

[52] T. Gray and M. Shirk, "2010 toyota prius vin 6063 hybrid electric vehicle battery test results," 1 2013. [Online]. Available: https://www.osti.gov/biblio/1073780 (Cited on p. 29.)

[53] M. Olszewski *et al.*, "Evaluation of the 2010 toyota prius hybrid synergy drive system," *Oak Ridge Nat. Lab., US Dept. Energy*, 2011. (Cited on pages 29 and 36.)

[54] E. Rask, M. Duoba, H. Lohse-Busch, D. Bocci *et al.*, "Model year 2010 (gen 3) toyota prius level 1 testing report." Argonne National Lab.(ANL), Argonne, IL (United States), Tech. Rep., 2010. (Cited on p. 29.)

[55] X. Zhang, "Design of power split hybrid powertrains with multiple planetary gears and clutches." Ph.D. dissertation, 2015. (Cited on p. 35.)

[56] N. Kim, A. Rousseau, and E. Rask, "Autonomie model validation with test data for 2010 toyota prius," SAE Technical Paper, Tech. Rep., 2012. (Cited on pages 41 and 43.)

[57] ——, "Vehicle-level control analysis of 2010 toyota prius based on test data," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 11, pp. 1483–1494, 2012. (Cited on p. 41.)

[58] N. Kim and A. Rousseau, "Thermal impact on the control and the efficiency of the 2010 toyota prius hybrid electric vehicle," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, no. 1, pp. 82–92, 2016. (Cited on p. 46.)

[59] 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, 2017. (Cited on pages 51, 58, 64, and 73.)

[60] J. Hong, Z. Wang, W. Chen, and Y. Yao, "Synchronous multi-parameter prediction of battery systems on electric vehicles using long short-term memory networks," *Applied Energy*, vol. 254, p. 113648, 2019. (Cited on pages 51, 58, 64, 69, 71, and 73.)

[61] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee, 2013, pp. 6645–6649. (Cited on p. 51.)

[62] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014. (Cited on p. 51.)

[63] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 2017, pp. 1643–1647. (Cited on p. 51.)

[64] W. Li, A. Kiaghadi, and C. Dawson, "Exploring the best sequence lstm modeling architecture for flood prediction," *Neural Computing and Applications*, vol. 33, no. 11, pp. 5571–5580, 2021. (Cited on pages 51 and 64.)

[65] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318. (Cited on p. 52.)

[66] C. Vidal, P. Malysz, P. Kollmeyer, and A. Emadi, "Machine learning applied to electrified vehicle battery state of charge and state of health estimation: State-of-the-art," *IEEE Access*, vol. 8, pp. 52 796–52 814, 2020. (Cited on p. 52.)

[67] Y. L. Murphey, J. Park, Z. Chen, M. L. Kuang, M. A. Masrur, and A. M. Phillips, "Intelligent hybrid vehicle power control—part i: Machine learning of optimal vehicle power," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3519–3530, 2012. (Cited on pages 53, 63, 73, 79, and 81.)

[68] Y. Gurkaynak, A. Khaligh, and A. Emadi, "Neural adaptive control strategy for hybrid electric vehicles with parallel powertrain," in *2010 IEEE Vehicle Power and Propulsion Conference*. IEEE, 2010, pp. 1–6. (Cited on p. 53.)

[69] H. Kong, Y. Fang, L. Fan, H. Wang, X. Zhang, and J. Hu, "A novel torque distribution strategy based on deep recurrent neural network for parallel hybrid electric vehicle," *IEEE Access*, vol. 7, pp. 65 174–65 185, 2019. (Cited on pages 53 and 58.)

[70] "Worldwide Harmonised Light Vehicles Test Procedure," [Accessed: 2021-09-15]. [Online]. Available: https://en.wikipedia.org/wiki/Worldwide_Harmonised_Light_Vehicles_Test_Procedure (Cited on p. 55.)

[71] J. Zhu, Z. Yang, Y. Chang, Y. Guo, K. Zhu, and J. Zhang, "A novel lstm based deep learning approach for multi-time scale electric vehicles charging load prediction," in *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*. IEEE, 2019, pp. 3531–3536. (Cited on pages 56 and 64.)

[72] "Normalization Methods," [Accessed: 2021-09-21]. [Online]. Available: https://en.wikipedia.org/wiki/Normalization_(statistics) (Cited on p. 56.)

[73] "Standard Score Normalization," [Accessed: 2021-09-21]. [Online]. Available: https://en.wikipedia.org/wiki/Standard_score (Cited on p. 56.)

[74] Z. Zhang, H. He, J. Guo, and R. Han, "Velocity prediction and profile optimization based real-time energy management strategy for plug-in hybrid electric buses," *Applied Energy*, vol. 280, p. 116001, 2020. (Cited on p. 58.)

[75] "The Unreasonable Effectiveness of Recurrent Neural Networks," [Accessed: 2021-09-21]. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/ (Cited on pages xii and 59.)

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. (Cited on p. 59.)

[77] "Understanding LSTM Networks," [Accessed: 2021-09-15]. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Cited on pages xiii, 59, and 60.)

[78] L. Shi, M. Zheng, and F. Li, "The energy management strategy for parallel hybrid electric vehicles based on mnn," *Multimedia Tools and Applications*, vol. 79, no. 7, pp. 5321–5333, 2020. (Cited on p. 63.)

[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. (Cited on p. 69.)

[80] Z. Chang, Y. Zhang, and W. Chen, "Effective adam-optimized lstm neural network for electricity price forecasting," in *2018 IEEE 9th international conference*

on software engineering and service science (ICSESS). IEEE, 2018, pp. 245–248. (Cited on p. 69.)

[81] J. Park, Z. Chen, L. Kiliaris, M. L. Kuang, M. A. Masrur, A. M. Phillips, and Y. L. Murphey, "Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4741–4756, 2009. (Cited on pages 73 and 79.)

[82] M. Haußmann, D. Barroso, C. Vidal, L. Bruck, and A. Emadi, "A novel multi-mode adaptive energy consumption minimization strategy for p1-p2 hybrid electric vehicle architectures," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*. IEEE, 2019, pp. 1–6. (Cited on pages 73 and 79.)

[83] "Long Short-term Memory Networks," [Accessed: 2021-09-21]. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html (Cited on pages xiii and 74.)

[84] J. Wang, Q. Wang, X. Zeng, P. Wang, and J. Wang, "Driving cycle recognition neural network algorithm based on the sliding time window for hybrid electric vehicles," *International Journal of Automotive Technology*, vol. 16, no. 4, pp. 685–695, 2015. (Cited on p. 79.)

[85] 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, 2016. (Cited on p. 79.)

[86] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2016, pp. 324–328. (Cited on p. 79.)

[87] Y. L. Murphey, J. Park, L. Kiliaris, M. L. Kuang, M. A. Masrur, A. M. Phillips, and Q. Wang, "Intelligent hybrid vehicle power control—part ii: Online intelligent energy management," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 69–79, 2012. (Cited on p. 79.)

[88] Z. Chen, C. C. Mi, J. Xu, X. Gong, and C. You, "Energy management for a power-split plug-in hybrid electric vehicle based on dynamic programming and neural networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1567–1580, 2013. (Cited on p. 80.)