Model and Control System Development for a Plug-In Parallel Hybrid Electric Vehicle

Eduardo De Jesus Marquez Brunal

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Douglas J. Nelson, Chair
John B. Ferris
Steve Southward

(May 3, 2015)
(Blacksburg, VA)

Keywords: Hybrid electric vehicle, model based design, vehicle model, control strategy, performance, fuel economy, energy consumption, bench testing, P3 parallel.

Model and Control System Development for a Plug-In Parallel Hybrid Electric Vehicle

Eduardo De Jesus Marquez Brunal

Abstract

The Hybrid Electric Vehicle Team (HEVT) of Virginia Tech is participating in the EcoCAR 3 Advanced Vehicle Technology Competition series organized by Argonne National Labs (ANL), and sponsored by General Motors (GM) and the U.S. Department of Energy (DOE). EcoCAR 3 is a 4-year collegiate competition that challenges student with redesigning a 2016 Chevrolet Camaro into a hybrid. The five main goals of EcoCAR 3 are to reduce petroleum energy use (PEU) and green house gas (GHG) emissions while maintaining safety, consumer acceptability, and performance, with an increased focus on cost and innovation. HEVT selected a P3 Plug-in Parallel hybrid electric vehicle (PHEV) to meet design goals and competition requirements. This study presents different stages of the vehicle development process (VDP) followed to integrate the HEVT Camaro. This work documents the control system development process up to Year 2 of EcoCAR 3.

The modeling process to select a powertrain is the first stage in this research. Several viable powertrains and the respective vehicle technical specifications (VTS) are evaluated. The P3 parallel configuration with a V8 engine is chosen because it generated the set of VTS that best meet design goals and EcoCAR 3 requirements. The V8 engine also preserves the heritage of the Camaro, which is attractive to the established target market. In addition, E85 is chosen as the fuel for the powertrain because of the increased impact it has on GHG emissions compared to E10 and gasoline. The use of advanced methods and techniques like model based design (MBD), and rapid control prototyping (RCP) allow for faster development of engineering products in industry. Using advanced engineering techniques has a tremendous educational value, and these techniques can assist the development of a functional and safe hybrid control system. HEVT has developed models of the selected hybrid powertrain to test the control code developed in software. The strategy developed is a Fuzzy controller for torque management in charge depleting (CD) and charge sustaining (CS) modes. The developed strategy proves to be functional without having a negative impact of the energy consumption characteristics of the hybrid powertrain. Bench testing activities with the V8 engine, a low voltage (LV) motor, and high voltage (HV) battery facilitated learning about communication, safety, and functionality requirements for the three components. Finally, the process for parallel development of models and control code is presented as a way to implement more effective team dynamics.

Model and Control System Development for a Plug-In Parallel Hybrid Electric Vehicle

Eduardo De Jesus Marquez Brunal

General Audience Abstract

The Hybrid Electric Vehicle Team (HEVT) of Virginia Tech is participating in the EcoCAR 3 Advanced Vehicle Technology Competition series organized by Argonne National Labs (ANL), and sponsored by General Motors (GM) and the U.S. Department of Energy (DOE). EcoCAR 3 is a 4-year collegiate competition that challenges student with redesigning a 2016 Chevrolet Camaro into a hybrid. The five main goals of EcoCAR 3 are to reduce petroleum energy use (PEU) and green house gas (GHG) emissions while maintaining safety, consumer acceptability, and performance, with an increased focus on cost and innovation. HEVT selected a P3 Plug-in Parallel hybrid electric vehicle (PHEV) to meet design goals and competition requirements. This study presents different stages of the vehicle development process (VDP) followed to integrate the HEVT Camaro. This work documents the control system development process up to Year 2 of EcoCAR 3.

Acknowledgements

I would like to thank the Hybrid Electric Vehicle Team of Virginia Tech and all the team members that I had the honor to work with during Year 1 and Year 2 of EcoCAR 3. HEVT has given me the opportunity to further my technical, communication, and leadership skills while working with extremely smart, capable, and fun individuals. I would also like to thank General Motors, the U.S. Department of Energy, and Argonne National Lab for sponsoring and organizing EcoCAR 3. I want to thank all sponsors of EcoCAR 3, especially The MathWorks, where I will start my engineering career after completing my graduate studies. I would like to thank the creators of 5-Hour Energy drink for coming up with such a great product that enabled me to complete this work.

Next I would like to thank Dr. Doug Nelson for being my faculty advisor and for teaching me valuable life and technical lessons. Dr. Nelson's desire to share his automotive knowledge and experience, and his attention to detail are the elements that hold HEVT at such high excellence standards.

I would like to thank my family for their support and love through the good and hard times. Their desire to see me succeed keeps me motivated through every challenge, no matter how difficult a situation may be. Finally and most importantly, I would like to thank God for his unconditional love.

Table of Contents

List of Multimedia Objects

Tables

Figures

List of Abbreviations

ANL	Argonne National Labs
APP	Accelerator Pedal Position
BCM	Body Control Module
BEV	Battery Electric Vehicle
BMEP	Brake Mean Effective Pressure
BPP	Brake Pedal Position
CAN	Controller Area Network
CD	Charge Depleting
CS	Charge Sustaining
DLC	Data Link Connector
DOE	Department of Energy
ECM	Engine Control Module
EPS	Electronic Power Steering
ECU	Electronic Control Unit
ESS	Energy Storage System
EV	Electric Vehicle
FTP	Forward Thinking Patriot
GHG	Green House Gas
GM	General Motors
HEVT	Hybrid Electric Vehicle Team
HIL	Hardware-In-the-Loop
HVSC	Hybrid Vehicle Supervisory Controller
HWFET	Highway Fuel Economy Test
HV	High Voltage
HVBC	High Voltage Battery Charger
ICD	Interface Control Documentation
ICE	Internal Combustion Engine
IO	Input Output
LHV	Lower Heating Value
MBD	Model Based Design
MIL	Model-In-the-Loop
MPG	Miles Per Gallon
MPGGE	Miles Per Gallon Gasoline Equivalent
P2	Position 2
PEU	Petroleum Energy Use
PHEV	Plug-In Hybrid Electric Vehicle
PID	Proportional Integral Derivative
RCP	Rapid Control Prototyping
SIL	Software-In-the-Loop
SOC	State of Charge
TCM	Transmission Control Module
UDDS	Urban Dynamometer Drive Schedule
VDP	Vehicle Development Process
VTS	Vehicle Technical Specifications
WTW	Well to Wheel
WTP	Well to Pump

1. Introduction

1.1. Hybridization in the Automotive Industry

The economic and environmental implications of petroleum energy use and GHG emissions strongly support the need for continued development and advancement of hybrid electric vehicles (HEVs). GHGs are linked to global climate change, and criteria emissions are known to harm air quality, especially in urban areas. The use of fossil fuels such as coal and petroleum also contributes to these problems and are unsustainable in the long term.

GHG emissions resulting from the transportation sector in the U.S. account for 27% of the total GHG emissions [1]. U.S. transportation GHG emissions could increase by about 10% and oil could reach $200 per barrel by 2035 if current energy policies are not changed and no new renewable energy sources garner success [1]. Light-duty vehicles account for more than half of U.S. transportation energy use and CO_2 emissions [1]. Thus, improvements in well-to-wheel (WTW) PEU, WTW GHG and criteria emissions, and overall energy efficiency in the U.S. light duty vehicle segment will have a large impact on the U.S. in many ways. These improvements will reduce criteria emissions, improve air quality at the local and regional levels and reduce GHG emissions, which are a major contributor to global climate change.

In response to the negative effects of fossil fuel use, the U.S. market for fuel-efficient vehicles has gained momentum in recent years. The sales of hybrid electric vehicles increased 84.1% from 2011 to 2013, with almost 500,000 HEVs sold in 2013 [2]. While HEVs currently enjoy commercial success, there is a gap in the market segment for performance HEVs. Currently, consumers must face a trade-off between making an environmentally responsible purchase and enjoying a high performance vehicle. Additionally, the commercially available HEVs present a significant cost to the consumer. These factors serve as motivation to the automotive engineering community to minimize the environmental impact of transportation, reduce the cost of hybrid electric vehicles, and address the performance vs. energy consumption tradeoff in hybrids.

1.2. EcoCAR 3

HEVT is one of 16 universities participating in EcoCAR 3, a four-year competition sponsored by General Motors and the U.S. Department of Energy. The purpose of HEVT is to empower its students by challenging them to apply knowledge in unique ways that create cost effective and innovative solutions to sustainable transportation issues. Accordingly, the EcoCAR 3 competition challenges teams from North America to hybridize the powertrain of a 2016 Chevrolet Camaro with the main goals of reducing greenhouse gas emissions and petroleum energy use while maintaining safety, performance, and consumer acceptability. The participating teams are also tasked with considering competition-specific cost and innovative features when designing their final hybrid vehicle.

The design constraints for vehicle hybridization are driven by the EcoCAR 3 rules. Participating universities are responsible for generating team-specific goals that align with the competition requirements. HEVT started the powertrain component selection process with the identification of a consumer target market, followed by the definition of design goals and VTS. Figure 1-1 illustrates the implemented method that enabled HEVT to select a parallel PHEV as the team-specific powertrain configuration.

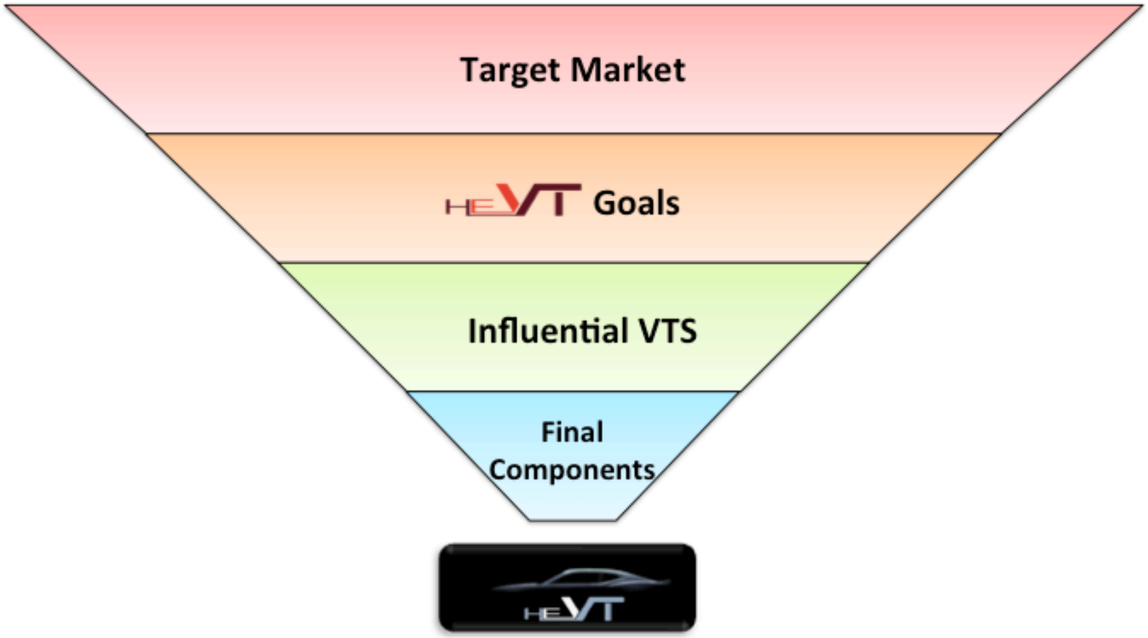


Figure 1-1: HEVT component down-selection process.

Definition of design goals and VTS is an intensive and iterative process that is covered in detail in following sections, and this study places more focus on VTS. The described engineering design process requires translating the voice of the target customer into engineering specifications; mathematical modeling of different hybrid components and powertrains is used to generate engineering specifications that can be evaluated to determine the success of the project. Furthermore, understanding and implementing advanced engineering techniques like MBD and RCP is critical for obtaining quantifiable results at every step of the down-selection process. Finally, VTS serve as metrics for success and they are used to evaluate obtained modeling results.

The hybrid powertrain chosen by HEVT to compete in EcoCAR 3 is shown in Figure 1-2. This PHEV is equipped with a 5.3 L V8 E85 GM engine, a 8 speed transmission, a 100 kW post-transmission (P3) custom motor designed with InMotion Systems, and a 12.6 kWh, 118 kW energy storage system (ESS) from A123 Systems.

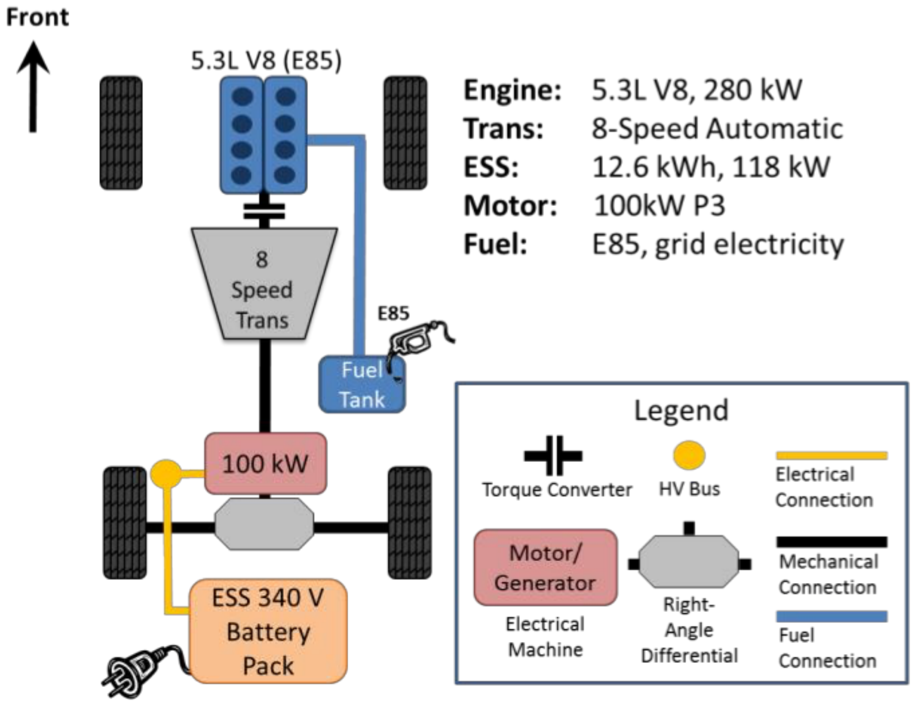


Figure 1-2: Powertrain configuration of PHEV studied in this thesis.

1.3. Vehicle Development Process

Hybridizing a modern vehicle is a complex procedure that requires understanding of the automotive system. Hybridization also requires elaborate design, planning, and execution of ideas developed through an engineered process. The VDP implemented by HEVT is largely based on the Global Vehicle Development Process that GM uses to develop new vehicle platforms. Figure 1-3 illustrates the different engineering stages over the four years of the EcoCAR 3 competition.

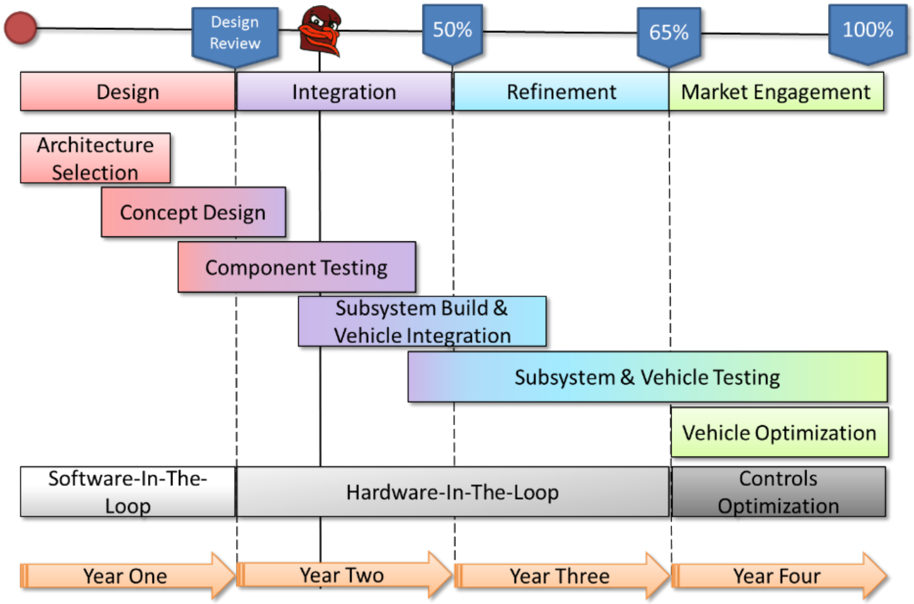


Figure 1-3: Vehicle Development Process implemented by HEVT [21].

The Hokie indicator shows the current progress at the time of this writing. Highlighting the current progress in the VDP is important to define scope of this study. Goals of the VDP include conducting a design review by the end of the first year and having a mule vehicle that is 50% production-ready by the end of the second year. Figure 1-3 also shows integration activities and control system development tasks that must be conducted during the four years of EcoCAR 3.

Advantages of using a process with the structure shown are reflected in the development cost and time for a new and final vehicle platform. Implementing the VDP reduces variation and rework in execution of the process. In addition, results of a well-defined and executed process are predictable, repeatable, and measurable [3]. When compared to older and more traditional approaches, the VDP reduces the necessary resources and time to generate a final product because extensive planning, design, modeling, and simulations are conducted before creating prototypes. The VDP encompasses the use of advanced tools and techniques like MBD, physical modeling, the V-Cycle in RCP, and iterative in-the-loop testing.

1.4. Model Based Design

Modern engineering standards demand less time and less use of resource to create new products and services [4]. In the automotive industry, conventional and hybrid vehicle systems have become highly complex, and it is not out of the ordinary for a modern car to have more lines of code than a commercial airplane [5]. The high complexity of vehicles requires design methods that enable rapid analysis and testing of different powertrains. In Model Based Design, mathematical models can simulate the operation of vehicle components. Furthermore, understanding the physical, mathematical, and engineering principles of vehicle components allows for preliminary result generation in a virtual environment. Software simulations for several powertrains configurations can be run in a more timely and cost effective manner than real tests with prototypes. Thus, performance characteristics and environmental impact can be quickly analyzed to determine if a powertrain configuration meets the goals of the designer or not. If the goals are not met, then the design becomes an iterative process.

In this study, different powertrains are modeled by varying the size and location of components in the vehicle. The list of analyzed powertrain components includes: internal combustion engines (ICE), automatic transmissions, electric motors, and HV battery packs. The P-notation is used to denote the position of electric machines, where a P1 motor is located on the accessory side of the engine, a P2 motor is between the engine and transmission, a P3 motor is post-transmission, and a P4 motor is located on the axle non driven by the engine. Table 1-1 details the studied powertrain components and the changed parameters for each component.

Table 1-1: Components evaluated with MBD.

Component	Varied Parameter
Engine	Displacement in liters
Transmission	Number of gears
Motor	Torque & speed characteristics
Differential	Final drive gear ratio
High Voltage Battery Pack	Energy capacity & power output

The software tools used to initially simulate the different powertrains are Autonomie [22] and MATLAB/Simulink. Autonomie is developed by Argonne National Laboratory and is widely used in the automotive industry due to its powerful capabilities and user-friendly interface. In this study, Autonomie serves as the initial and fastest method of generating and testing different powertrain configurations. After down-selecting a specific powertrain, MATLAB/Simulink is used to gain more freedom in the modeling and testing stages. The powertrain selection section provides extensive detail on the used tools for modeling and testing platforms.

1.5. Rapid Control Prototyping

Rapid control prototyping is a process that allows engineers to create, test, and calibrate control algorithms on target hardware to accelerate the development time of an engineering product. Historical design techniques attempt to first develop prototypes that can be tested, perform the tests, and then make decisions based the observed, measured, and calculated results. Prototypes are expensive in terms of time and resources. New design theories in the automotive industry are moving towards a new way of development that create mathematical models to generate preliminary results [6]. Models are created before expensive prototypes are. These models can be tested much faster than a real prototype. Thus, with new design techniques the product time to market is drastically reduced. Rapid control prototyping provides a method to deploy mathematical models and control algorithms to target hardware and run them on real-time tests with a detailed input/output (I/O) interface [6]. In-the-Loop testing platforms are implemented with rapid control prototyping. Model-In-the-Loop (MIL), Software-In-the-Loop (SIL), and Hardware-In-the-Loop (HIL) are commonly used test settings that enable model testing at different levels of detail and fidelity. In-the-Loop testing is explained in more detail in Section 3 and Section 4 of this research. Special emphasis is placed on MIL and SIL as they are extensively used in this study.

1.6. Objectives

This research study aims to present and document different stages of the control system development process for a parallel plug-in hybrid electric vehicle. In addition, this study aims to detail the process for selecting a hybrid powertrain that meets specific design goals. This work also seeks to show the theory, methods, and techniques used to develop a hybrid vehicle model that represents the expected performance of the modified Camaro. The value of developing a vehicle model is the ability to test the algorithm needed to control the hybrid vehicle. Thus, another objective of this study is to detail the development of the control system, including control requirements, control code, and control architecture for controller area network (CAN) communication. Furthermore, conducting and documenting bench testing of components is another goal of this work. Component testing is a valuable activity that provides information about communication, safety, and functionality requirements for operation. Finally, this study aims to present the processes and tools used to develop control code logic and models in parallel (with several team members); presenting the method for seamless migration between testing platforms while keeping all models up to date is also an objective of this work.

2. Literature Review

2.1. Control Strategies for Hybrid Electric Vehicles

The article written by Salmasi [18] presents an overview of state-of-the-art control strategies for hybrid electric vehicles. The paper highlights the advantages and disadvantages of different approaches. This article provides a strong foundation for learning about the different control methods being currently implemented, and helps the reader understand the work that has been done in the area. The challenges of designing energy management systems are discussed, and the main objectives of control strategies are presented as: meeting driver demand, sustaining battery state of charge (SOC), and optimizing the drivetrain efficiency, fuel consumption, and emissions. According to the author, the most common hybrid powertrain configurations are general series, parallel, and power split. Subsequently, the classification of hybrid powertrain control systems is presented and discussed. The main classification is between rule-based and optimization-based strategies. Rule-based control systems are divided into two categories: Fuzzy rule-based and deterministic rule-based. Similarly, optimization-based systems are divided into global and real-time optimization.

Deterministic rule-based methods like thermostat, power follower, modified power follower, state machine-based are explained. Thermostat strategies are presented as simple and primitive for the needs of a hybrid vehicle system, as only battery state of charge is monitored to be maintained within the specified window. A simple set of rules are shown for a power follower strategy, where thresholds, lower and upper limits can be specified for vehicle speed, power demand and state of charge. A more complex adaptive rule-based strategy is also shown in the paper; the adaptive rule-based strategy seeks to optimize the energy use and emissions of the hybrid powertrain by using a cost function that represents overall fuel consumption and emissions at possible operating points. Finally, state machine-based methods are discussed. Different states can be created to make the powertrain behave in a desired manner, and the transitions between states are defined by simple sets of rules.

Conventional, adaptive, and predictive strategies are presented and discussed as Fuzzy Rule-based methods. Fuzzy logic controllers are presented as an extension of conventional rule based-methods. The main advantages of Fuzzy methods are robustness due to the tolerance to imprecise inputs, and adaptation due to the ability to tune Fuzzy rule sets. The discussion on Fuzzy strategies from the paper by Salmasi is useful to develop the conventional Fuzzy strategy used in this study in Section 5.

The limitation of global optimization control strategies is the inability to be used directly for real-time implementation. However, global optimization methods are useful for design and comparison purpose as they can find the optimal torques (for powertrain components) and gear ratios over fixed drive cycles. On the other hand, methods that use an instantaneous cost function can optimize powertrain operation in real-time applications. Optimization strategies that use instantaneous cost functions should include an equivalent fuel consumption model. Real-time optimization strategies can also account for gear shifting response and drivability.

The information from this paper is valuable because it discusses the work that has been done and the techniques that have been applied so far in the field of control systems for hybrid electric vehicles. Although optimization methods are highly complex and not used in this study, the information presented is insightful and motivates the reader to consider advanced control system methods. Finally, the paper by Salmasi helps establish goals for this study by highlighting the challenges faced and the problems that must be solved when designing a hybrid vehicle control system.

2.2. Design of Powertrains using Model-Based Design

Ord [15] presents a systematic approach for modeling several hybrid electric powertrains using model-based design. The purpose of modeling several powertrains is to generate sufficient data to guide the selection of a hybrid powertrain to meet very specific targets of performance, emissions, and energy consumption. Series and parallel powertrains are analyzed to find the right combination of component sizes and characteristics that can meet the design criteria. This aspect of Ord's study relates heavily to the powertrain selection process (Section 3) conducted in this research. In addition, Ord's work provides necessary background to understand how the energy consumption and emissions criteria of a hybrid electric vehicle are evaluated. Concepts like cycle weighting and utility factor are explained in detail. The goal is to help the reader understand the impact that using a battery (displacing fuel use with grid electric energy) can have on total energy consumption for a hybrid electric vehicle. Applying these concepts is relevant to the present research because a powertrain also has to be selected (Section 3) based on modeling results to best meet a set of design targets and vehicle technical specifications.

Ord's thesis covers several relevant concepts for model-based design. Some of these concepts are power and torque loss modeling, and efficiency considerations. The component modeling techniques and theory shown by Ord are of high value to this research because they provide the necessary information to represent components and hybrid vehicle operation. The representation of vehicle dynamics is explained in such a way that implementation in Simulink is easy. Although simple, the glider model shown is a powerful and robust subsystem that is highly compatible with other powertrain component plants. In addition, the engine and motor models described help in understanding scaling techniques that can be applied to represent operation of similar motors and engines with different sizes (scaling by motor power output and by engine displacement). The concept of power loss modeling serves as a tool in this research to develop additional plants models for auxiliary components, like a DC/DC converter and a high voltage battery charger (HVBC).

Finally, several control strategies, like thermostatic and power following, are discussed for different powertrain configurations. The discussion on hybrid vehicle supervisory controller (HVSC) development highlights the impact that component sizing can have on the control strategy of a specific powertrain. Although the present research is focused on the development and use of a conventional Fuzzy logic, the rule-based strategies for parallel powertrains discussed by Ord provide valuable guidelines for structure and rule development of a hybrid vehicle control system.

2.3. Development of Energy Management Strategies for Hybrid Electric Vehicles

Manning [19] focuses on implementation of industry methods like the GM vehicle development process and rapid control prototyping to produce a hybrid control system for a series-parallel powertrain. The model results from different hybrid powertrains are discussed and shown to drive the selection of the series-parallel configuration that best meets design goals. The control system development conducted by Manning for a series-parallel powertrain is a sound process that starts by defining control requirements for the hybrid vehicle components. Then, requirements are defined for the simulation platforms and plant models; doing so, generates information about the expected results and knowledge that should be obtained from testing in different platforms like Software-In-the-Loop and Hardware-In-the-Loop.

The control algorithm discussed by Manning is the main part of the control system in his work. The algorithm developed is responsible for monitoring, managing, and commanding all vehicle components with a passive control strategy. The code structure proposed by Manning is comprised by a vehicle input/output interface, a diagnostics subsystem, a selection subsystem, and an execution subsystem. The proposed structure identifies faulted components, and can take remedial action by implementing a fault mitigation strategy to continue safe vehicle operation. The functionality of the diagnostic subsystem is to monitor the status of all components during operation and report faults. The diagnostic system requirements are developed from DFMEA so that the most undesired faults can be identified and mitigated. Consequently, the selection subsystem proposed by Manning is influenced by the diagnostics portion. The selection subsystem decides the hybrid mode of operation based on driver demand and status of components (SOC, vehicle speed, etc). Finally, the execution portion of the code communicates the commands from the hybrid vehicle supervisory controller to the component controllers. The algorithm structure from Manning's work provides a logical sequence of operations to effectively control a hybrid electric powertrain. The same structure is used in the control code developed in this research. Although the outer structure of the code developed in this research and Manning's control algorithm is the same, the code subsystems, details, and operations are not equal. The vast internal differences occur because each powertrain configuration requires a very specific and unique control algorithm.

Manning's work highlights that bench testing of powertrain components is an important activity that provides valuable information about performance characteristics and control interfaces. The operation of a specific component can be evaluated in a controlled environment outside of the vehicle. Additionally, information can be gathered to develop code to better interface with that component in the vehicle. Manning characterized the thermal and performance behavior of a P2 motor system through dynamometer testing, and also established communication with an engine on a stand. Testing results can and should be used to drive model development and improvement and control algorithm functionality. These activities motivate the present research to conduct and document as many bench testing results as possible from available components. Section 6 in this research details testing of an engine, motor, and battery systems separately.

The overall process document by Manning has a very similar scope compared to the process documented in this research. Both processes highlight the importance of a safe and functional supervisory system that can effectively control a hybrid electric vehicle.

2.4. Energy Management Strategy Based on Fuzzy Logic

Gao [20] develops an energy management strategy for a fuel cell hybrid bus with two energy storage devices, a high voltage battery and an ultracapacitor. The energy management strategy uses Fuzzy logic to control the energy flow in the powertrain. The proposed strategy determines the power output from the fuel cell system, high voltage battery, and ultracapacitor based on driver demand and regen braking energy available. A clear explanation of the basic concepts of a Fuzzy inference system is provided for background. The Fuzzy logic controller relates the outputs to the inputs using a list of IF-THEN rules. A Fuzzy rule is, as defined by Gao, an IF-THEN statement in which some words are characterized by membership functions. The IF and THEN portions of the statement are called the antecedent and the consequent, respectively. Gao argues that the Fuzzy logic controllers are very suitable for processes with complex models. In addition, he mentions that the specific rule set for the Fuzzy logic controller depends largely on the designer's knowledge about the power supplies and traction device limits. The developed Fuzzy logic controller has three input variables (driver power demand, battery SOC, and ultracapacitor SOC) and two output variables (power output from the fuel cell system, and power output from the ultracapacitor). The battery power output is determined from the difference between the total power demand and the power provided by the fuel cells and the ultracapacitor. Gao explains that the performance of the energy management strategy is determined by the Fuzzy rules, and the number and shapes of the membership functions for each variable of the Fuzzy inference system. Descriptive figures showing the membership functions make this paper very easy to read and understand. Gao's work is of tremendous importance for this research because it drives the motivation to implement a Fuzzy control strategy for the CD and CS operation torque regulations. This study uses Fuzzy control logic because it provides a learning opportunity for design while staying away from the high complexity of an optimization strategy. Gao's membership functions serve as guidelines for the Fuzzy logic inference system used in this thesis.

Finally, testing is performed to validate the behavior of the Fuzzy energy management strategy developed. The Fuzzy control strategy is tested on an actual hybrid bus in the streets of Beijing for 1000 seconds. The results shown prove that the output of the Fuzzy inference system are as expected. However, there is no additional data from a conventional or a different hybrid electric bus for comparison.

2.5. Structuring Models for Version Control and Work with Multiple Developers

Crain [17] details the development process and structure of the hybrid vehicle supervisory controller used by the University of Washington during EcoCAR 2. The developed supervisory system is created in Stateflow and Simulink, and is designed to control a parallel through the road powertrain. Crain argues that the graphical interface of the coding tools used offers an advantage over text-based code. However, using version control tools in complex models where several people are doing development can be difficult. Partitioning the hybrid vehicle supervisory controller in different subsystems of components allows using linked library blocks that facilitate code development in parallel. The methods and tools proposed by Crain are used in this research to enable parallel code development for the hybrid supervisory controller and hybrid vehicle model. In addition, these methods facilitate and complement the process established in this research to seamlessly migrate between Software-In-the-Loop and Hardware-In-the-Loop testing platforms.

2.6. Summary of Literature Review

The literature review stage in this study serves to explore the significant work and accomplishments in the field of hybrid electric vehicle controls. The reviewed papers cover a wide range of systems, methods, topics, and techniques that are relevant to specific sections of this research. The Salmasi paper presents an overview of state-of-the-art control strategies for hybrid electric vehicles, and provides a strong foundation to understand the characteristics, advantages, and limitation of common hybrid vehicle strategies. The study by Ord presents a systematic approach for modeling several hybrid electric powertrains and developing different control strategies using model-based design. The theory, concept, and models presented by Ord guide the modeling efforts in this research. Similarly, the work done by Manning strongly influences the control system development process implemented in this thesis. The control algorithm structure and the importance of bench testing activities are aspects of this research that build on Manning's work. Gao's article provides a simplified, yet informative explanation and example of using a Fuzzy control strategy to determine a power split in a hybrid vehicle. Gao's work truly serves as motivation to look deeper into the technicalities and possibilities for hybrid vehicle control with Fuzzy logic. Finally, Crain's paper proposes a modeling structure and procedure to enable parallel code development while minimizing version control conflicts. The methods and tools proposed by Crain are used to also complement the established procedure for migration between Software-In-the-Loop and Hardware-In-the-Loop testing platforms.

3. Hybrid Powertrain Selection

3.1. Vehicle Technical Specifications

Creating design goals and VTS is the translation from the voice of the target customer into engineering metrics that can be measured to determine success in the VDP. Thus, the main goal of this subsection is to clearly present the final set of VTS for the parallel PHEV shown in **Figure 1-1.** Table 3-1 shows the generated specifications for the powertrain configuration chosen by HEVT. Extensive detail on setting VTS is provided in [21]

Table 3-1: HEVT vehicle technical specifications.

Vehicle Technical Specification	Comp. Req.	HEVT Target	HEVT Model P3 V8
Acceleration, IVM-60 mph, sec	7.9	5.0	4.9
Acceleration, 50-70 mph (Passing), sec	9.9	5.0	3.5
Braking, 60-0 mph, ft	135	128	< 128
Acceleration Events Torque Split (Front/Rear)	49% F	0% F	0% F
	51% R	100% R	100% R
Lateral Acceleration, 300 ft. Skid Pad, G	0.8	0.85	0.82
Double Lane Change	52	58	53
Highway Gradeability, for 20 min	6%	6%	6%
	@ 60 mph	@ 60 mph	@ 60 mph
Cargo Capacity, ft^3	2.4	> 2.4	4
Passenger Capacity	2	4	4
Curb Mass, lb (kg)	4273 (1938)	≤ Comp. Req	4277 (1939)
Starting Time, sec	15	2	3 (est)
Total Vehicle Range*, mi (km)	150 (241)	187 (300)	170 (274)
CD Mode Range*, mi (km)	N/A	22.5 (36.2)	25.5 (41.1)
Pure EV			
CD Mode Total Energy Consumption* AC Wh/km (mpgge)	N/A	280 (75)	275 (76)
CS Mode Fuel Consumption*, Wh/km (mpgge)	N/A	840 (25)	810 (26)
UF-Weighted Fuel Energy Consumption*, Wh/km (mpgge)	N/A	480 (44)	430 (49)
UF-Weighted AC Electric Energy Consumption*, AC Wh/km	N/A	120	130
UF-Weighed Total Energy Consumption*, Wh/km (mpgge)	840 (25)	600 (35)	560 (37)
UF-Weighted WTW PEU*, Wh PE/km	750	150	125
UF-Weighted WTW GHG Emissions*, g GHG/km	250	175	170

Vehicle Technical Specification	Comp. Req.	HEVT Target	HEVT Model P3 V8
UF-Weighted WTP Criteria Emissions* g/km (score)	TBD	≤ Comp. Req	0.0418 (24.9)

The project management subteam of HEVT conducted a consumer market research report. The consumer market study concluded that the forward-thinking patriots (FTP) are the ideal target market segment. The term FTP labels consumers in the Roanoke Valley with the following characteristics: retirement-age, politically conservative, loyal to American products, caring about energy independence, seeking excitement, and intelligent savers and spenders. According to the data gathered in the regional profile of the consumer market research report, this market segment is more capable, and more likely, than any other market segment in the Roanoke Valley to purchase a performance hybrid Camaro. From a pure consumer perspective, the top goals should be: exciting performance and minimum energy consumption for energy independence and frugality [7].

A strong understanding of the target market provides a solid foundation for the generation of engineering metrics that quantify the success of the VDP. The emphasis of this thesis is placed on VTS because these vastly overlap with design goals. The goals generated in Year 1 of EcoCAR 3 are: meet all minimum competition requirements, avoid designs that incur known penalties, meet 2020 CAFE standards for Camaro class vehicle, maintain Camaro stock seating for four persons, maintain 100% torque on rear axle, and match performance of V8 model while improving fuel economy to surpass that of V6 model. The engineering design of a hybrid electric vehicle requires addressing tradeoffs at different stages. Some tradeoffs encountered in this study include (not limited to): IVM-60 mph performance versus fuel consumption, vehicle weight versus electric range, and vehicle handling versus weight distribution.

Powertrain-specific VTS are created for all extensively analyzed configurations after studying the target market and generating design goals. These sets of VTS are the result of iterative modeling and simulations covered in detail in the following subsections. Initially, five powertrain configurations are proposed as viable to meet design goals, and only advantages and disadvantages of each powertrain are exposed. Subsequently, four different powertrain configurations are extensively analyzed and compared to finally select the hybrid architecture that best meet design goals and satisfies the target market. The need to propose several viable powertrains arises from the constraints placed by the EcoCAR 3 competition. These constraints are driven by the balance that competition organizers and teams have to strike between obtaining sponsored components and paying for powertrain components. In addition to cost and component sourcing constraints, teams must evaluate the feasibility of all proposed powertrains – this evaluation drives the analysis of mechanical integration aspects like torque couplings, component packaging and mounting, cooling, weight limits, and more.

3.2. Feasibility Study

Conducting a feasibility study helps to better understand key aspects and constraints involved in the selection of a hybrid powertrain configuration for the design process. First, emissions, energy consumption, and performance characteristics of the stock 2016 Camaro are evaluated through modeling in Autonomie. The baseline vehicle characterization contains a description and an analysis of the assumptions/limitations of the model. In addition, the PTW energy consumption results are analyzed, and the vehicle acceleration performance is simulated. Subsequently, several fuel types are investigated to analyze their impact on emissions and energy consumption.

3.2.1. Autonomie Modeling - Assumptions and Limitations

The Autonomie model of the stock 2016 Camaro is provided by ANL in collaboration with GM. The given model consists of three main systems that affect the vehicle response: the driver, the environment, and the vehicle propulsion architecture. The vehicle propulsion architecture is composed of highly detailed models for twelve major architecture components shown in Figure 3-1. Component models are related to each other based on input and output signals of energy flow and logic. For instance, the alternator subsystem accepts a torque input from the torque coupling and outputs electrical power to either the 12 Volt battery system or electrical accessories or a combination of both. This rational linkage of models accounts for all component interactions from the throttle position to the torque output at the wheels. Each component model is organized hierarchically by complexity of sub-models to capture both the function and control strategy of the component.

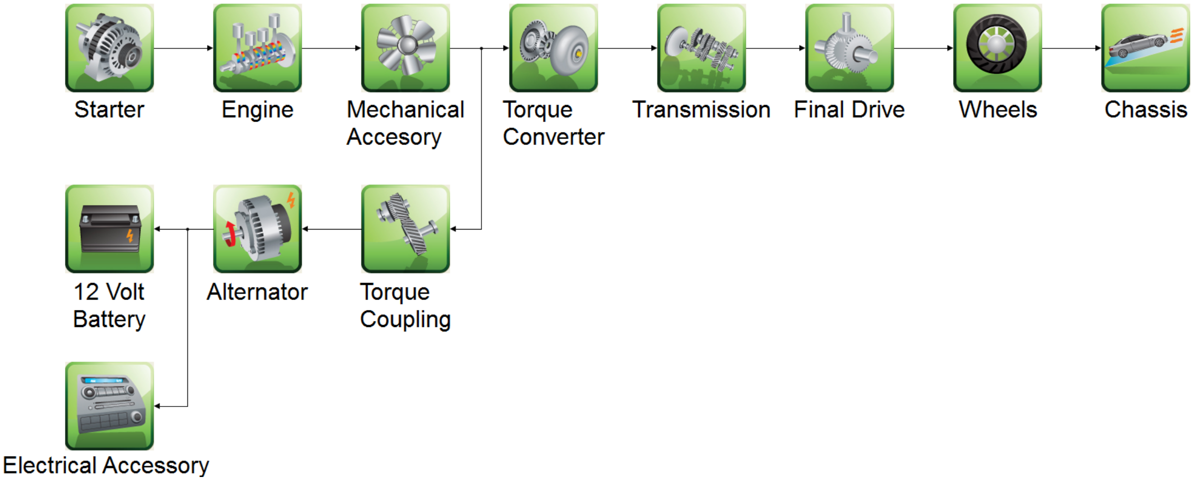


Figure 3-1: Autonomie model structure for stock 2016 Camaro.

Autonomie is a powerful modeling tool that accounts for many vehicle parameters to produce accurate results. However, careful analysis of simulations is necessary to ensure realistic results. There are certain parameters that can have relevant impact on energy consumption and vehicle performance that are not modeled by the software tool. The following list outlines limitations and sensitivities of the studied stock model:

- Engine hot fuel map only accounts for hot start, so the engine is assumed to be at steady state thermal conditions immediately upon initial operation.
- Engine fuel consumption is a static fueling rate table at discrete torque and speed points, and does not include transient or environment effects.
- Engine torque is limited to prevent wheel slip, but the engine controller model can be modified to overcome the torque limitation.

- Engine overheating is not modeled, thus the engine controller allows the plant to operate at very high RPM indefinitely.
- Air conditioning (A/C) is assumed not to be in use (zero accessory load). A/C use has have a considerable effect on energy consumption.
- All operators are assumed to have the same driving habits.
- Weather conditions like snow and rain are neglected as they are difficult to model accurately. However, ambient pressure and temperature are taken into account.

3.2.2. PTW Energy Consumption Analysis for Stock Vehicle Model

The PTW energy consumption for the stock Camaro vehicle model is derived from the EcoCAR 4-cycle drive schedule. This drive schedule combines the measured energy consumption from drive cycles shown in Table 3-2 and weights them with specific percentages. The EcoCAR 4-cycle is meant to simulate the Environmental Protection Agency (EPA) five-cycle without weather testing or A/C use. The provided Autonomie model is simulated for each of the four drive cycles, and the fuel economy and PTW energy consumption values are displayed in Table 3-2.

Table 3-2: Weighted mpg and fuel consumption for EcoCAR 3 drive cycle

Cycle	Weight (%)	Fuel Economy (mpg)	Energy Consumption (Wh/km)
505	29	29.2	715
HWFET	12	40.9	510
US06 City	14	15.9	1314
US06 Highway	45	30.6	682
	Weighted Total	27.5	763

The stock vehicle is modeled with a mass of 1873 kg. This test mass is determined by using a curb mass of 1691 kg (found in a different stock vehicle model provided by MathWorks and GM) and adding two 91 kg passengers. Energy consumption is calculated using 33.7 kWh/gallon as the energy density of gasoline. Instead of explicitly stating the type of fuel used during simulation, Autonomie lists the lower heating value (LHV) and carbon content of the fuel. The stock model lists a lower heating value of 42.8 MJ/kg and a carbon content of 0.87 kg/kg, which corresponds most closely with characteristics of gasoline (LHV of 42.9 MJ/kg and carbon content of 0.842 kg/kg) [8]. The weighted PTW energy consumption and fuel economy for the 4-cycle drive schedule are 763 Wh/km and 27.5 mpg, respectively. The fuel economy modeled for each cycle is show in Table 3-3.

Table 3-3: Drive cycle fuel economy results from Autonomie.

Drive Cycle	Fuel Economy (mpg)
505	29.17
HWFET	40.90
US06 City	15.87
US06 Highway	30.58

Subsequently, the energy equivalent of gasoline (33,700 Wh/gallon of gasoline) is divided by each fuel economy to obtain energy consumption in Wh/mi, as show in Table 3-4.

Table 3-4: Drive cycle energy consumption in Wh/mi.

Drive Cycle	Energy consumption (Wh/mi)
505	1155.3
HWFET	823.96
US06 City	2123.5
US06 Highway	1102.0

Finally, Wh/mi is converted to Wh/km by dividing the values in Table 3-4 by 1.60934. Table 3-5 shows the energy consumption results for each cycle in Wh/km.

Table 3-5: Drive cycle energy consumption in Wh/km.

Drive Cycle	Energy consumption (Wh/km)
505	717.86
HWFET	511.98
US06 City	1319.5
US06 Highway	684.77

Cycle energy consumption values are multiplied by their respective weight factor (0.29, 0.12, 0.14, and 0.45) and summed together to get the weighted PTW energy consumption in Wh/km and fuel economy in mpg. The EPA sticker adjusted fuel economy for a 2015 Camaro (22 mpg) with a 6.3 Liter V8 engine and 6-speed transmission is compared to the obtained results for verification purposes [9]. The 5.5 mpg difference between the 2015 Camaro (2016 Camaro was not commercially available at time of modeling) and the Autonomie model may be due to the absence of cold weather cycles and A/C cycles in the EcoCAR 4-cycle. Cold weather cycles are present in the EPA 5-cycle and can increase the energy consumption of the vehicle. Additionally, cold starts are present in the EPA 5-cycle, which are not accounted for in the Autonomie model.

3.2.3. <u>Component Efficiency & Energy Losses in Stock Model</u>

The efficiency and energy losses of each component are analyzed for each drive cycle from the EcoCAR 4-cycle simulations. The efficiency and energy loss of selected components are shown in Table 3-6**.** Figure 3-2 displays this information graphically.

Table 3-6: Efficiencies and losses for various components for each cycle in the EcoCAR 4-cycle.

Component		505		HWFET		US06 City		US06 Highway	
		Efficiency (%)	Losses (Wh)	Efficiency (%)	Losses (Wh)	Efficiency (%)	Losses (Wh)	Efficiency (%)	Losses (Wh)
Engine		26.4	3085.7	29.1	5998.3	31.3	2625.5	31.4	4719.7
Gearbox		94.0	62.0	93.1	166.2	93.4	70.3	92.9	153.6
Vehicle Glider	Chassis	80.1	240.8	47.5	1073.9	89.5	158.0	50.2	1069.7
	Wheel	59.9	520.9	62.5	908.4	60.2	597.0	70.9	645.8

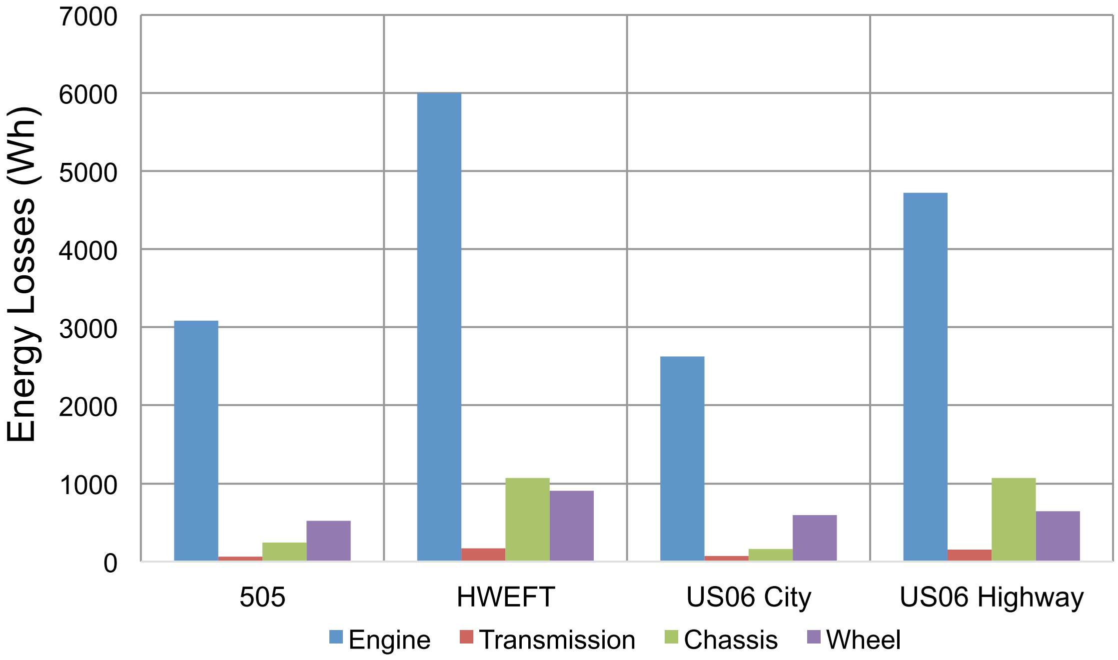


Figure 3-2: Energy losses for different components for each cycle of the EcoCAR 4-cycle.

Across all drive cycles, the engine is the least efficient component. Engine efficiency is dictated by combustion, which is naturally an inefficient process. The engine efficiency ranges from 26.4% for the 505 drive cycle to 31.4% for the US06 Highway drive cycle. Typical engine efficiency ranges from 14-30% for combined city and highway driving [10]. The mechanical accessory loads are not accounted for in Autonomie simulations. These accessories, particularly A/C, would create losses in the vehicle. The transmission has an efficiency of approximately 93% for all cycles. This is expected, as gearboxes are designed to have low losses and vehicle control codes typically initiate efficient shifting. The vehicle glider, which includes both the tires and chassis of the vehicle, accounts for the second highest energy loss values. The highway drive cycles have significantly lower chassis efficiencies than the city drive cycles due to the increased aerodynamic drag at higher speeds.

Engine torque-speed hot maps are generated to examine engine operation during each cycle of the EcoCAR schedule. Figure 3-3 to Figure 3-6 show the engine-operating characteristic for each drive cycle. For the highway drive cycles, the engine consistently operates above 30% efficiency. The high efficiency of the US06 City drive cycle is also reflected in the engine hot map, with many points above the 30% efficiency line. This drive cycle also shows a cluster of negative torque operating points, which represents increased braking incidents during a city drive cycle. The low engine efficiency of the 505 Cycle is demonstrated by the many instances between 15 and 25% efficiency, with a large number of negative torque points due to braking. The value of this kind of hot maps is that they can be used to determine the most efficient points and to develop control strategies that operate the engine in more efficient zones of the hot map.

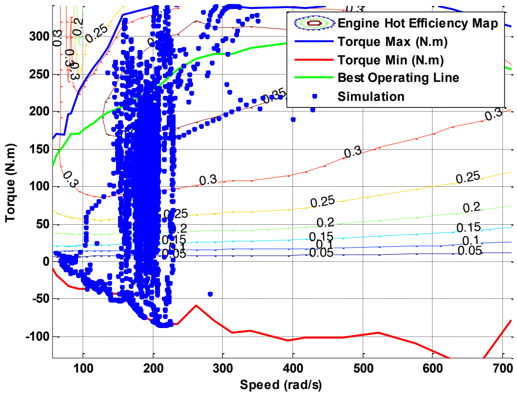


Figure 3-3: US06 Highway engine hot map.

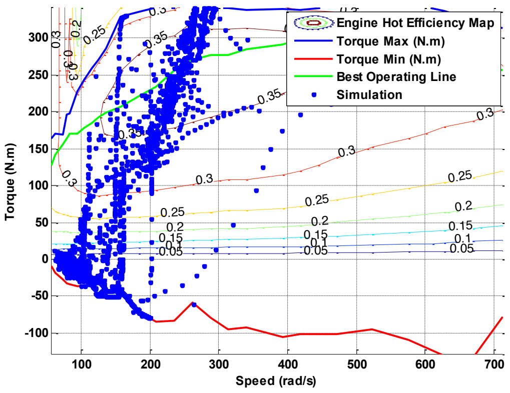


Figure 3-4: US06 City engine hot map.

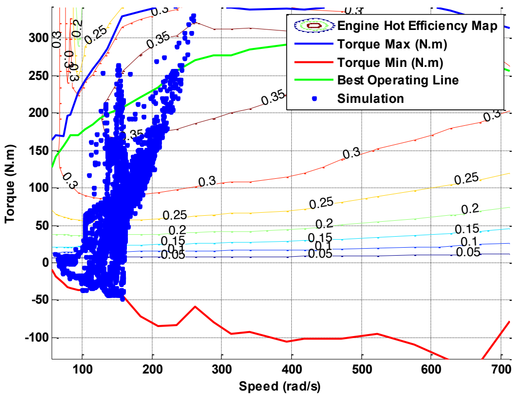


Figure 3-5: 505 Cycle engine hot map.

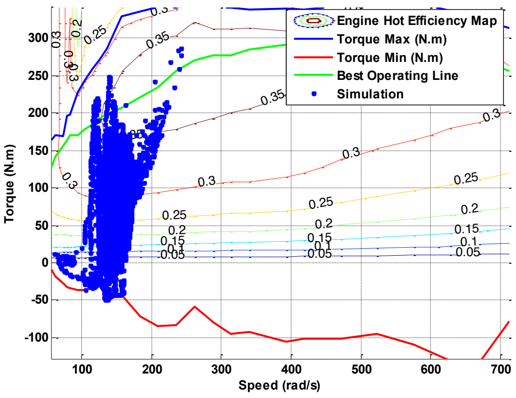


Figure 3-6: HWFET Cycle engine hot map.

3.2.4. Performance Characteristics of Stock Vehicle Model

The stock vehicle model performance is analyzed and several modifications are implemented to evaluate their impact on the system response. Modifications in the model include: engine preloading, changing transmission shift points, and wheel torque for traction limits in the Camaro Autonomie model.

Engine preloading increases the engine speed before the acceleration event. A preloaded engine will reach higher RPM and peak torque faster, improving acceleration performance. Upscaling the transmission shift map also provides faster acceleration by raising the engine speed at which the transmission upshifts. By reaching higher engine speeds, the engine achieves greater power leading to faster acceleration. The engine controller model also incorporates a conservative torque limitation based on wheel slip; this torque limitation was reduced to ensure the torque capability of the engine is reflected in the acceleration performance. Table 3-7 displays the effects of model changes on acceleration performance. The results from Table 3-7 indicate that while engine preloading and shift map scaling have a slight effect, the torque limitation has the largest effect on acceleration performance.

Table 3-7: Acceleration Performance Characteristics for Model Changes

A - Engine Preload B - Upscale Shift Map C - Remove Torque Limitation						
Run @ 1873 kg	**1**	**2**	**3**	**4**	**5**	**6**
Changes Applied	-	A	B	C	A, B	A, B, C
Δt 0 to 60 mph (s)	7.6	7.4	7.3	6.9	7.2	6.3
Δt IVM to 60 mph (s)	7.0	6.9	6.7	6.3	6.7	5.8
Δt 50 to 70 mph (s)	6.9	6.0	6.5	6.9	5.7	5.1

3.2.5. Study of Fuel Types

Three fuel blends are studied for potential use in the Camaro. The first two fuels, E10 and E85, are gasoline and ethanol mixtures that are 10% and 85% ethanol by volume, respectively. The third potential fuel, B20 biodiesel blend, is a mixture of 20% biodiesel and 80% petroleum diesel. These three fuels are evaluated for GHG emissions, PEU, and criteria emissions over the EcoCAR 4-cycle. E10 is used as the baseline to which E85 and B20 are compared. The stock vehicle model uses a 3.6 L V6 GM LFX engine. The energy consumption result for E10 is found using the method described in Section 3.2.2. The PEU and GHG and criteria emissions are calculated using the methods outlined in the EcoCAR 3 rules [8] while holding the E10 energy consumption result constant for all fuels. Table 3-8 shows the results of the calculations.

While constant energy consumption results provide preliminary information about each fuel, they do not account for engine efficiency changes with different fuels. Based on GM LFX data for both gasoline and E85, there is no appreciable difference in drive cycle efficiency [11]. However, a dedicated E85 engine will operate about 3% more efficiently than an E10 engine due to increased compression ratio [12]. To simulate this slight increase in engine efficiency, the E10 energy consumption is scaled by 0.971 and is reported as the E85 energy consumption for varied engine efficiency in Table 3-8.

A B20 engine is implemented and scaled in the provided stock vehicle model to match the IVM-60 time of the E10 engine, ensuring that the E10 and B20 engines have comparable performance. Additionally, an engine running on B20 will provide the highest efficiency of all three fuels [13]. The stock vehicle model equipped with this scaled B20 engine is then run on the EcoCAR 4-cycle. The resulting EC is reported in Table 3-8.

Table 3-8: Impact of analyzed fuels.

Category	**Energy Consumption Held Constant for Each Fuel**			**Engine Efficiency Varied for E85 and B20**	
	E10	**E85**	**B20**	**E85**	**B20**
Four-Cycle Energy Cons. (kWh/mi)	1.35	-	-	1.31	1.26
GHG_WTP (g/mi)	94	-21	36	-20	33
GHG_PTW (g/mi)	356	349	367	339	344
GHG_WTW (g/mi)	450	328	402	319	378
PEU_WTP (kWh PE/mi)	0.104	0.085	0.102	0.082	0.096
PEU_PTW (kWh PE/mi)	1.25	0.284	1.09	0.276	1.02
PEU_WTW (kWh PE/mi)	1.36	0.369	1.19	0.358	1.12
CO_WTP (g/mi)	0.018	0.010	0.016	0.010	0.015
THC_WTP (g/mi)	0.073	0.066	0.014	0.064	0.013

Category	Energy Consumption Held Constant for Each Fuel			Engine Efficiency Varied for E85 and B20	
	E10	E85	B20	E85	B20
NOx_WTP (g/mi)	0.039	0.023	0.034	0.022	0.032

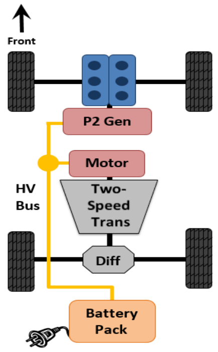
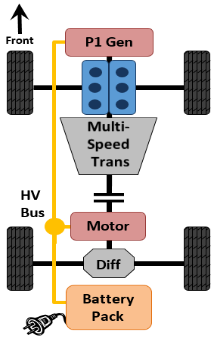
*Abbreviations; COWTP = well-to-pump carbon monoxide emissions; THCWTP = well-to-pump total hydrocarbon emissions; NOXWTP = well-to-pump oxides of nitrogen emissions

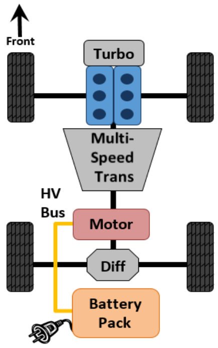
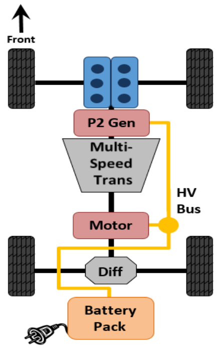
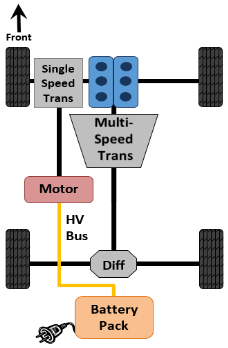
Of the fuel blends, E85 performs the best in all PEU and emission categories with the exception of total WTP hydrocarbon emissions. Additionally, E85 provides a negative WTP GHG emission due to the carbon dioxide gasses that the ethanol source crop removes from the environment during cultivation. E85 is chosen as the fuel for the final hybrid powertrain based on results from this fuel analysis, VTS, and consumer research.

3.3. Concept Generation

Five hybrid powertrain configurations are initially proposed as viable for implementation. The known strengths and risks of the hybrid powertrains are exposed. This initial concept generation is a strong starting point for the concept generation phase. The five options in Table 3-9 use different energy flows, including series, series-parallel, and parallel-through-the-road.

Table 3-9: Potentially viable powertrain configurations.

Powertrain	Vehicle Powertrain Architecture Diagram	Benefits	Risks
Series PHEV 3.6 L V6 WT DI P2 Generator- 30 kW 12 kWh plug in battery pack P3 Motor – 100 kW Peak		• Engine operates at optimal efficiency • Idle charging with P2 generator • Space in transmission tunnel for generator packaging • Abundant space in engine compartment for inverter packaging • Additional GM vehicles to model rear trans-axle design	• Implementation of motor and rear trans axle design • No mechanical connection between engine and wheels- single point of failure • Potentially low performance depending on power output of ESS, motor and generator • Potentially not capable of meeting vehicle technical specifications such as gradeability
Series-Parallel PHEV 3.6 L V6 WT DI P1 Motor-20 kW P3 Motor – 125 kW Peak 12 kWh plug in battery pack		• Regenerative braking capabilities • Idle charging with P1 generator • Increased tractive power using both motor and engine • Improved efficiency through optimal engine loading	• Clutch on transverse axle may cause operational malfunctions • Mechanical torque couplings increase design complexity • Lack of efficient component space for P1 generator

Powertrain	Vehicle Powertrain Architecture Diagram	Benefits	Risks
Turbocharged Parallel PHEV 3.6 L V6 VVT DI Twin Turbo P3 Motor – 100 kW Peak 12 kWh plug in battery pack		• Regenerative braking capabilities • Increased tractive power using both motor and engine • Improved efficiency through optimal engine loading • Improved engine efficiency and power output due to turbocharger	• Multiple points of failure because of torque coupling interfaces • Limited engine compartment space for twin turbo setup
Series-Parallel PHEV 3.6 L V6 WT DI P2 Generator-30 kW P3 Motor – 100 kW Peak 18.8 kWh plug in battery pack		• Regenerative braking capabilities • Idle charging with P2 generator • Increased tractive power from motors and engine • Improved efficiency through optimal engine loading • Efficient power generation of P2 generator by avoiding transmission losses	• Implementation of P2 between engine and transmission due to removal of torque converter • Multiple points of failure because of mechanical torque couplings
Parallel-Through-the-Road PHEV 3.6 L V6 WT DI P4 Motor – 125 kW Peak 12 kWh plug in battery pack		• Three possible modes: EV only, Conventional RWD, and a combination of both • Ability to retain stock drivetrain, except engine may be swapped for E85 V6Regenerative braking capabilities	• Complex packaging with motor and additional transmission on front axle • Weight distribution with majority of weight distributed over the front end of the vehicle

After initial concept generation, four specific powertrains (different from initial five) are analyzed in more detail to generate sets of VTS that can be compared to drive the down-selection process of the final hybrid powertrain. Packaging and mass considerations are also critical to the viability of each proposed powertrain in this extensive and more detailed analysis. The four proposed powertrains are detailed in Figure 3-7. The VTS for each proposed powertrain are shown in Table 3-10**.** Note that all powertrains are modeled with a curb mass of 2020 kg for consistency of results. In addition, the weight limits required by EcoCAR 3 were unknown at the time of the modeling activities because weight information was not available from GM. Thus, the maximum possible weight is used for modeling.

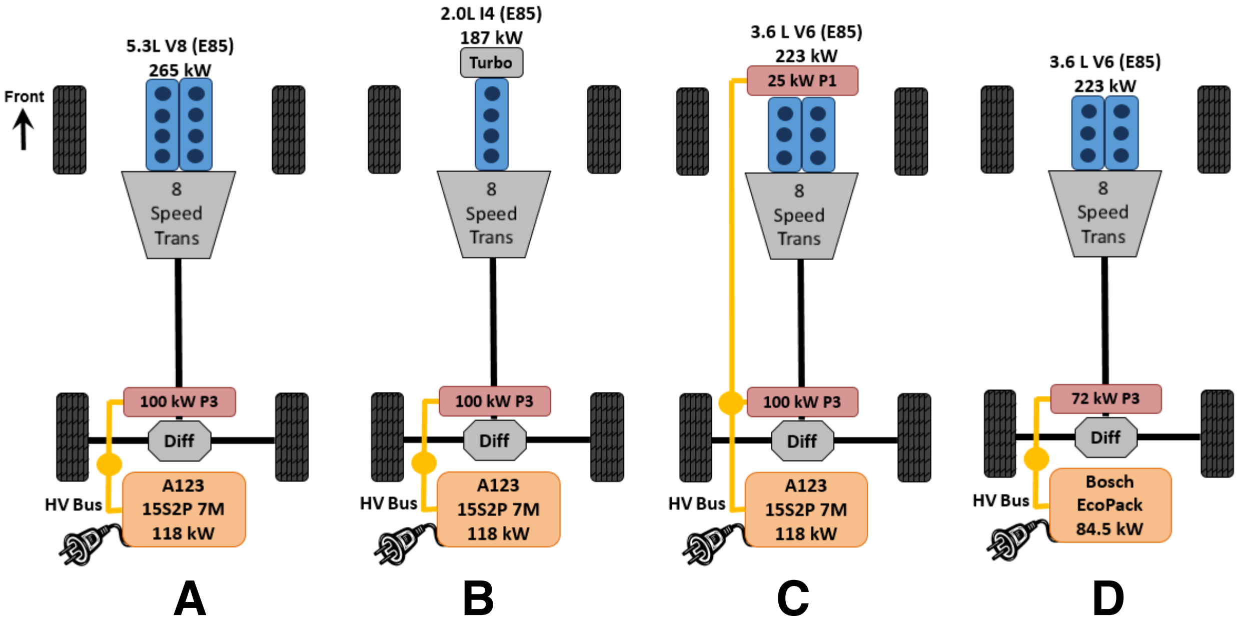


Figure 3-7: Diagrams of powertrains analyzed in detail.

Table 3-10: Vehicle technical specifications for all extensively analyzed powertrains.

Specification	**Competition**		**HEVT Target**	**Powertrain Configuration**			
	Req.	**Target**		**A P3 V8**	**B P3 I4**	**C P1/P3**	**D P3 Bosch**
Acceleration, IVM-60 mph, sec	7.9	5.9	5.0	3.8	4.6	4.4	4.9
Acceleration, 50-70 mph (Passing), sec	9.9	7.3	5.0	4.2	4.7	4.7	4.9
Braking, 60-0 mph, ft	135	128	128	<128	<128	<128	<128
Acceleration Events Torque Split (Front/Rear)	49% F 51% R	0% F 100% R	0% F 100% R	0% F 100% R	0% F 100% R	0% F 100% R	0% F 100% R
Lateral Acceleration, 300 ft. Skid Pad, G	0.80	0.85	0.85	NM	NM	NM	NM
Double Lane Change	52	55	58	NM	NM	NM	NM
Highway Gradeability, @ 20 min	6% @ 60 mph	6% @ 60 mph	6% @ 60 mph	6% @ 60 mph	6% @ 60 mph	6% @ 60 mph	6% @ 60 mph
Cargo Capacity, ft^3	2.4	--	>2.4	6.4	6.4	6.4	7.8
Passenger Capacity	2	4	4	4	4	4	4
Curb Mass, lb (kg)	TBD	--	≤ Comp. Req	4450 (2020)	4450 (2020)	4450 (2020)	4450 (2020)
Starting Time, sec	15	2	2	3 (est)	3 (est)	3 (est)	3 (est)
Total Vehicle Range*, mi (km)	150 (226)	--	187 (300)	189 (304)	257 (413)	221 (355)	206 (331)
CD Mode Range*, mi (km) Pure EV	N/A	--	22.5 (36.2)	25.5 (41.1)	25.5 (41.1)	25.5 (41.1)	20.6 (33.2)
CD Mode Total Energy Consumption*, AC Wh/km (mpgge)	N/A	--	280 (74.8)	275 (76.1)	275 (76.1)	275 (76.1)	262 (79.9)
CS Mode Fuel Consumption*, Wh/km	N/A	--	840 (24.9)	810 (25.9)	573 (36.6)	680 (30.8)	716 (29.3)

Specification	Competition		HEVT Target	Powertrain Configuration			
	Req.	Target		A P3 V8	B P3 I4	C P1/P3	D P3 Bosch
(mpgge)							
UF-Weighted Fuel Energy Consumption*, Wh/km (mpgge)	N/A	--	480 (43.6)	429 (48.8)	303 (69.1)	368 (57.0)	425 (46.6)
UF-Weighted AC Electric Energy Consumption*, AC Wh/km	N/A	--	120	130	130	130	106
UF-Weighed Total Energy Consumption*, Wh/km (mpgge)	840 (25)	700 (30)	600 (35)	558 (37.5)	433 (48.4)	489 (42.8)	532 (39.4)
UF-Weighted WTW Petroleum Energy Use*, Wh PE/km	750	420	150	122	87.3	103	120
UF-Weighted WTW Greenhouse Gas Emissions*, g GHG/km	250	225	175	168	137	151	156
UF-Weighted WTP Criteria Emissions* g/km (score)	TBD	TBD	≤ Comp. Req	0.0418 (24.9)	0.0325 (33.4)	0.0709 (28.4)	0.0415 (24.4)
UF-Weighted WTW Criteria Emissions* g/km (score)	TBD	TBD	≤ Comp. Req	NM	NM	NM	NM

*Evaluated by using the EcoCAR 3 combined “4-cycle” weighting method. NM – Not Modeled, IVM – Initial Vehicle Movement, UF – Utility Factor, WTW – Well to Wheel, WTP – Well to Pump, mpgge – Miles Per Gallon Gasoline Equivalent, P1- Position 1, P3 - Position 3

3.4. Final Selection of Hybrid Powertrain Configuration

Selection of a hybrid powertrain configuration is a critical step in the VDP. The final decision must be driven by the results of the engineering calculations that show which proposed powertrain best meet design goals. A detailed selection matrix is necessary to compare the fuel economy, emissions, performance, risk, and resources associated with the four proposed powertrains. The matrix shown in Table 3-11 is divided into three categories: fuel economy/emissions, performance, and risk/resources.

Table 3-11: Powertrain Architecture Selection Matrix

	Metric	Points	Weight	Architecture			
				A	B	C	D
Fuel Economy/ Emissions	Energy Consumption	15	6.00%	5	3	4	5
	WTW Greenhouse Gases	15	6.00%	3	5	4	3
	Petroleum Energy Usage	15	6.00%	3	5	4	3
	WTW Emissions	15	6.00%	3	5	4	3
	Over-the-road Event	12.5	5.00%	3	5	4	3
	Subtotal	72.5	29.00%	**247.5**	**332.5**	**290.0**	**247.5**
Performance	IVM to 60	10	4.00%	5	3	4	2
	50 to 70	10	4.00%	5	3	4	2
	Braking	10	4.00%	5	4	4	4
	Handling (Lane Change+Max Lat.)	7.5	3.00%	5	4	4	4
	Autocross	12.5	5.00%	5	3	4	2
	Ride Quality (DCA)	10	4.00%	3	3	3	3
	AVL DRIVE Quality	10	4.00%	3	3	3	3

	Metric	Points	Weight	Architecture			
				A	B	C	D
	Subtotal	70	28.00%	**310.0**	**227.5**	**260.0**	**195.0**
Risks/Resources	Packaging Complexity	10	4.00%	4	4	3	5
	Waiver Complexity	10	4.00%	4	4	4	5
	Team Facilities	10	4.00%	5	4	5	5
	Team Experience	15	6.00%	5	3	4	5
	Engineering Value	20	8.00%	5	5	5	4
	Component Availability	10	4.00%	4	4	3	5
	Systems/Controls Complexity	10	4.00%	4	3	2	5
	Consumer Appeal	20	8.00%	5	3	2	1
	Competition Cost	2.5	1.00%	3	2	4	5
	Subtotal	107.5	43.00%	**492.5**	**400.0**	**380.0**	**437.5**
	Total	250	100.00%	**1050.**	**960.**	**930.**	**880.**

The selection matrix contains a total of 250 points. All powertrains are rated for each category as one through five. A rating of 1 is the most undesirable rating and 5 is the most desirable rating. The point breakdown and weight distribution for the fuel economy/emissions and performance categories are derived from the EcoCAR 3 Year 4 dynamic events points. The point distribution for risks/resources category is subjectively determined because there is no competition point distribution as in the case of dynamic events; this logic is the reason why metrics like packaging complexity, team experience, engineering value, and consumer appeal are heavily weighted. The emissions category is not as highly weighted as the other categories because the emphasis in EcoCAR 3 slightly shifted from decreasing emissions and energy consumption to improving performance and capturing the target market.

Finally, The results from the selection matrix prove Powertrain A to be the most effective hybrid configuration in meeting design goals due to its increased performance characteristics. In addition, Powertrain A has the highest total number of points and outscores all of the other proposed configurations in the performance and risks/resources categories.

4. Plant Model Development for Selected Hybrid Powertrain

The control system development for the selected powertrain requires creating a control algorithm that can safely and effectively operate the hybrid vehicle. A vehicle model in software is necessary to create a robust, functional, and effective control algorithm. Vehicle models consist of two major parts: the component plant model and the software representation of the component electronic control unit (SoftECUs). Three major testing platforms are used to develop and refine the hybrid vehicle control code with the developed vehicle model. The platforms used are: MIL, SIL, and HIL.

MIL is used in a virtual environment. A unique characteristic of this platform is that the model used in MIL and the final control code that will operate on the vehicle can be completely different. MIL serves as a starting point to develop models, obtain results, and make high-level design decisions for powertrain selection. As detailed in Section 3 of this study, MIL simulations are conducted to create performance validation metrics for SIL and HIL modeling activities.

SIL is the next step in the process of developing vehicle models. In SIL, the control code and vehicle model are constructed in a software environment (Simulink) that can eventually be compiled and uploaded to the target controller and a simulator respectively. SIL testing occurs in a virtual environment, and the model is executed at a fixed time step to better reflect how the communication signals are transmitted in the vehicle. In addition, an SIL models has an HVSC code that communicates with the plant model only through a bus of signals. This separation between the plant and controller models imitates the I/O of the HVSC in the vehicle.

HIL is the last step in vehicle modeling and control development before vehicle testing begins. The compiled control code from SIL is uploaded onto the actual HVSC. The plant model corresponding to the vehicle is then uploaded to a simulator. A connection between the vehicle simulator and the controller is made using an electrical wiring harness. This wiring harness can represent the actual pinout for the I/O in the final vehicle. In this setup, the plant model is run in real time and the target controller executes the code as if it were connected to a real vehicle. HIL serves as preliminary validation for control code, and allows for testing of safety critical features.

This section focuses on the details and theory of component plant models and SoftECUs. Validation of results of the models against data gathered from component bench testing is covered in Section 6. The model development strategy in this study follows a typical V-cycle shown in Figure 4-1. First, the vehicle level requirements are defined based on design goals and desired VTS. These vehicle level requirements are then broken down into subsystem level requirements and finally component level requirements. Furthermore, control and model architectures are generated and tested based on component level requirements. From testing results, the model and control code are refined until the output of the model meets the vehicle requirements.

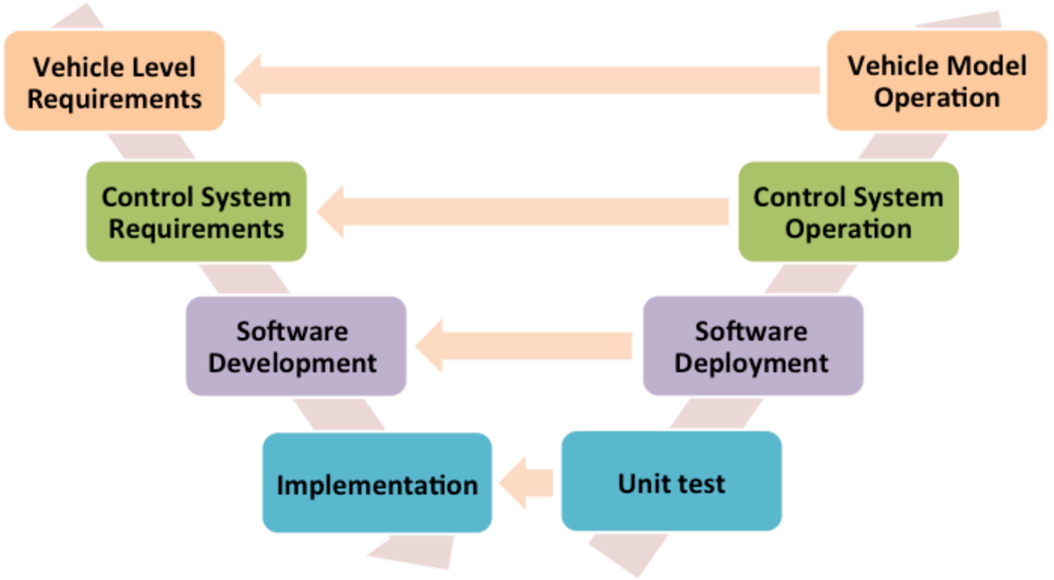


Figure 4-1: Diagram of V-cycle implemented for model development.

4.1. Driver Controls

The vehicle model requires a subsystem that can regulate power demand from the powertrain based on the drive cycle being tested. A driver model uses a proportional integral derivative (PID) controller to minimize the error between drive cycle speed and measured vehicle speed. The PID controller outputs a command to the torque producing components in the powertrain to determine the tractive or braking power needed to meet the cycle trace. In conventional and hybrid vehicle models, the driver PID outputs an accelerator pedal position (APP) and brake pedal position (BPP), however the torque split for the hybrid vehicle is performed by the HVSC. In addition, the driver controller response is characterized by three gains: proportional (P), integral (I), and derivative (D). Figure 4-2 shows the structure of the PID controller implemented as a driver for the vehicle model.

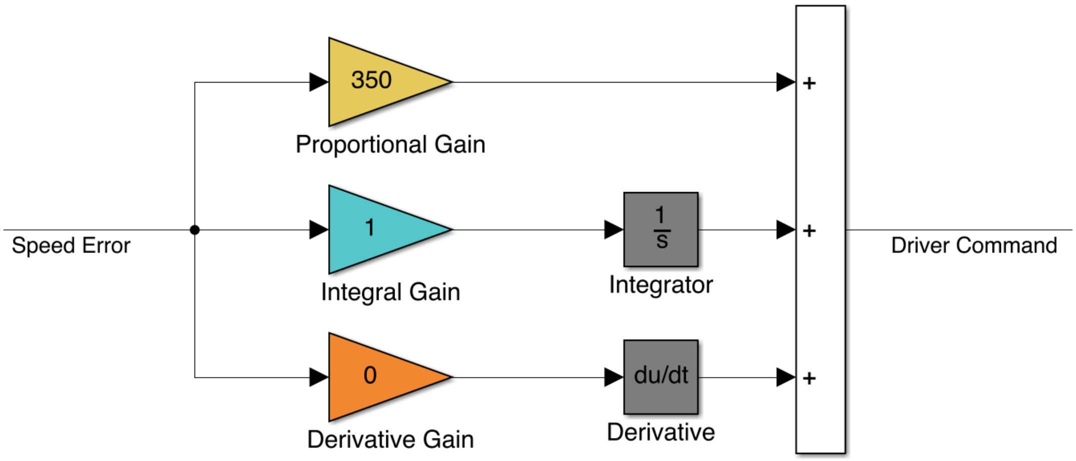


Figure 4-2: PID controller structure.

The values of the PID gains are tuned using the Ziegler-Nichols method, which first defines a value for the P gain, and then it determines the value of the I and D gains respectively based on the desired and actual response [14]. The gains used for the driver PID controller in this study are listed in Table 4-1.

Table 4-1: PID gains.

Gain	Value
Proportional	350
Integral	1
Derivative	0

The derivative gain for the driver subsystem is zeroed because the response given by the PI controller is enough to meet the evaluated drive cycle to acceptable standards. Trace misses are used to evaluate how well a vehicle or model meets a certification drive cycle. A vehicle/model is required to meet the cycle trace within ±2 mph for all legs of the EcoCAR 4-cycle, except for the US06 cycle. The US06 accepts trace misses larger than ±2 mph as long as the vehicle/model is trying to meet the cycle trace at best effort.

4.2. SoftECUs

This subsection provides a brief description of the used SoftECUs in the vehicle model. The development of SoftECUs is driven by the need to represent the functionality of component controllers. In vehicles, the control unit of a specific component acts as a communicator with other controllers, and transmits commands/sensor feedback to/from the component. For instance, the engine control unit in a vehicle communicates important engine torque status information to the transmission control module (TCM) to ensure smooth gear shifting. In addition, the engine control unit translates the accelerator pedal position into a throttle valve opening while reading the feedback from throttle position sensor. The component SoftECUs model the signal processing done by the real ECUs in the vehicle. Additionally, SoftECUs can run specific diagnostics and determine rates or limitations for the component plant model.

4.2.1. Engine SoftECU

The engine SoftECU structure is shown in Figure 4-3. All signals from the high-speed GM bus (HS GMLAN) are accounted for in simulation, but only a few are active in the model. The active received (Rx) signals are routed to the decision structure, a state machine created with Stateflow blocks. The decision structure contains states for startup and shutdown as well as AFM logic. Rx CAN signals containing HVSC requests are renamed as physical signals and are routed to the plant model. Engine status signals are returned to the decision structure, renamed as actual CAN signals, and transmitted (Tx) back to the HVSC.

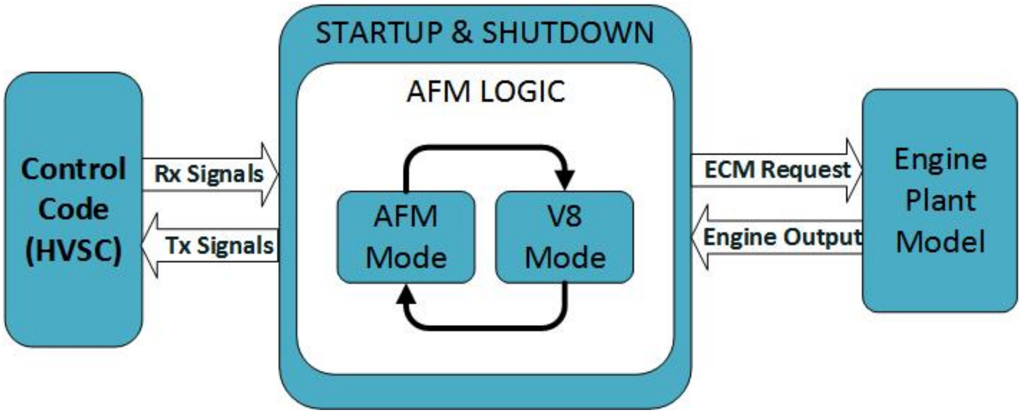


Figure 4-3: Engine SoftECU logic and communication structure.

Inside the controller logic is a PowerOff and PowerOn section. The PowerOff represents when the vehicle is on, but the engine has not be started (utilized in accessory mode in most real vehicles). The only transition from Off to On is a simple “engine crank” voltage, normally sent by the body control module (BCM) or the HVSC. The other transition between these overarching engine states is a loss of engine power (controlled from outside the model by the HVSC). While the engine is on, an AliveRollingCounter is on, continuously increasing in value, until it resets once a maximum value is reached (1 to 4). This counter is simply for verification that the engine module is on and functioning by other modules on the CAN bus.

The simulated AFM logic starts with the engine on, with all eight cylinders active. The logic constantly checks conditions to enter AFM Active status. To enter and remain in AFM mode, the conditions in Table 4-2 must be met and maintained. These parameters were are based on requirements for AFM activation found in GM service information and verified through in-vehicle testing with a 2014 Silverado.

Table 4-2: Stateflow AFM conditions

Condition	Lower Limit	Upper Limit
Transmission Gear	3rd Gear	8th Gear
Transmission Shift Lever Positon (PRNDL)	Drive	Drive
Engine Speed	700 RPM	2800 RPM
Vehicle Velocity	25 kph	N/A
Accelerator Pedal Positon	0 %	15 %
Engine On	30 seconds	N/A

Once all the conditions have been met for AFM, the model waits one simulation cycle time and the active state switches. Only one signal is changed from within the Stateflow diagram: AFM state. The AFM state signal is used by the engine plant model to determine which calculations to perform. If at any time one of the enumerated parameters falls outside the defined limits, the system will return to all standard 8-cylinder operation. AFM mode will also end after 10 minutes (as suggested by documentation). The engine will then wait 60 seconds, and will re-enter the mode. Once the engine has been removed from AFM, it can re-enter after 30 seconds, and once the optimal ranges for the conditions are met.

4.2.2. Transmission SoftECU

In an automotive system, the transmission control module is responsible for selecting a gear based on driver demand, vehicle speed, engine ability to produce torque and other parameters. In addition, a transmission control module must handle torque transmission during shifting and must control torque converter and torque converter clutch operation (or interact with a torque converter control module, if present). The developed SoftECU selects a gear, thus a gear ratio, by looking at vehicle speed, APP, and gear lever position. The model is simple compared to real vehicle operation, and torque converter and torque converter clutch operation is not modeled. The model of the transmission control module uses shift schedules that are very specific to the 8-speed transmission used in the HEVT vehicle. These shift schedules (provided by General Motors) determine a vehicle velocity threshold for upshifting/downshifting gears based on driver demand and vehicle speed. Note that a real TCM in a vehicle determines the shifting thresholds by looking at transmission output speed instead of vehicle speed, although both are related by the final drive ratio and the wheel radius.

4.2.3. Motor SoftECU

The motor system SoftECU is responsible for I/O and control logic for the motor plant model. The SoftECU is developed using CAN database information and predicted performance specifications provided by InMotion, the motor manufacturer. The SoftECU is functional and communicates with the control code, battery plant model and SoftECU, and motor plant model, as shown in Figure 4-4. Stateflow logic is used to program the SoftECU, enabling traction and regenerative braking modes within the vehicle model. An electromagnetic (EM) command message from the HVSC is sent to the SoftECU that specifies torque request signal for the motor. EM limits are specified for the maximum motor torque (500 N-m) and power (100 kW). The SoftECU then translates the torque request from the HVSC into a current demand that is sent to the battery.

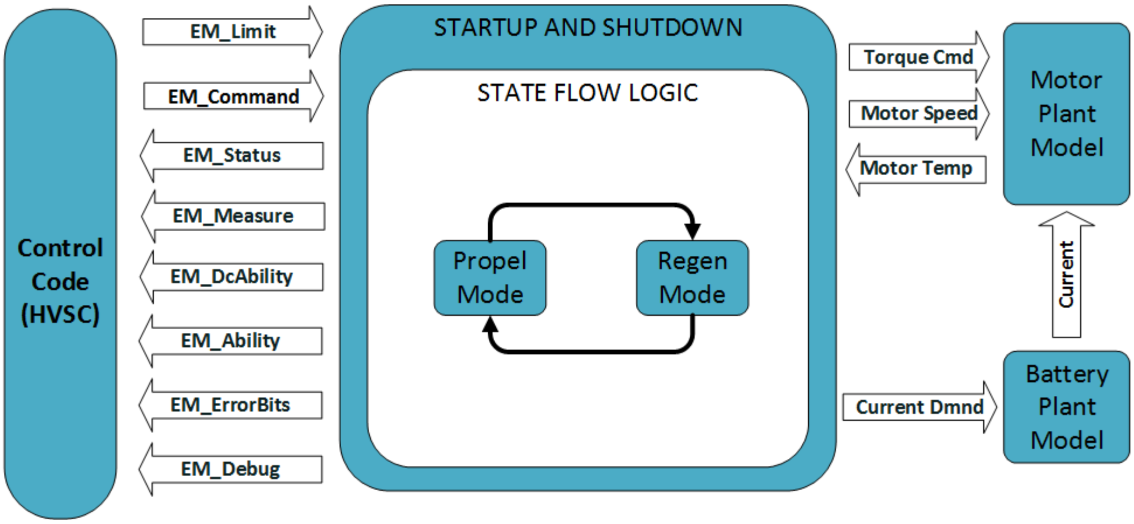


Figure 4-4: Motor SoftECU signal flow and logic diagram.

Charge and discharge buffers and power limits are implemented to keep the output current within acceptable boundaries. The motor plant converts the DC current achieved into a torque output that is transmitted to the driveline and reported to the motor SoftECU. The motor plant model receives a motor speed signal from the motor system SoftECU which is calculated from vehicle speed in the chassis dynamics model. The motor SoftECU transmits back to the HVSC the status of the motor, detailed EM status, ability of the motor to produce torque and mechanical power, and errors or faults encountered. In actual vehicle operation the inverter sends AC current to the motor. However, DC to AC current inversion is not modeled for the motor system SoftECU.

4.2.4. <u>High Voltage Battery SoftECU</u>

The battery SoftECU, or energy storage control module (ESCM), is responsible for I/O and control for the battery plant model. The logic contained in the battery SoftECU is nearly identical to the ESCM to be used in the vehicle, as the SoftECU is developed with documentation provided by A123 systems. The overall structure of the battery SoftECU is shown below in Figure 4-5. The ESCM consists of four subsections: startup and shutdown sequence, contactor logic and control, thermal and state of charge checks, and battery status. The aforementioned logic procedures are performed in Stateflow where a request message containing an enable command, emergency power off (EPO) command, and a close contactors command is sent from the HVSC to the ESCM. The ESCM sends status, data, and fault messages back to the HVSC. The SoftECU receives measured values from the plant model including current delivered, SOC, voltage, and temperature. The battery plant model receives a current demand from the motor system and returns a current within allowable limits.

Battery system functionality is verified by testing the motor model, battery model, and control code as a battery electric vehicle (BEV) over standard drive cycles. CAN communication between the ESCM and HVSC physically occurs though VEH_CAN pins and are modeled in SIL on a virtual bus. The signals are transferred through the wiring harness connected to the HVSC in the HIL model. Additional messages sent between the ESCM and the NLG5 charger exist, but are a low priority, not required to run the vehicle model, and are not currently in development. According to interface component documentation (ICD), the ESCM is included on the MOD_CAN network, an internal communication channel used only by the ESS components. For simplicity, this network is not modeled, and the seven battery modules are lumped together as one. The functions of individual modules and sensors are contained either in the SoftECU or the plant model.

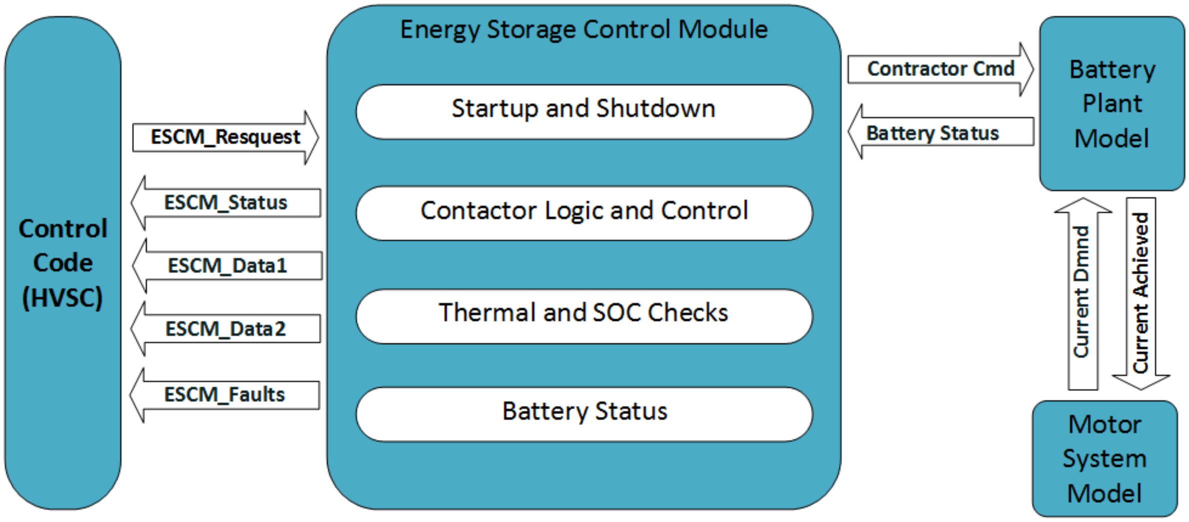


Figure 4-5: ESCM Signal Flow and Logic Diagram

4.2.5. DC/DC Converter SoftECU

The DC/DC converter SoftECU performs the startup and enable logic for the plant model of the component. In addition, this SoftECU continuously monitors the operating conditions of the converter and reports when faults occur. The startup sequence in the controller model starts by monitoring the wakeup current; if that current is present, then the SoftECU enters Buck Mode operation, where the DC/DC converter only steps high voltage to low voltage. During operation, the controller model performs the following interval checks (in the written order) to ensure that all variables are within acceptable range and no faults are present: high voltage, low voltage, low voltage current, high voltage current, temperature, and setpoint command from HVSC. The need to monitor voltages and currents separately arises because the model treats both variables independently. If any of the variables are outside of range, a fault is triggered and all conversion operation is terminated. If no faults are present and the DC/DC SoftECU is receiving an enable command, then the high to low voltage conversion is performed and all checks are continuously executed.

4.2.6. High Voltage Battery Charger SoftECU

The functionality of the HVBC SoftECU is very similar to that of the DC/DC converter controller model. The charger SoftECU receives a pilot enable command from the from the charging station (external input in model), which initializes charging operation. The following variables are continuously monitored during charger operation to ensure that no faults are present in the subsystem: enable command present, charger power output, charger temperature, and lower voltage limit for the charger. If all values are within acceptable limits, then normal operation mode is entered and the HVBC SoftECU commands its plant to send current to the ESS.

4.3. Plant Models for Powertrain & Auxiliary Components

The physical operation of different components is represented using mathematical models. This section presents an overview of the implemented theory and models. All plant models used for SIL and HIL testing are developed in MATLAB and Simulink. Generally, MATLAB scripts contain the variables and data necessary to run the models; these scripts are called initialization files. Conversely, Simulink is the graphic coding environment in which the mathematical calculations are performed to model component operation. The component plant models communicate heavily with their respective SoftECU. Additionally, plant models can interact with each other to represent vehicle system operation. For instance, the engine plant model receives commands and communicates sensor feedback from and to its SoftECU, and also generates a torque output that is sent to the transmission plant. Initially, plant models and SoftECUs are developed for powertrain and driveline components only. Auxiliary components are added at a later stage to better represent the operation of a hybrid vehicle. The plant models implement power/torque loss principles supplemented with lookup tables for data and other Simulink tools that facilitate modeling and result generation.

4.3.1. Engine Plant Model

The L83 engine plant model is built in Simulink using brake mean effective pressure (BMEP) scaling, lookup tables (LUT), and MATLAB functions. The first step in the engine plant model is to determine the output mechanical power requested by the driver model. In a conventional vehicle model, the driver subsystem generates an APP fraction between zero and one. That number is used to meet driver demand by scaling the wide-open throttle (WOT) curve of the engine times the APP fraction. In the Simulink environment, the torque point in the engine WOT curve is selected by inputting engine speed to a 1-D LUT and the torque output of the LUT is multiplied by the APP fraction. In the hybrid vehicle model, the engine torque command is generated by the HVSC based on the total torque demand from the driver subsystem. Figure 4-6 shows the WOT curve of the 5.3L engine on E10 fuel.

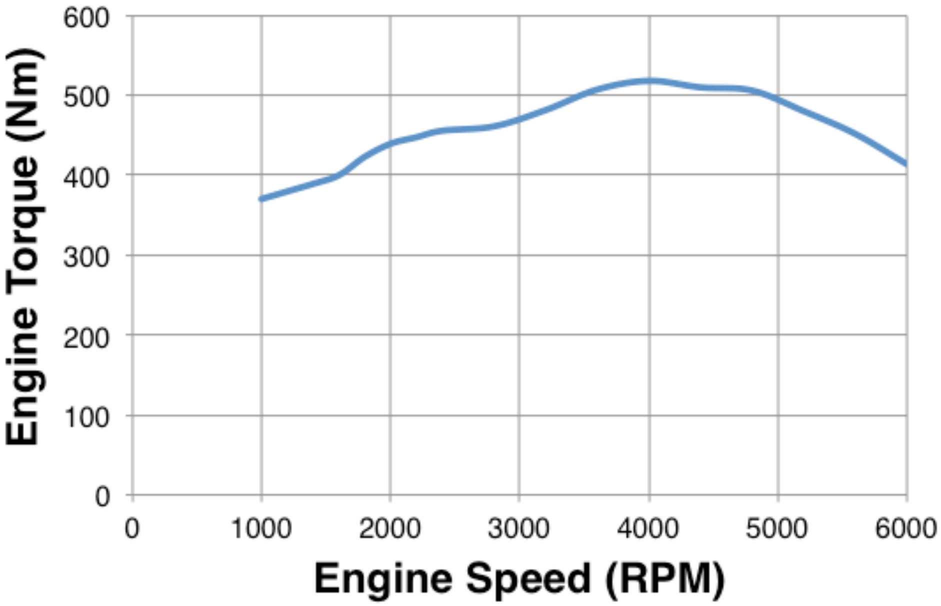


Figure 4-6: Engine WOT curve based on Engine Speed.

Equation 4-1 defines brake mean effective pressure,

$$BMEP = \frac{4\pi\, T_{eng}}{V_d} \qquad \text{Equation 4-1}$$

where T_{eng} and V_d respectively are the torque and volumetric displacement of the engine. The driver requested torque, T_{eng}, and engine speed, ω, are then used to calculate the instantaneous engine efficiency, η_{eng}, according to Equation 4-2,

$$\eta_{eng} = \frac{\eta_{therm}}{1+\frac{fmep_0+fmep_1\omega+fmep_2\omega^2}{BMEP}} \qquad \text{Equation 4-2}$$

where η_{therm}, *BMEP*, *fmep0*, *fmep1*, and *fmep2* are the engine thermal efficiency, brake mean effective pressure, and friction mean effective pressure coefficients respectively. The friction mean effective pressure coefficients used to model the 5.3L engine are given in Table 4-3. Further detail on engine modeling and scaling by displacement is provided in [23].

Table 4-3: Friction mean effective pressure coefficients.

Parameter	Value	Units
η_{eng}	0.44	-
fmep0	140	kPa
fmep1	0	kPa/(rad/s)
fmep2	$2.23\text{x}10^{-4}$	kPa/(rad/s)^{2}

Knowing the engine power output and efficiency allows for calculation of the fuel power input and instantaneous fuel use. The instantaneous input and output power are integrated to calculate the input and output energy used over a drive cycle. The instantaneous fuel rate is integrated to determine the total amount of fuel used over a drive cycle. Energy consumption results in Wh/mi are obtained by normalizing energy use by the distance the vehicle travels over the drive cycle. Figure 4-7 shows contours of engine efficiency with torque and speed operating points in P3 hybrid vehicle model. Note that the efficiency contours shown in Figure 4-7 are obtained by scaling fuel rate data provided by MathWorks in Year 1 of EcoCAR 3.

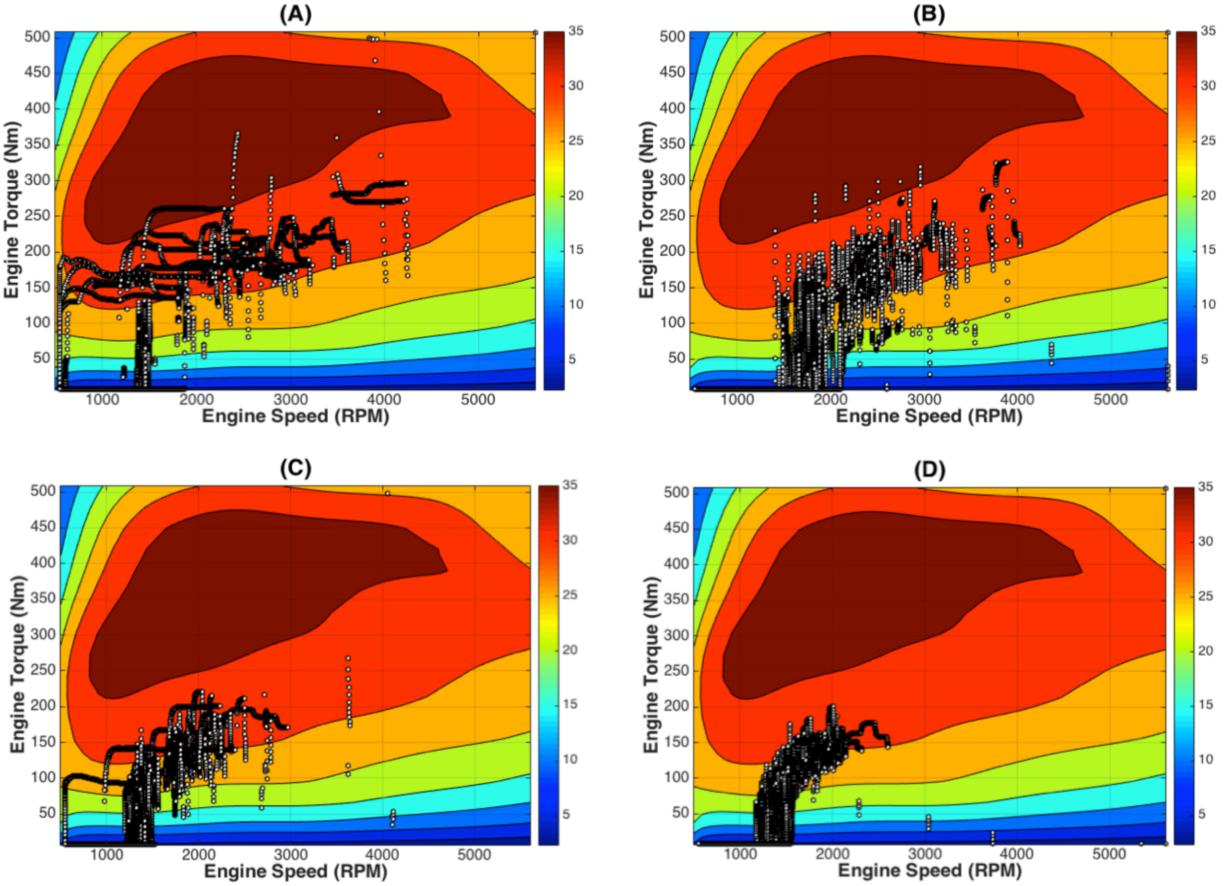


Figure 4-7: Engine model efficiency contour with torque and speed operating points in conventional vehicle model. (A) US06 City drive cycle. (B) US06 Highway drive cycle. (C) 505 drive cycle. (D) HWFET drive cycle.

4.3.2. Transmission Plant Model

The model used to represent the transmission operation is a torque loss model that is centered on a reference speed of 2000 RPM. The transmission input torque is the known output from the engine. Knowing the torque input and the torque losses, the torque output of the transmission, T_{out}, can be calculated from Equation 4-3. The transmission model does not account for torque converter or torque converter clutch operation, therefore it is a simple model.

$$T_{out} = T_{in} + T_{loss} \qquad \text{Equation 4-3}$$

where T_{in} and T_{loss} are the input toque and torque loss respectively. Furthermore, the torque losses in the transmission are calculated with Equation 4-4, which uses coefficients (C_0, C_1, and C_2) in conjunction with reference torque and speed values (T_{ref} and S_{ref}) known from testing [15].

$$T_{loss} = C_0 + \frac{C_1 T_{in}}{T_{ref}} + \frac{C_2(S-S_{ref})}{S_{ref}} \qquad \text{Equation 4-3}$$

The values used for Equation 4-3 are provided in Table 4-4.

Table 4-4: Transmission model characteristics.

Transmission Model Characteristics	
Coefficients & Reference (Units)	**Value**
C_0 (Nm)	8
C_1	10
C_2 (Nm/RPM)	4
T_{ref} (Nm)	200
S_{ref} (RPM)	2000

Figure 4-8 shows transmission losses in different gears for maximum engine torque. Engine torque is relevant because it is the input torque to the transmission (torque converter or torque converter clutch not modeled).

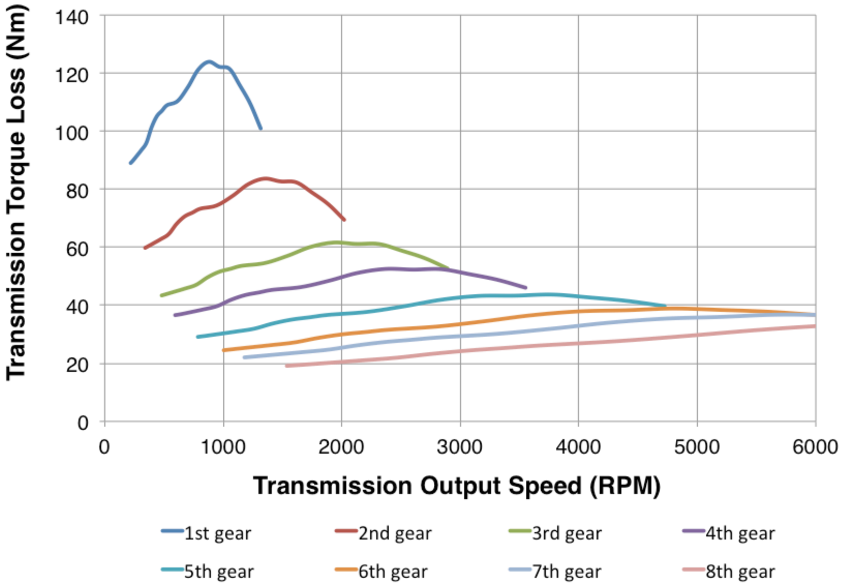


Figure 4-8: Transmission torque losses in all gears at engine WOT.

The shifting response of the transmission model in a 0-60 mph acceleration test is shown in Figure 4-9. The shift maps used are not shown as they contain proprietary information that belongs to General Motors.

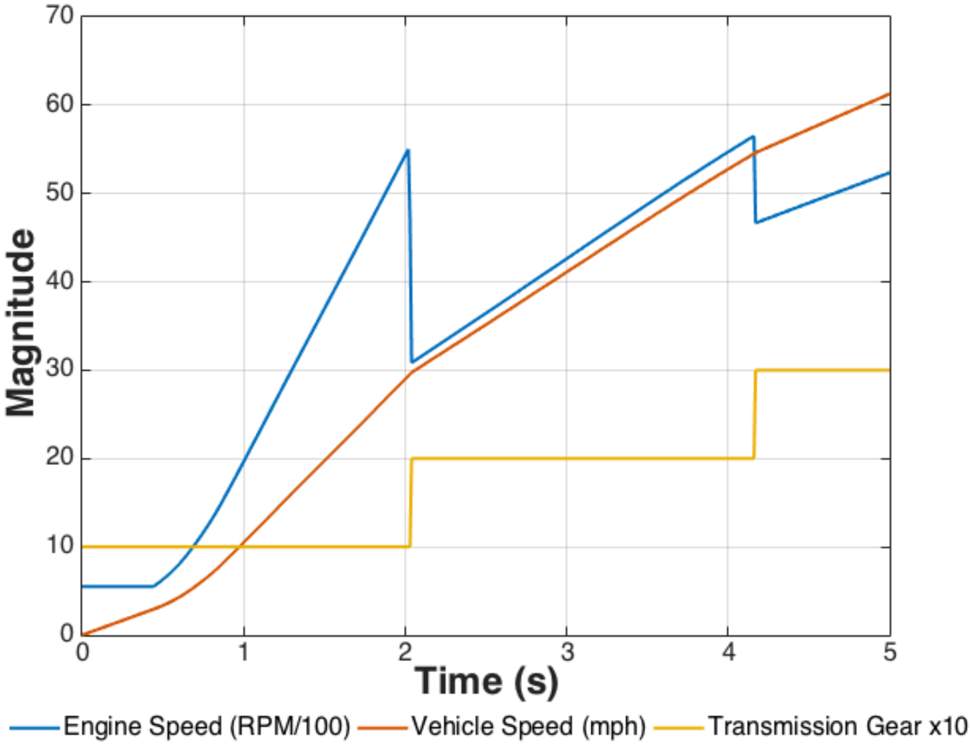


Figure 4-9: Shift behavior of transmission model.

In the acceleration test, the engine speed maxes out at 5600 RPM. This behavior is caused by adequate tuning of the transmission shift maps. If the shift maps are not modified in the model, the engine speed would attempt to exceed the engine speed limit of 5600 RPM. However, for actual vehicle implementation, the shift behavior of the TCM would have to be modified. Advanced external assistance would be required for modification of the TCM as HEVT does not have the capabilities to develop or deploy TCM calibration software. If the transmission is not modified, there is no major negative impact on the design goals. The expected in-vehicle behavior for part load driving is that the transmission would upshift sooner, which can help load the engine to a more efficient operating point.

4.3.3. Motor Plant Model

The motor plant model in this study utilizes the concept of power loss highlighted in [15]. The power loss technique can be applied to calculate the torque and speed characteristics of motors with different power outputs. If operating characteristics are known for a specific motor, the characteristics of a different motor can be determined by power scaling. The P3 motor is designed to output 100 kW. Equation 4-4 gives the mechanical power output of the motor, P_{mot},

$$P_{mot} = T_{mot}\omega_{mot} \quad \text{Equation 4-4}$$

where T_{mot} and ω_{mot} are the motor torque and speed respectively. Equation 4-5 is used to calculate the power losses of the scaled motor of interest, P_{loss_scale}. The P3 motor manufacturer, InMotion, provided data points that are used to determine the values and coefficients in Equation 4-5.

$$P_{loss,scale} = \left[\frac{T_{max}\omega_{base}}{T_{ref}\omega_{ref}}\right] [K_p \left|(T_{ref}\omega_{ref})(\frac{T\omega}{T_{max}\omega_{base}})\right| + \ k_c \times (1+\alpha(Temp_{wind} - Temp_{ref})) \times T_{ref}^2 \left(\frac{T}{T_{max}}\right)^2 + k_i \omega_{ref} (\frac{\omega}{\omega_{base}}) + k_w \omega_{ref}^3 \left(\frac{\omega}{\omega_{base}}\right)^3 + C] \qquad \text{Equation 4-5 (A)}$$

$$P_{loss,scale} = \left[\frac{T_{max}\omega_{base}}{T_{ref}\omega_{ref}}\right] [K_p \left|(T_{ref}\omega_{ref})(\frac{T\omega}{T_{max}\omega_{base}})\right| + \ k_c \times (1+\alpha(Temp_{wind} - Temp_{ref})) \times T_{ref}^2 \left(\frac{T}{T_{max}}\right)^2 \times (\frac{\omega}{\omega_{base}}) + k_i \omega_{ref} (\frac{\omega}{\omega_{base}}) + k_w \omega_{ref}^3 \left(\frac{\omega}{\omega_{base}}\right)^3 + C] \qquad \text{Equation 4-5 (B)}$$

T and ω are the instantaneous torque and rotational speed for the motor. All other variables and values for the reference and scaled motor are provided in Table 4-5. Equation 4-5(A) is used below the based speed, and Equation 4-5(B) is used above the base speed (an additional $\frac{\omega}{\omega_{base}}$ is multiplied with the k_c term).

Table 4-5: Parameter values for motor power loss equation.

Parameter	Value	Units	Description
T_{max}	500	Nm	Maximum torque of motor to be scaled
ω_{base}	199	rad/s	Base speed of motor to be scaled
e	2.71	-	Mathematical constant
k_p	2.4E-7	W/W	Motor loss constant from reference motor
k_c	0.0224	$\frac{s}{Kg\ m^2}$	Motor loss constant from reference motor
k_i	0.74	J	Motor loss constant from reference motor
k_w	4.5E-6	$Kg\ m^2$	Motor loss constant from reference motor
C	100	W	Motor loss constant from reference motor
T_{ref}	500	Nm	Maximum torque of reference motor
ω_{ref}	209.4	rad/s	Base speed reference motor
α	0.01	1/ºC	Temperature coefficient of resistance
$Temp_{wind}$	155	ºC	Windings temperature
$Temp_{ref}$	25	ºC	Reference temperature

Since the motor can have bidirectional flow of energy, the definition of efficiency, η, is different for propel and regen operation. Equation 4-6 and Equation 4-7 provide the mathematical definition of efficiency in both modes.

$$\eta_{propel} = \frac{T_{mot}\omega_{mot}}{T_{mot}\omega_{mot} + P_{loss}} \qquad \text{Equation 4-6}$$

$$\eta_{regen} = \frac{T_{mot}\omega_{mot}+P_{loss}}{T_{mot}\omega_{mot}} \qquad \text{Equation 4-7}$$

Efficiency maps are developed with the power loss modeling technique to study the operation characteristics of the motor model. Figure 4-10 shows contours of motor efficiency with operating point of torque and speed. The plots shown in Figure 4-10 are helpful in analyzing the behavior of the developed motor model over the four drive cycles of interest for EcoCAR 3. Based on the plots, the motor behavior in the P3 model follows the prescribed envelope from torque and power limitations in electric vehicle (EV) operation. The peak torque of the scaled P3 motor occurs at 500 Nm up to a base of 1900 RPM. This operation proves that the motor modeled produces 100 kW of power output. In addition, the operation during regenerative braking can also be observed. Note that a cutoff for regen is implemented for vehicle speeds less than 5 mph, which corresponds to a motor speed of approximately 205 RPM (final drive ratio of 3.27 and tire size of 0.34 m). Regen torque is limited to be negative half of the propel torque based on operation determined historically by HEVT.

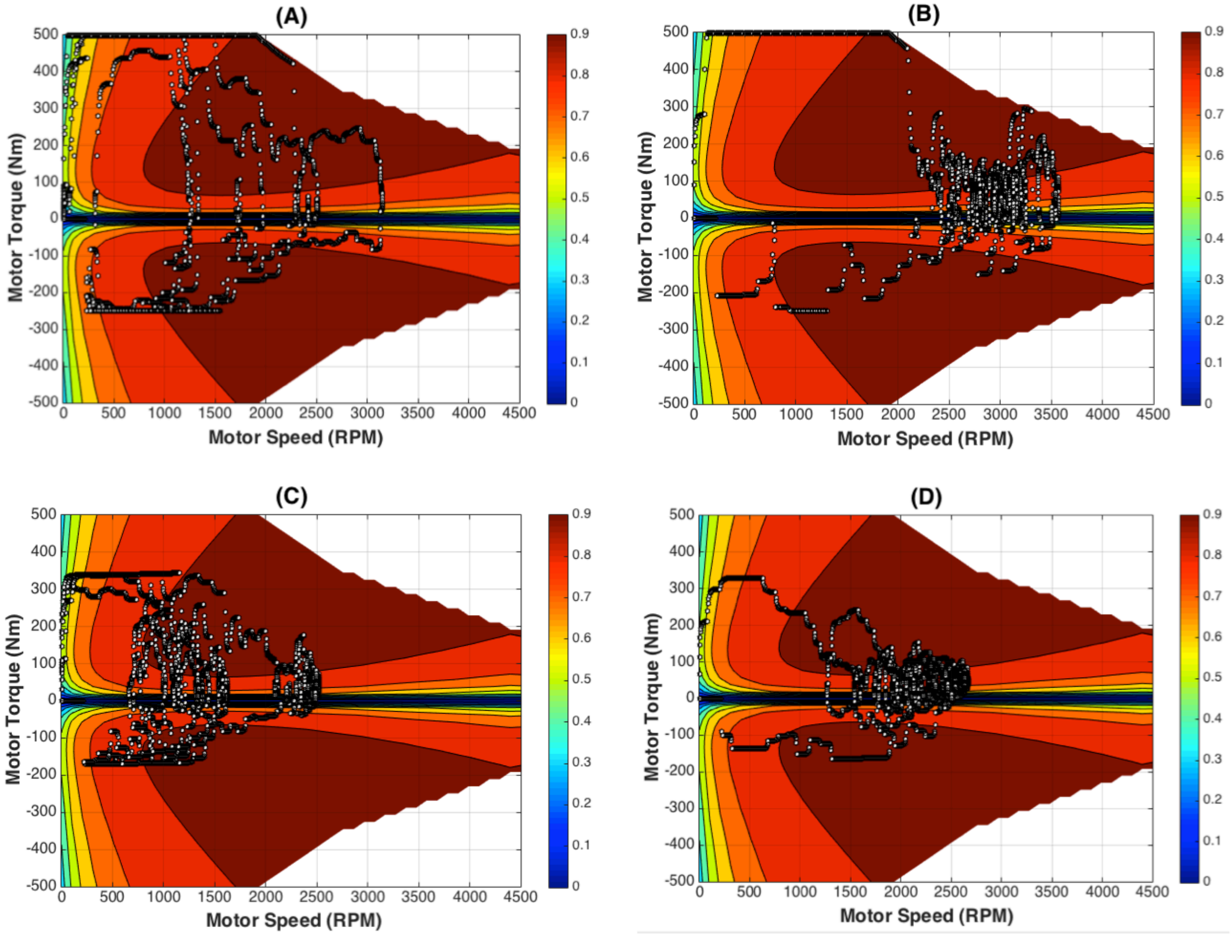


Figure 4-10: Scaled motor efficiency contours with operating points in EV mode. (A) US06 City drive cycle. (B) US06 Highway drive cycle. (C) 505 drive cycle. (D) HWFET drive cycle.

4.3.4. High Voltage Battery Plant Model

A simple internal resistance model is used for the high voltage battery to calculate SOC and energy losses due to the internal resistance of the supply. Figure 4-11 shows the representation for the high voltage battery in this study. Further detail on the model can be found in [15].

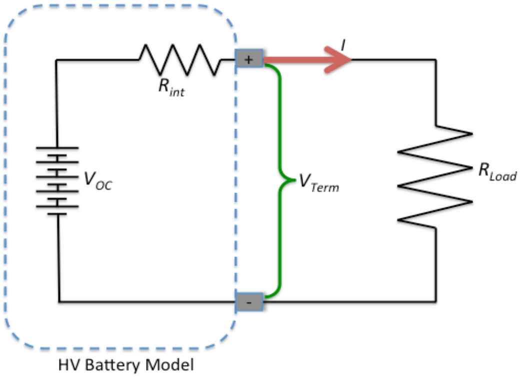


Figure 4-11: High voltage battery representation.

Despite its simplicity, the model shown in Figure 4-11 provides the necessary information and variables to run with the developed motor and HVSC models. The variables V_{OC}, R_{int}, I, and V_{Term} used in Figure 4-11 respectively represent the open circuit voltage (340 V), internal resistance (0.1 Ohm), current draw, and battery voltage at the terminals.

Calculating the power losses (P_{loss}) in the battery is possible by knowing the power output (P_{actual}) that the battery should provide to the motor, and the ideal power (P_{ideal}) that the battery could produce at open circuit voltage. The voltage at the battery terminals (V_{term}) can also be determined from knowing P_{actual}. Equation 4-8 through Equation 4-11 show the mathematical definitions.

$$P_{ideal} = P_{actual} + P_{loss} \qquad \text{Equation 4-8}$$

$$P_{ideal} = IV_{OC} \qquad \text{Equation 4-9}$$

$$P_{loss} = I^2 R_{int} \qquad \text{Equation 4-10}$$

$$P_{actual} = IV_{OC} - I^2 R_{int} \qquad \text{Equation 4-11(A)}$$

$$V_{term} = V_{OC} - IR_{int} \qquad \text{Equation 4-11(B)}$$

The battery current, I, is needed to track the SOC of the battery model. Solving the quadratic equation from the actual battery power, the current is given by Equation 4-12.

$$I = V_{OC} - \sqrt{\frac{V_{OC}^2 - 4R_{int}P_{actual}}{2R_{int}}} \qquad \text{Equation 4-12}$$

The SOC is calculated using Equation 4-13,

$$SOC_{new} = SOC_{old} + 100(\frac{dE_{int}}{E}) \qquad \text{Equation 4-13}$$

where *E* is the total internal energy capacity of the battery, and *∂E* is the change in *E* during a simulation time step.

Figure 4-12 shows the response of the current and voltage at the terminals of the battery model over the EcoCAR 3 cycles.

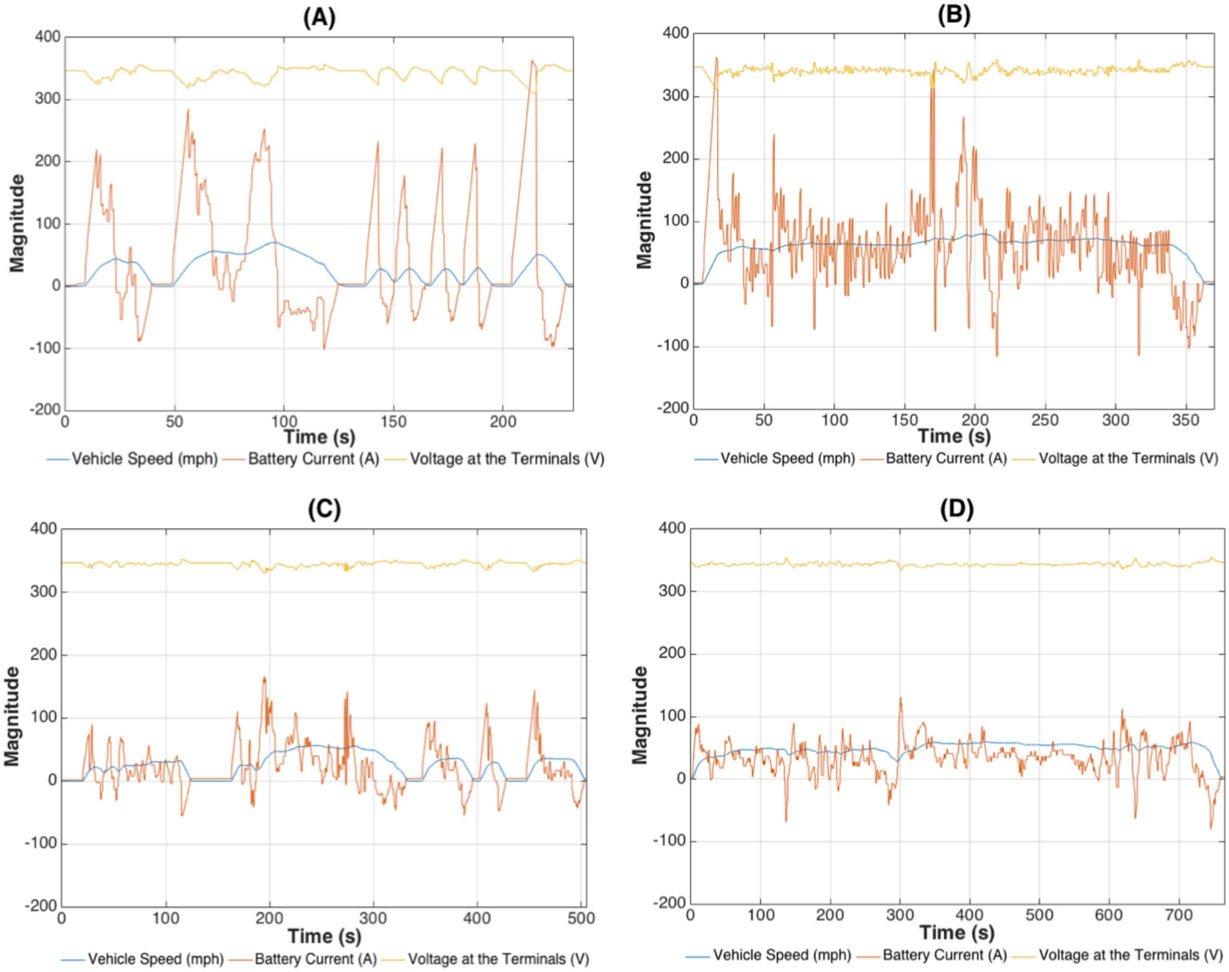


Figure 4-12: Battery voltage and current response over the EcoCAR 3 four cycles. (A) US06 City drive cycle. (B) US06 Highway drive cycle. (C) 505 drive cycle. (D) HWFET drive cycle.

4.3.5. <u>DC/DC Converter & High Voltage Battery Charger Plant Models</u>

The DC/DC converter steps high voltage to low voltage in the vehicle. The plant model for the DC/DC converter does not model the physics or details of stepping high voltage to low voltage. The HVBC in the vehicle transforms AC current from the wall into DC current that the battery can store. The plant model for the charger does not model the AC to DC conversion.

The models developed for these non-powertrain components determine the power losses from the known constant efficiency, $\eta_{constant}$, and power output, P_{out}, terms. Equation 4-14 shows the power loss equation. Equation 4-15 defines the actual efficiency that varies with load, and Equation 4-16 defines the power loss term from Equations 4-14 and 4-15.

$$P_{in} = P_{out} + P_{loss} \qquad \text{Equation 4-14}$$

$$\eta_{actual} = \frac{P_{out}}{P_{in}} \qquad \text{Equation 4-15}$$

$$P_{loss} = P_{out} \frac{(1-\eta_{constant})}{\eta_{constant}} \qquad \text{Equation 4-16}$$

Figure 4-13 shows the plant model structure implemented for the DC/DC and HVBC. Both components have a constant power loss term, *C*, associated with them. The power input, P_{in}, term is what the battery provides to the DC/DC converter, or what the wall provides to the HVBC.

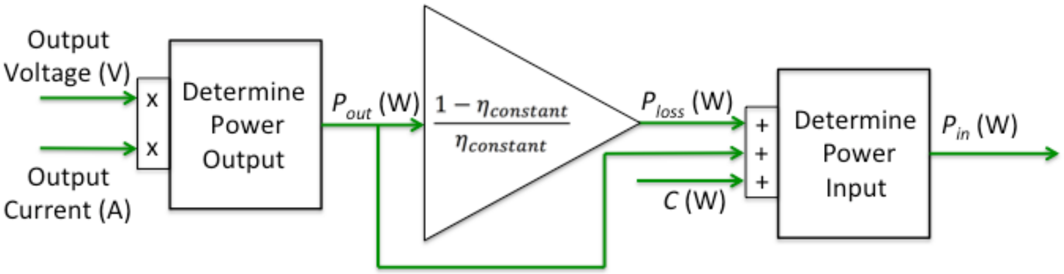


Figure 4-13: Plant model structure for DC/DC and HVBC.

Table 4-6 shows the constant efficiency, *C* power loss term, and operating voltage used for the model each component.

Table 4-6: Parameter values for DC/DC and HVBC plant models.

Parameter	DC/DC Converter	HVBC
Constant Efficiency (%)	95	92.3
Constant Power Loss Term, *C* (W)	25	90
Operating Voltage (V)	14	360

Figure 4-14 shows how the actual efficiency of both components varies at different power outputs.

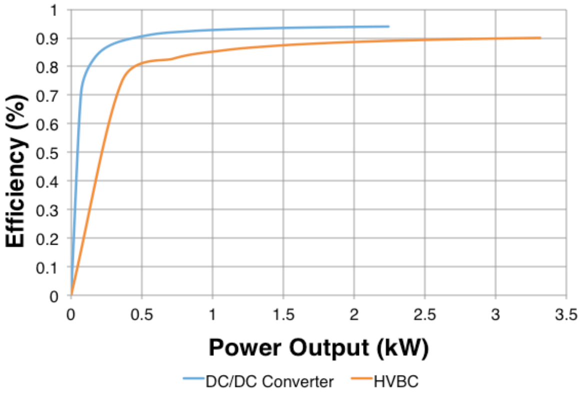


Figure 4-14: Actual efficiency for DC/DC converter and HVBC at different power outputs.

4.4. Vehicle Dynamics

The glider characteristics are modeled in the vehicle dynamics subsystem. Figure 4-15 shows the different forces that act on the vehicle model; opposing forces are shown in red, and the tractive effort needed to overcome the resistances is shown in green.

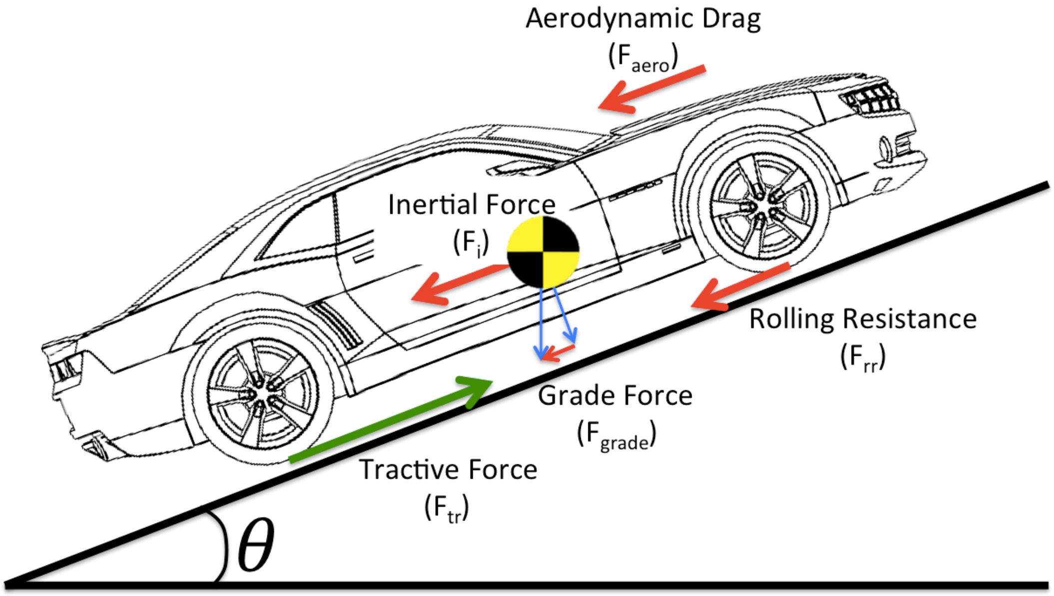


Figure 4-15: Vehicle glider model.

Equation 4-17 gives the mathematical definition of tractive force,

$$F_{tr} = F_{aero} + F_i + F_{grade} + F_{rr} \quad \text{Equation 4-17}$$

Where,

$$F_{aero} = \frac{1}{2}\rho C_d A_f V^2 \quad \text{Equation 4-18}$$

$$F_i = am_i \quad \text{Equation 4-18}$$

$$m_i = 1.04m \quad \text{Equation 4-19}$$

$$a = \frac{F_{tr}-(F_{aero}+F_{grade}+F_{rr})}{m_i} \quad \text{Equation 4-19}$$

$$F_{grade} = mg\ sin(\theta) \quad \text{Equation 4-20}$$

$$F_{rr} = mgC_{rr} \quad \text{Equation 4-21}$$

Table 4-7 provides the description and values of all parameters used to model the glider dynamics of the Camaro proposed by HEVT in EcoCAR 3.

Table 4-7: Vehicle glider parameters.

Parameter	Value	Units	Description	Is Constant
$\boldsymbol{\rho}$	1.23	kg/m^3	Air density	Yes
$\boldsymbol{C_d}$	0.38	-	Drag coefficient	Yes
$\boldsymbol{A_f}$	2	m^2	Vehicle frontal area	Yes
$\boldsymbol{V}$	Variable	m/s	Vehicle speed	Calculated in model
$\boldsymbol{a}$	Variable	m/s^2	Vehicle acceleration	Calculated in model
$\boldsymbol{m_i}$	2288	kg	Vehicle inertial mass	Yes
$\boldsymbol{m}$	2120	kg	Vehicle mass	Yes
$\boldsymbol{g}$	9.81	m/s^2	Gravity	Yes
$\boldsymbol{\theta}$*	0	$Degrees$	Road angle	Yes
$\boldsymbol{C_{rr}}$	0.01	-	Rolling resistance coefficient	Yes

* Can be set to any desired value. Zero for drive cycles.

The models developed during Year 1 and Year 2 of EcoCAR 3 are ready and available for SIL and HIL testing. Potential improvements and additions to the models include: adding a thermal system for the engine, adding a power electronics coolant loop, and adding details to SoftECU operation . Having models for the engine, transmission, motor, ESS, DC/DC converter, HVBC, and vehicle dynamics provides a strong foundation for modeling activities for Year 3 of EcoCAR 3.

5. Control System Development

Control of a hybrid electric vehicle system entails the development of functional and safety requirements, creation of a control algorithm to meet those requirements, and creation of an effective control architecture to run the proper hardware. Safety and functionality are the two most important criteria to develop a hybrid vehicle control system. The HVSC is the master controller that monitors and controls vehicle level operation. Understanding the actions and tasks that the master controller needs to perform is critical to write the control algorithm for the vehicle that is deployed onto the HVSC. Control requirements are the first step in the V-cycle proposed in Figure 4-1. The vehicle level requirements must first be defined and then broken down into subsystem level, and component level requirements. All requirements are developed towards meeting the initial design goals and VTS. All component and subsystem interactions relevant for vehicle operation must be accounted for before developing requirements. Accounting for component interactions ensures that there is an understanding of how the systems work together and what the expected behavior should be. Figure 5-1 shows the CAN, analog, and physical interactions that are expected to occur in the HEVT Camaro. The interactions show are based on the operation of components and the parallel hybrid powertrain. The driver physically provides the inputs necessary to initiate vehicle operation; those inputs are received and interpreted by the HVSC to command the other controllers via CAN. Consequently, each control module sends the proper commands to its specific component while receiving sensor feedback. Finally, the components provide the necessary functionality to make the vehicle do what the driver expects. In EV operation, for instance, the driver presses the brake, selects Park, and presses and ignition button; those actions are read by the HVSC and trigger the closing of the contactors in the ESS. The high voltage bus becomes energized with contactors closed, and the DC/DC converter begins operation to support 12V systems. When the vehicle is ready to produce torque, the “powertrain ready” light informs the driver, and if requested, the P3 motor can receive electric power to produce torque that is sent to the wheels.

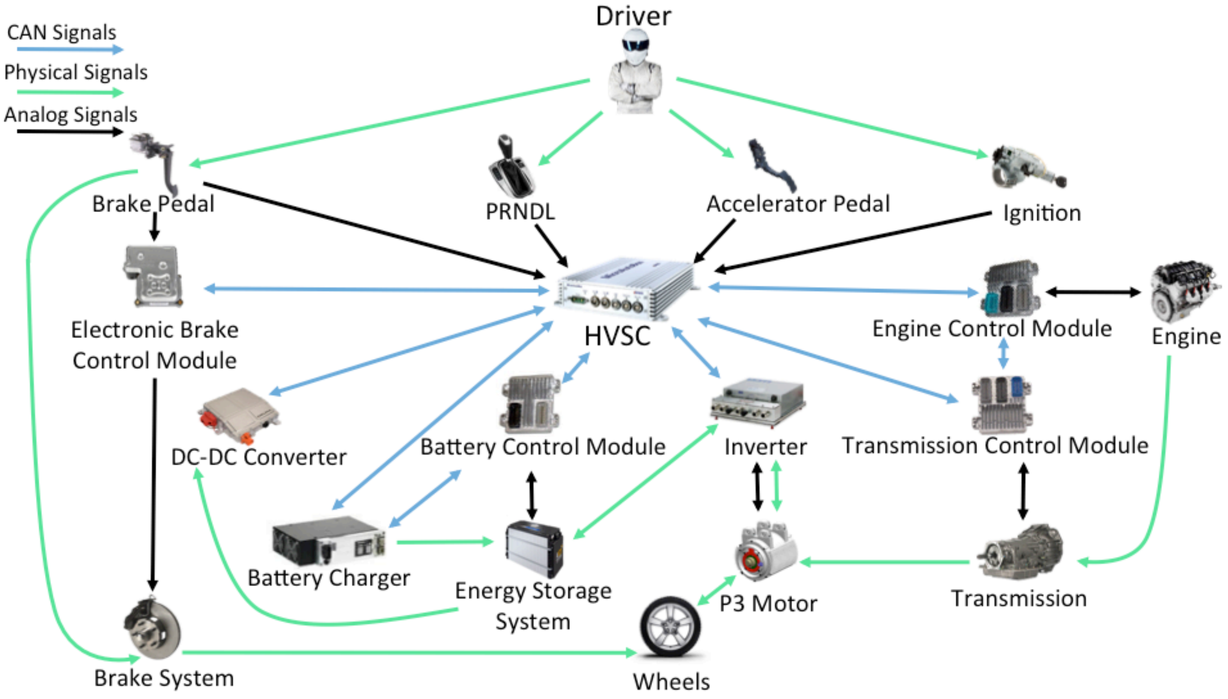


Figure 5-1: Component interaction for HEVT Camaro.

Following the development of control requirements, a control algorithm must be developed to accomplish safe hybrid vehicle operation. The algorithm developed in this study is structured according to functionality of different subsystems in the code. The three main functions of the control code are diagnostics of the operating state of components, selection of a operating mode, and execution of specific commands to component controllers. Another important piece of the control system for a hybrid electric vehicle is the architecture used for communication between ECUs. The CAN protocol is the communication standard in the automotive industry. The CAN architecture defined by HEVT separates ECUs by functionality and also by communication speed. ECUs are grouped in different CAN busses, and the HVSC has the ability to connect to several busses to introduce the commands to each component. Details about the proposed CAN network are provided in Section 5.6.

5.1. Control Requirements

The control requirements for the HEVT hybrid powertrain are developed at three different operational levels: vehicle level, subsystem level, and component level. A multilevel structure is implemented to better mimic the vehicle development process implemented by General Motors. The vehicle level requirements are general and are written based on how the driver interacts with the car as a system. The vehicle is made of subsystems with specific functionality, thus the subsystem requirements are defined to evaluate components working together and to ensure that each one performs its tasks while meeting driver demand. Finally, component levels are defined to evaluate different characteristics of the components, like communication, startup, shutdown, normal operation and more. The significance of a multilevel structure is that each component requirement can be linked back to the levels above (subsystem and vehicle). Having a sound structure for control requirements facilitates and guides the development of the control algorithm. Figure 5-2 displays the multilevel structure implemented. At the highest level, the identifier is just an integer number. Then each subsystem in the vehicle is identified with a roman numeral and a letter is assigned to each requirement at this level. Finally, each individual component, operational category, and component requirement are identified with integer numbers separated by dots.

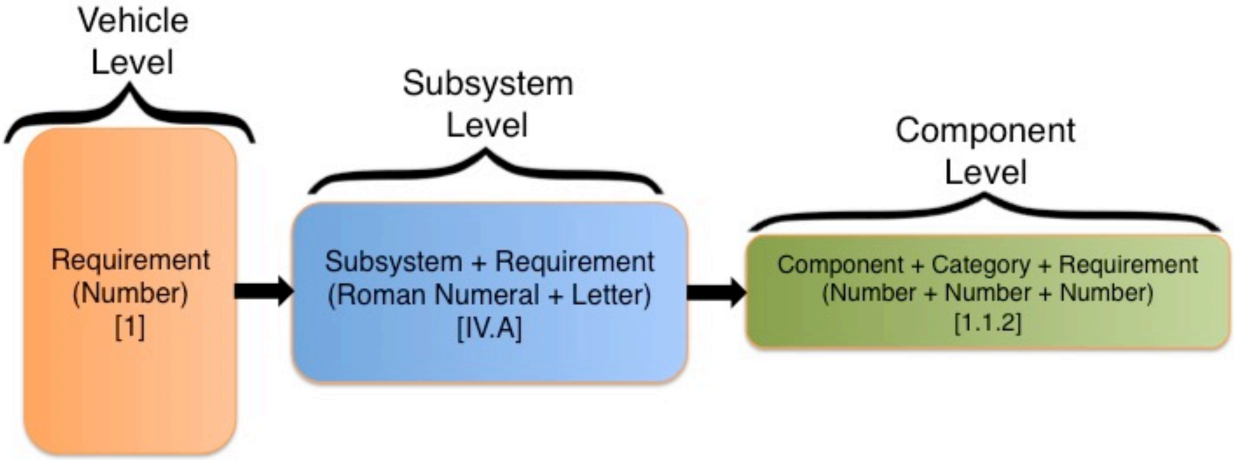


Figure 5-2: Multilevel structure of control requirements.

Table 5-1 lists a sample of five vehicle requirements that encompass vehicle startup operation, brake pedal, PRNDL, EDS and fault response. Table 5-2 shows a sample of requirements developed for the electric traction subsystem; these requirements are identified as shown in Figure 5-2 and are linked to items at the vehicle level. The ESS, P3 motor, inverter, high voltage bus, brake pedal, accelerator pedal, and brakes, are components in the electric traction subsystem. Finally,

Table 5-3 identifies a sample of control requirements for the P3 motor and links them back to the subsystem and vehicle levels. Item identification at the component level is given by the component number, the operational category, and the requirement ID. Every item at the component level can be linked to one or more requirements at the subsystem and vehicle levels. The linking to more than one higher requirement occurs because components are often part of more than one subsystem, and some subsystems may work together to meet a vehicle level requirement. Control algorithm development starts when an extensive and logical list of component requirements is compiled. Then, after validating that items at the component level are met, the validation propagates to the upper levels. Appendix A contains the full list of control requirements for vehicle, subsystem, and component level operation.

Table 5-1: Sample set of control requirements at the vehicle level.

Number	Vehicle Requirements
1	Vehicle shall be drivable within 10 seconds of startup
2	Vehicle shall not move when PRNDL is set to P
3	Vehicle shall slow down and stop when the brake pedal is pressed
4	Vehicle shall allow forward motion when PRNDL is set to Drive or Manual and brake pedal is not pressed
5	Vehicle shall stop operation when EDS switch is pressed or faults are present

Table 5-2: Sample set of control requirements for the electric traction subsystem.

Subsystem	Vehicle Requirement Link	Requirement ID	Subsystem Requirement
Electric Traction [II]	1	II.A	ESS shall close contactors when driver enables the vehicle
	1	II.B	Inverter shall control the motor while the high voltage bus is enabled
	2	II.C	Inverter shall command zero motor torque when PRNDL is set to P
	2	II.D	Wheels shall not receive torque from any component when PRNDL is set to P
	3	II.E	ESS shall receive power from motor during regenerative braking
	3	II.F	Brakes shall be applied when the brake pedal is pressed
	4	II.G	ESS shall provide power to the inverter to power the traction motor
	4	II.H	Inverter shall command motor torque
	4	II.I	Motor shall provide torque to the wheels to move in the direction the driver desires
	5	II.J	ESS shall open contactors if emergency stop switch is pressed
	5	II.K	High voltage bus shall be disabled if emergency stop switch is pressed

Table 5-3: Sample set of control requirements for the P3 motor.

Component	Category	Requirement ID	Component Requirement	Subsystem Requirement Link	Vehicle Requirement Link
3. Motor	1. Starting	3.1.1	The HVSC shall enable the power relay to provide 12V power to the inverter	II.B	1
		3.1.2	The HVSC shall verify that the inverter is sending CAN messages	II.B	1
		3.1.3	The HVSC shall start sending the CAN command message with the Inverter_Enable signal equal to 0	II.B	1
		3.1.4	The HVSC shall verify that there are no faults being transmitted by the inverter, if there are, they must be identified, resolved, and cleared before proceeding	II.B	1, 5
		3.1.5	The HVSC shall enable the inverter through both the CAN command message and the digital enable signal	II.B	1
	2. Communication Input	3.2.1	The inverter shall receive a torque request from the HVSC	II.B , II.H	1,3,4
		3.2.2	The inverter shall receive a motor speed signal from motor speed sensor	II.B	1,3,4
	3. Communication Output	3.3.1	The inverter shall draw a current from the ESS	II.G , II.B	1, 3, 4
		3.3.2	The inverter shall output a temperature signal from the inverter temperature sensor	II.B	1
		3.3.3	The inverter shall output a fault signal to the HVSC if any motor constraints are not met	II.B	5
	4. Constraint	3.4.1	The motor shall not exceed the maximum rated torque	II.B, II.H, II.I	5
		3.4.2	The motor shall not exceed the maximum rated speed	II.B, II.H, II.I	5
		3.4.3	The motor shall not exceed the maximum rated temperature	II.B, II.H	5
		3.4.4	The motor shall decay from peak torque to continuous torque based on thermal limits	II.B, II.H, II.I	5

Component	Category	Requirement ID	Component Requirement	Subsystem Requirement Link	Vehicle Requirement Link
		3.4.5	The motor shall not exceed the torque rate limit	II.B, II.H, II.I	5
		3.4.6	The motor shall not exceed the speed rate limit	II.B, II.H, II.I	5
	5. Shutdown	3.5.1	The HVSC shall command a zero speed to the inverter through the CAN command message	II.B, II.H	2
		3.5.2	The HVSC shall disable the inverter through both the CAN command message and the digital enable signal	II.B	2
		3.5.3	The HVSC shall disable the power relay that provides 12V power to the inverter	II.B	2

Requirements are developed for all identified subsystems and components. A total of 16 vehicle control requirements are generated. Identified subsystems include (not limited to): ignition, electric traction, conventional traction, safety (emergency disconnect), low voltage, braking, cooling, and charging systems. Similarly, components for which requirements were generated include: engine, transmission, inverter, motor, ESS, DC/DC converter, HVBC, and HVSC.

5.2. Algorithm Structure

The HVSC control code consists of three major blocks: The diagnostics block, the selection block, and the execution block. Signals from the vehicle I/O are fed into each of the three blocks. Signals from the diagnostic block are fed to the selection and execution blocks, and signals from the selection block are fed into the execution block. Figure 5-3 shows the structure of the HEVT control algorithm.

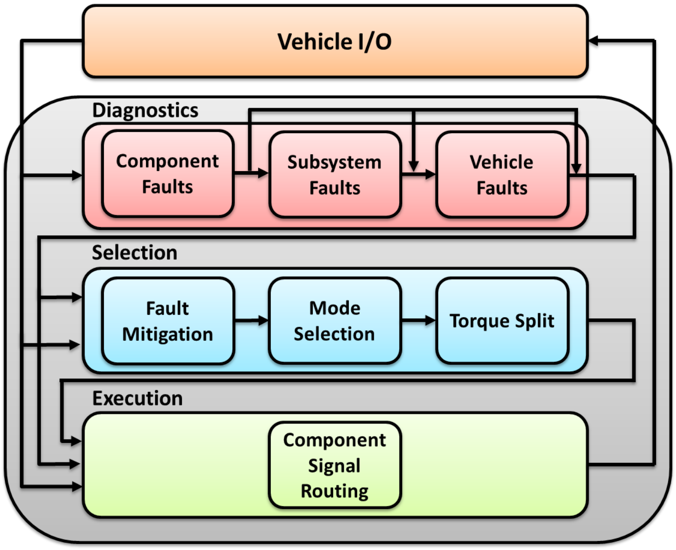


Figure 5-3: Control code structure.

5.3. DFMEA & Diagnostics

The diagnostic block in the HVSC is used to collect and generate information about the status of the vehicle. One major function of the diagnostic block is to detect faults, which is accomplished by receiving signals from components and performing logic to determine if the received signals indicate that a specific component can operate safely. The diagnostics block can also receive fault signals directly from components. The component information passed into the diagnostic block is also used to perform status checks at the vehicle and subsystem level. Once the fault signals are generated and collected in the HVSC, they are passed to the Selection block, which chooses what action should be taken based on the diagnostics.

There are several types of faults detected in the diagnostic block, including temperature, voltage, current, and other general faults for each component. Two prominent faults identified as high priority through DFMEA and incorporated into the model are: faulty accelerator pedal signal and loss of CAN communication with powertrain components and ESS. Table 5-4 shows the ten most relevant failure modes identified through DFMEA. The RPN is calculated as the product of severity, occurrence and detection.

Table 5-4: Top 10 DFMEA failure modes.

Component	Potential Failure Mode	Failure Effect	Severity	Occurrence	Detection	RPN
Accelerator Pedal	Inaccurate voltage signals	Unintended acceleration	10	8	9	720
Hybrid CAN Bus	Excess noise in CAN bus	Inability to control hybrid components	9	9	8	648
Inverter	Loss of CAN communication	Inability to control and communicate with the inverter	9	8	8	576
Brake Pedal	Inaccurate voltage signal	Inability to brake, unintended deceleration	9	7	9	567
ESS	Loss of CAN communication	Inability to control and communicate with the ESS	8	8	8	512
ESS	Loss of internal CAN communication	Inability to control contactor status	8	7	8	448
Transmission	Internal Mode Switch voltage out of range	Inability to force Neutral in EV mode	7	9	7	441
Brakes	Brake Pressure Loss	Loss of Braking	10	5	8	400
DC/DC Converter	Loss of CAN communication	Inability to charge 12V system in accessory or EV mode	6	7	9	378
ESS	Inadequate isolation of high voltage system	Ground fault triggered by EsCM	6	8	7	336

Fault conditions are identified by implementing logic operations that determine whether a component is exceeding its operating limits or if a required input signal to the component is missing. In the case of a CAN communication fault, the rate at which controllers communicate is known; if a CAN message fails to be received by the HVSC within a window of 5x the transmit rate, then a communication fault is indicated. In the case of the accelerator pedal signal, the code performs a plausibility check for both APP signals to ensure that the ratio between sensor voltages is equal to 0.5. If this ratio is not within the expected range, a fault is reported because unintended acceleration can occur. In the current HVSC code, if an accelerator pedal fault occurs, then a zero torque command in sent to the motor and engine. Figure 5-4 shows the results of a fault insertion case with intermittent APP loss. The hybrid powertrain reacts as desired (coasts down to zero speed) based on fault mitigation actions derived with DFMEA and control requirements.

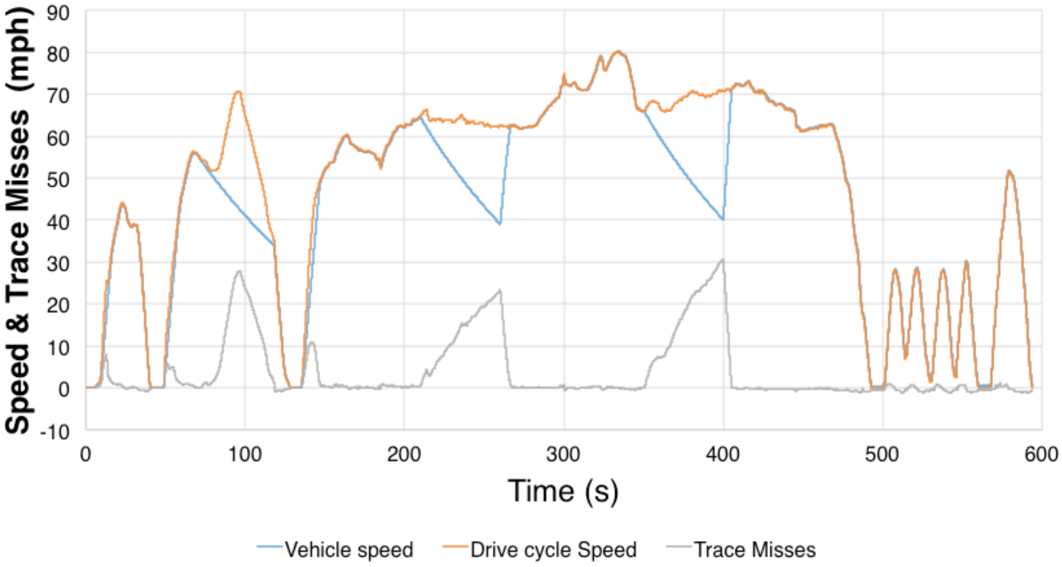


Figure 5-4: APP fault insertion result.

5.4. Mode Selection

The second major component of the HVSC is the selection block. This subsystem receives inputs from both the diagnostic block and vehicle I/O. The controller block in the HVSC code encompasses two major functions: choosing a course of action based on fault signals from the diagnostics block and determining the optimal torque split between the engine and the motor. The selection block does not detect faults, but rather receives fault detection information from the diagnostic block and uses logical operators and calculations to decide what actions need to be taken to ensure safe vehicle operation. Actions taken in response to a fault signal range in intensity. Some faults will require vehicle operation to be suspended, some will inhibit a specific mode of operation, and some will simply result in the display of an indicator light on the user interface. In the HVSC algorithm, vehicle operation is stopped when an accelerator pedal fault occurs. However, in the case of ground fault detection, only a warning is generated depending on the value of the resistance drop between the high voltage bus and chassis ground. This warning is triggered by interval tests that check if the resistance drop is within the specified limits.

The second major function of the selection block in the HVSC is to choose the mode of operation. The logic used to determine the mode of operation occurs in the Stateflow environment within Simulink. First, this logic structure determines whether or not the diagnostics block has sent signals to either force only the engine to run or to force the vehicle into CS mode; these are the highest priority checks, because they reduce the modes in which the vehicle can operate. If neither of these checks are active, the selection block determines the operating mode of the vehicle. Upon initial discharge of the ESS, the vehicle operates in CD mode. CS operation starts after a minimum SOC threshold is reached (30%). In CS mode, the selection block considers the driver demand and current operating status of the vehicle to calculate a desired torque split between the engine and motor. There are two separate torque calculations within the selection block: One determines the torque required at the wheels based on driver demand and vehicle speed. The other calculation determines the torque split ratio between the motor and the engine to meet the torque requirement at the wheels. These calculations are complex because it is necessary to find a torque split that can operate the powertrain at high efficiency, and there are many variables involved. The priority for EcoCAR 3 in Year 2 is safety and functionality of the control system, thus the developed control strategy does not optimize powertrain or component efficiency. Instead, a preliminary Fuzzy control strategy is developed to ensure functionality without a negative impact on energy consumption and performance VTS.

5.4.1. <u>Charge Depleting Strategy with Fuzzy Logic</u>

A Fuzzy control logic is developed and implemented for CD operation. In this mode, only the P3 motor provides torque to the wheels. Figure 5-5 shows the interaction between hybrid components during CD operation in the HEVT Camaro. The CD strategy developed is not a blended strategy. Instead, the hybrid powertrain operates as an EV when the SOC is between 98% and 30%. Below 30% SOC, the vehicle operates in CS mode. Details of CS operation are covered in the next subsection.

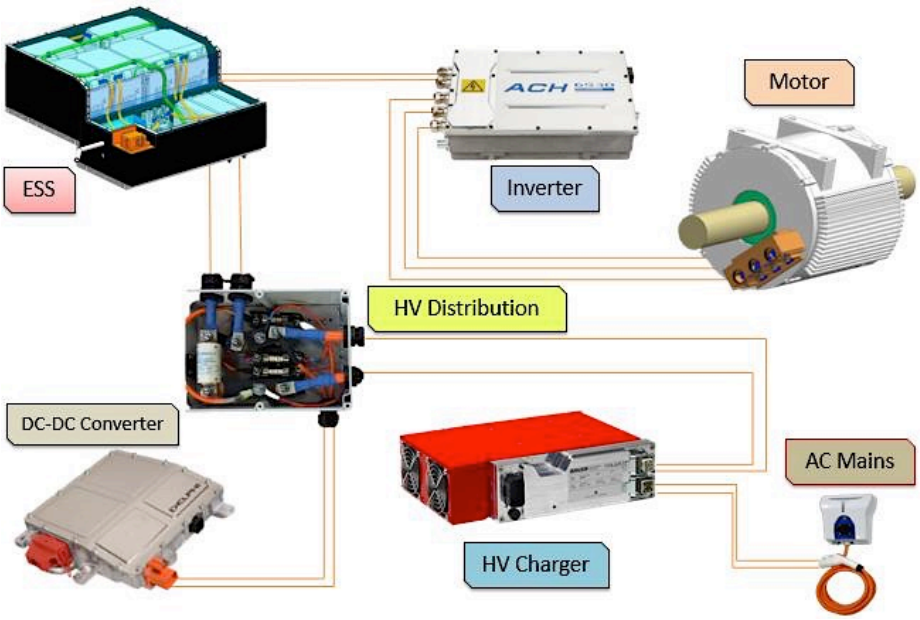


Figure 5-5: Component interaction and energy flow in CD operation.

The two input variables of the CD Fuzzy logic are driver demand (in the form accelerator pedal position) and vehicle speed. The output torque of the electric motor is the output variable of the

Fuzzy inference system. A surface map of motor torque is generated by creating the membership functions for each one of the defined input and output variables. The motor torque map resulting from the Fuzzy rule set is shown in Figure 5-6.

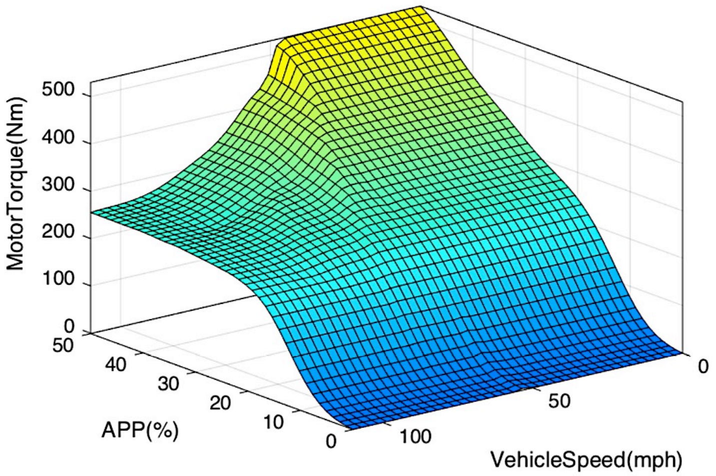


Figure 5-6: Motor torque map as function of accelerator pedal position and vehicle speed.

The driver demand in Figure 5-6 only goes to 50% accelerator pedal position. This pedal calibration is done for drivability because the powertrain can still add engine and transmission torque to motor torque. Thus a pedal position greater than 50% will be used to factor in engine power in charge sustaining operation. The motor torque command for regen braking operation is not processed by the Fuzzy logic; instead, the braking command is processed in the HVSC based on vehicle speed, BPP, and regen torque limits for both CD and CS operation. Table 5-5 shows the membership functions used for input and output parameters in the CD Fuzzy inference system. The CD membership functions plots are shown in Figure 5-7. The motor torque membership functions in Figure 5-7(C) are evaluated from -100 Nm to 650 Nm for visualization purposes (ease of manipulation in MATLAB Fuzzy logic toolbox). However, the propel torque that the motor can produce is limited by the control algorithm from 0 Nm to 500 Nm up to the base speed and limited by constant power past the base speed (1900 RPM). A variety of function types are used to obtain the desired output torque with the CD Fuzzy inference system.

Table 5-5: Membership functions for each variable in CD Fuzzy inference system.

APP (%)	Vehicle Speed (mph)	Motor Torque (Nm)
Passive	City	Continuous Low
Moderate	Mid-range	Continuous High
Aggressive	Highway	Peak Low
-	-	Peak High

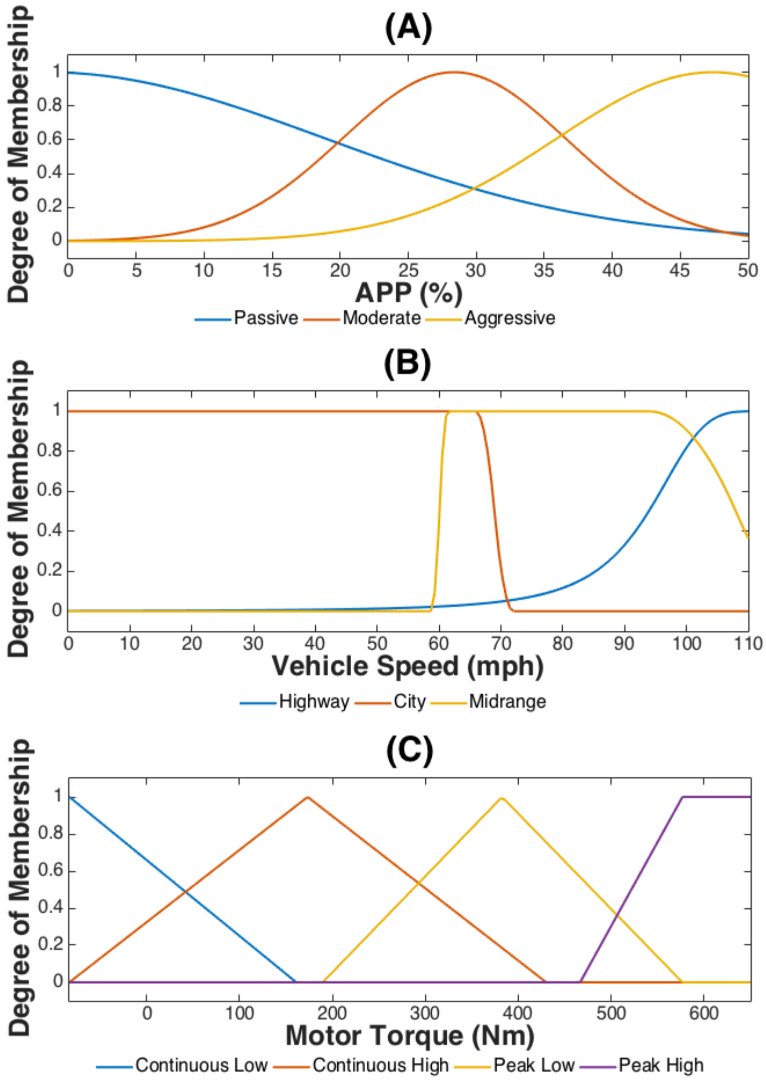


Figure 5-7: Membership functions for input and output variables of CD Fuzzy inference system.

Table 5-6 contains the set of Fuzzy rules developed to create the motor torque map in Figure 5-6.

Table 5-6: Fuzzy rule set for CD operation.

IF APP	AND Vehicle Speed	THEN Motor Torque
Passive	City	Continuous Low
Moderate	City	Peak Low
Aggressive	City	Peak High
Passive	Mid-range	Continuous Low
Moderate	Mid-range	Peak Low
Aggressive	Mid-range	Peak Low
Passive	Highway	Continuous Low
Moderate	Highway	Continuous High
Aggressive	Highway	Continuous High

5.4.2. *Charge Sustaining Strategy with Fuzzy Logic*

The HEVT Camaro operates as an EV in charge depleting mode. CS sustaining operation starts when the SOC reaches a threshold of 20%. The SOC then oscillates between 20% and 30% while meeting driver demand and operating the powertrain within limits. The control logic developed for CS mode is a preliminary Fuzzy inference system that seeks to be safe and functional initially. Year 3 and Year 4 of EcoCAR 3 can focus on the improvement of the developed Fuzzy logic for CS operation to optimize system efficiency while meeting driver demand. Before defining the variables and membership functions that the Fuzzy inference system uses, an understanding of powertrain operation and desired torque split is necessary. Figure 5-8 shows the interactions that the torque producing components have with the HVSC during CS operation. The APP read by the HVSC is converted into a torque requests for the motor and the engine. The HVSC is programmed to run calculations and generate commands in terms of driveshaft torque. Driveshaft torque means a direct command to the motor, and only the torque multiplication in the transmission has to be accounted for the engine command. Since there are no torque sensors in the vehicle, the HVSC relies on reported torque estimations from each controller. In addition, the HVSC performs torque diagnostics based on vehicle acceleration information from sensors, and from glider calculation in the vehicle model.

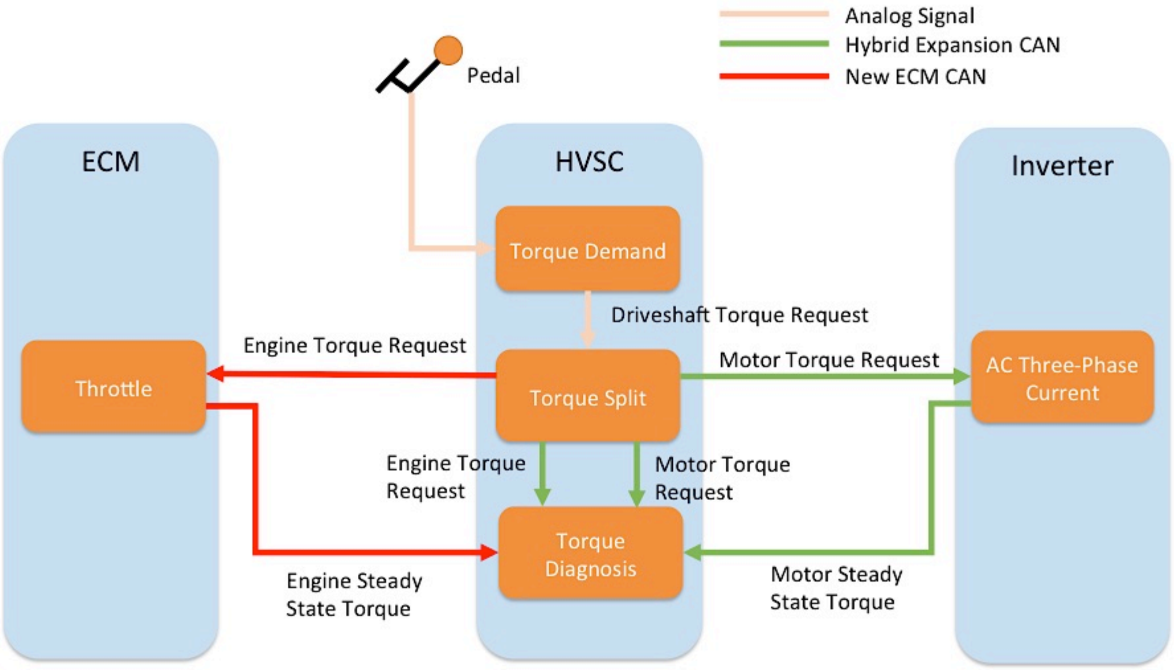


Figure 5-8: Interaction of torque producing components with HVSC.

In Year 1 of EcoCAR 3, the torque split curves in Figure 5-9 were proposed as an initial estimation of what the hybrid powertrain should do. The shape and characteristics of these curves are based on knowledge of the component limitations (engine and motor power) and on the desired powertrain behavior in different driving conditions. Although the initially developed curves represent an estimation of the desired powertrain behavior, they play an important role in the development of a functional torque split. These curves guided the development of rules for the Fuzzy inference system used for CS operation.

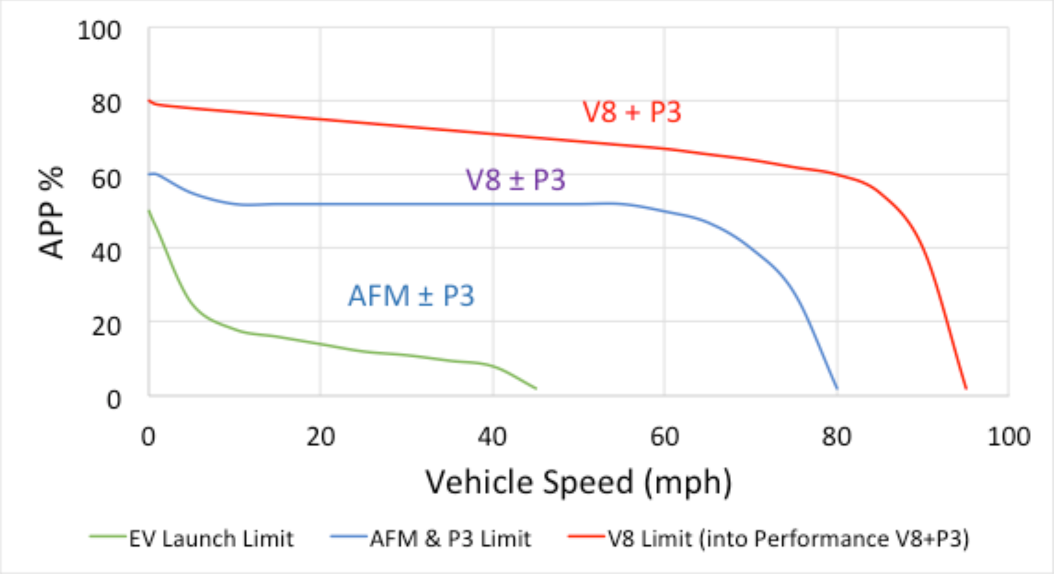


Figure 5-9: Initially proposed torque split between V8 engine with AFM and P3 motor as function of APP and vehicle speed.

After refining the preliminary control strategy (current Fuzzy logic), the desired final torque split is as follows: at low speeds and low driver demand, the motor is expected to provide torque to the wheels to keep the engine from operating at poor efficiency (low load driving). At low and medium driver demand, the engine can be operated at the right conditions for AFM while the motor can load or assist to augment AFM operation and reduce fuel consumption. During engine loading operation, the motor absorbs driveshaft torque from the transmission to charge the battery. The current Fuzzy logic does not seek to operate the engine at peak efficiency or to augment time spent in AFM mode. Finally at high speeds, the V8 engine can be loaded or assisted by the P3 motor to meet high driver demand. In addition to APP and vehicle speed, the initial Fuzzy control strategy takes SOC, SOC direction in the CS band (value of 0 when going from 20% to 30%; 1 when going from 30% to 20%) as inputs. The output variable is a unitless number between 0 and 1.5 that determines the engine and motor torque split. Table 5-7 shows all the membership functions for the four inputs and the one output variable.

Table 5-7: Membership functions for each variable in CS Fuzzy inference system.

APP (%)	Vehicle Speed (mph)	SOC (%)	SOC Direction (0l1)	Output Torque Split (0l1.5)
Very Small	City Slow	Low	Bottom Top	All Electric
Small	City Fast	Window	Top Bottom	Large Assist
Medium	Mid Range	High	-	Medium Assist
Large	Hwy Slow	-	-	Small Assist
Very Large	Hwy Fast	-	-	Little Load
-	-	-	-	Mid Load
-	-	-	-	Large Load

Figure 5-10 shows all membership functions used for the input variables of the CS Fuzzy inference system. Similarly, Figure 5-11 shows the membership functions used for the output torque split for the CS Fuzzy logic. Appendix B lists the full set of Fuzzy rules used for CS mode.

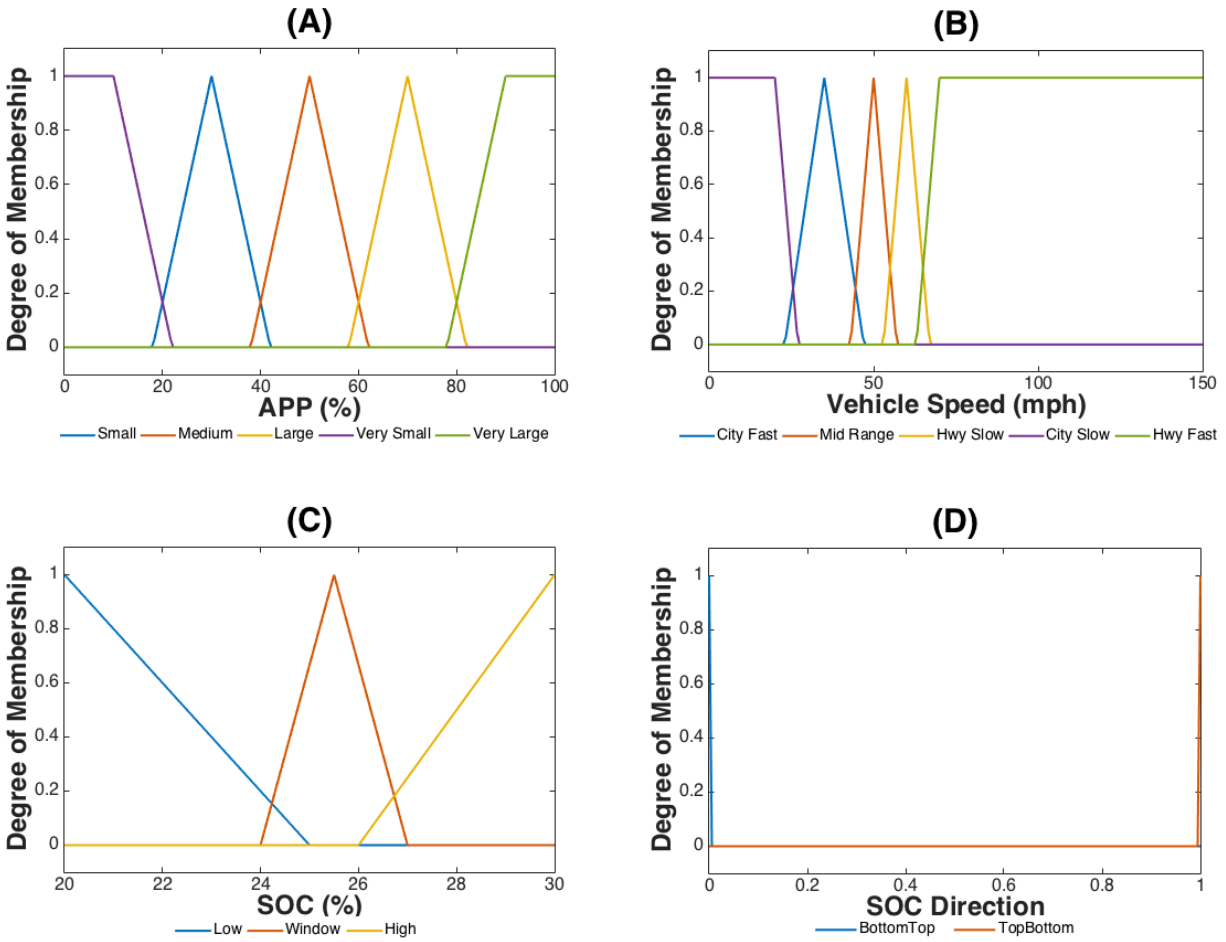


Figure 5-10: Membership functions for input variables of CS Fuzzy inference system. (A) Membership functions for APP. (B) Membership functions for vehicle speed. (C) Membership functions for SOC. (D) Membership functions for SOC Direction.

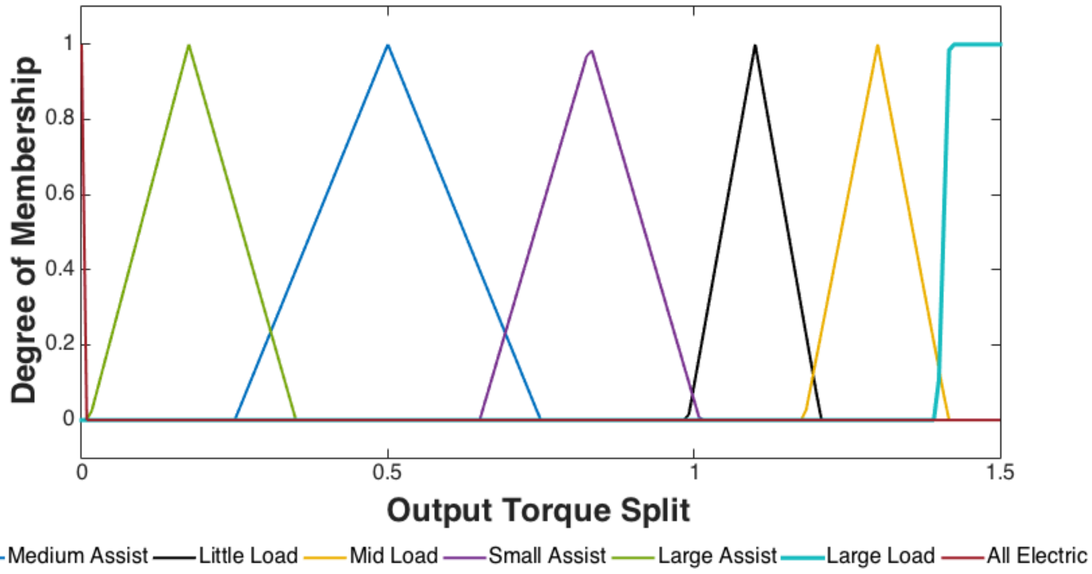


Figure 5-11: Membership functions for output variable of CS Fuzzy inference system.

The output of the Fuzzy logic is centered at 1; zero means that only the motor produces torque, 1 means only the engine produces torque, and 1.5 means that the engine produces torque and the motor loads the engine. Output numbers between zero and 1 indicate that the motor assists the engine torque, and numbers between 1 and 1.5 means that the motor loads the engine. Figure 5-12 presents the operations performed with the output of the Fuzzy control strategy to determine the values of motor and engine torque.

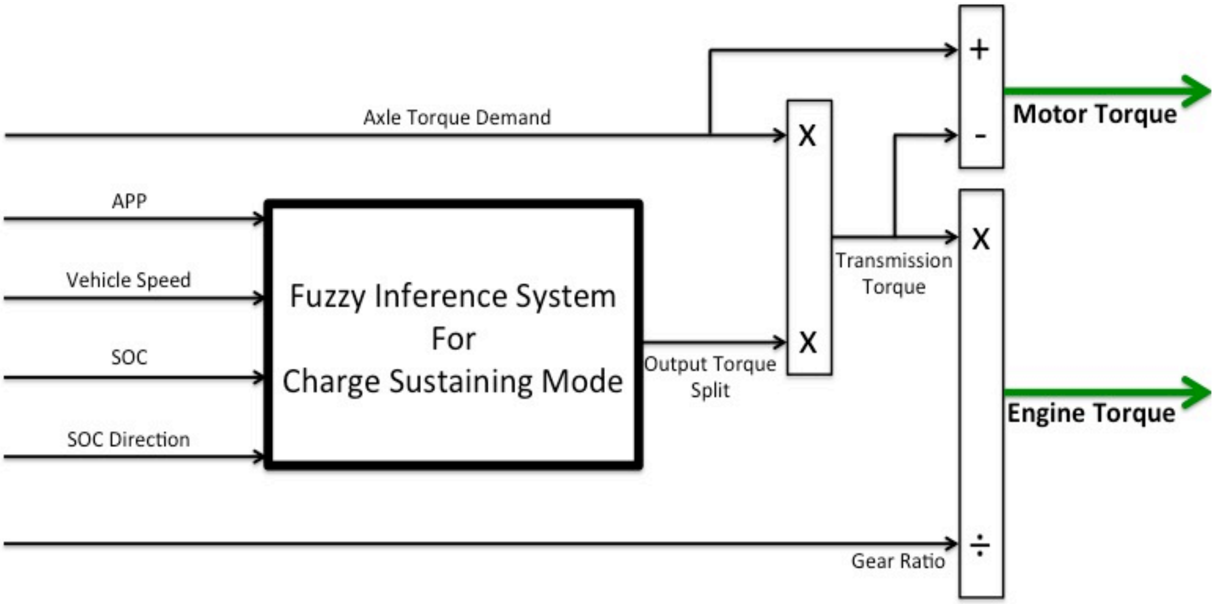


Figure 5-12: Diagram of torque split operation.

The CS Fuzzy logic determines the proportions of the torque split to meet driver demand. These proportions are determined by the 150 rules in the inference system based on the values of the input variables. The maximum loading torque that the motor can apply is given by an output torque split number of 1.5. This number can be tuned in the future to yield a more desirable response if changes are made to the model or control code. The value of 1.5 is determined from the maximum possible torque that the transmission can provide at any given gear. Transmission output torque is relevant because it determines the motor torque command. In the case of maximum loading operation, the motor command can be up to 50% (from 1.5 minus 1, See Figure 5-12) the magnitude of the transmission output torque at WOT. For instance, assuming that the output of the Fuzzy inference system is 1.5, then Figure 5-13 indicates that in second gear, the motor loading command could go up to -750 Nm. This loading torque is outside of the capabilities of the P3 motor; therefore it is necessary to implement robust limitations to avoid commanding such high torques. For simplicity in the initial control strategy, the limitations implemented in the control code keep the loading and regen torques under one half (250 Nm) of the peak torque of the motor (500 Nm).

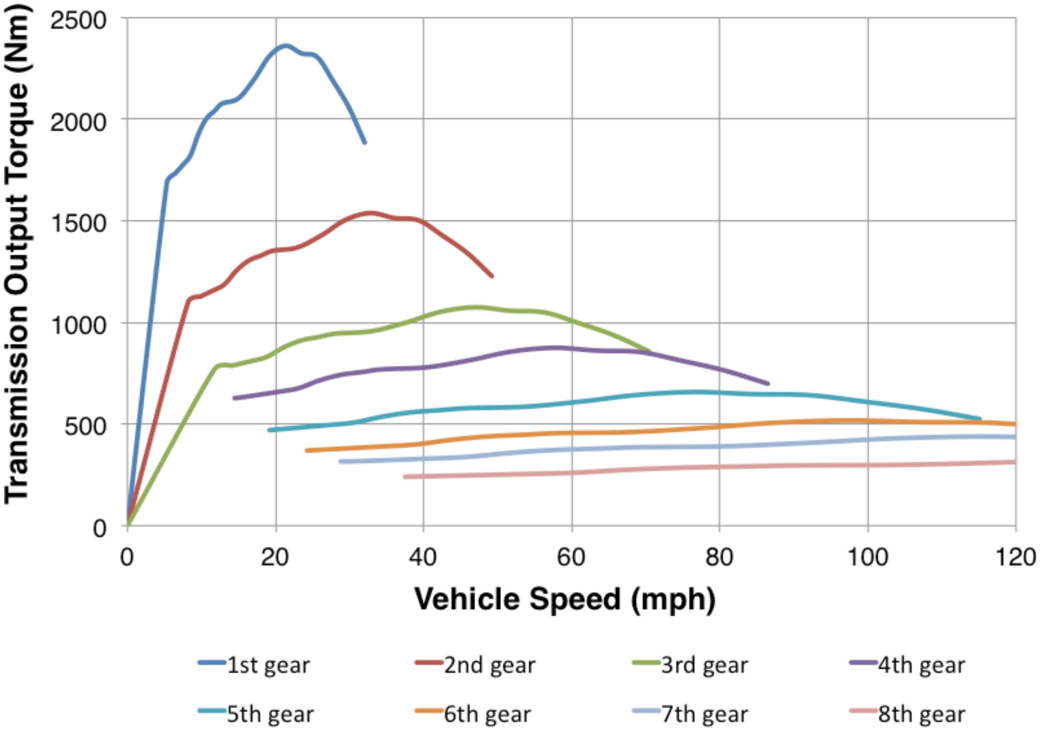


Figure 5-13: 8L90 Transmission output torque curves (without transmission losses) when engine operates at WOT.

The engine loading that the motor does at higher gears (4th through 8th) is mild compared to that at lower gears due to the operating principle of the CS strategy. Figure 5-14 shows how the model SOC fluctuates in CD and CS operation over eight US06 City cycles. The starting SOC for the test is 40%. The model response shows that the ESS is depleted much faster than it is charged, which can be attributed to the mild charging that occurs past 3rd gear. The SOC remains within the desired window from 20% to 30% when the vehicle operates in CS mode.

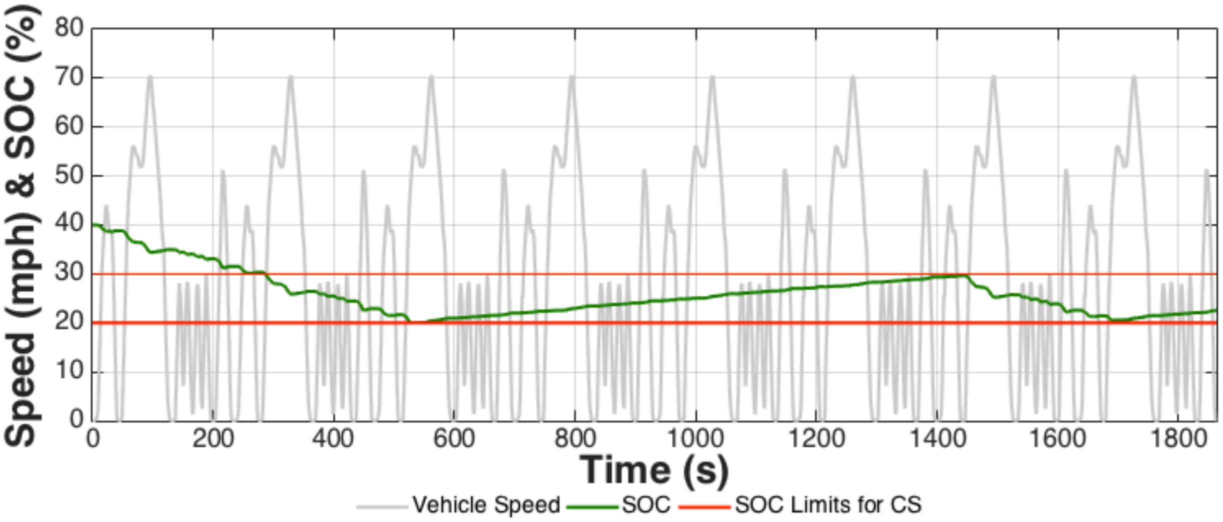


Figure 5-14: SOC response during CD and CS operation over several US06 City cycles.

Figure 5-15 shows the behavior of the control code and model when tested over a single US06 City cycle. The initial state of charge for the sample test is 20% and the final state of charge is 23.5%. The control code enters CS operation as soon as the state of charge falls below 20%. The Fuzzy logic output oscillates between 0 and 1.5 as expected. During operation in low transmission gears, the motor puts the highest load on the engine. Finally the engine is able to work with the motor as a system to meet driver demand and maintain SOC. Efficiency considerations are to be incorporated in the future.

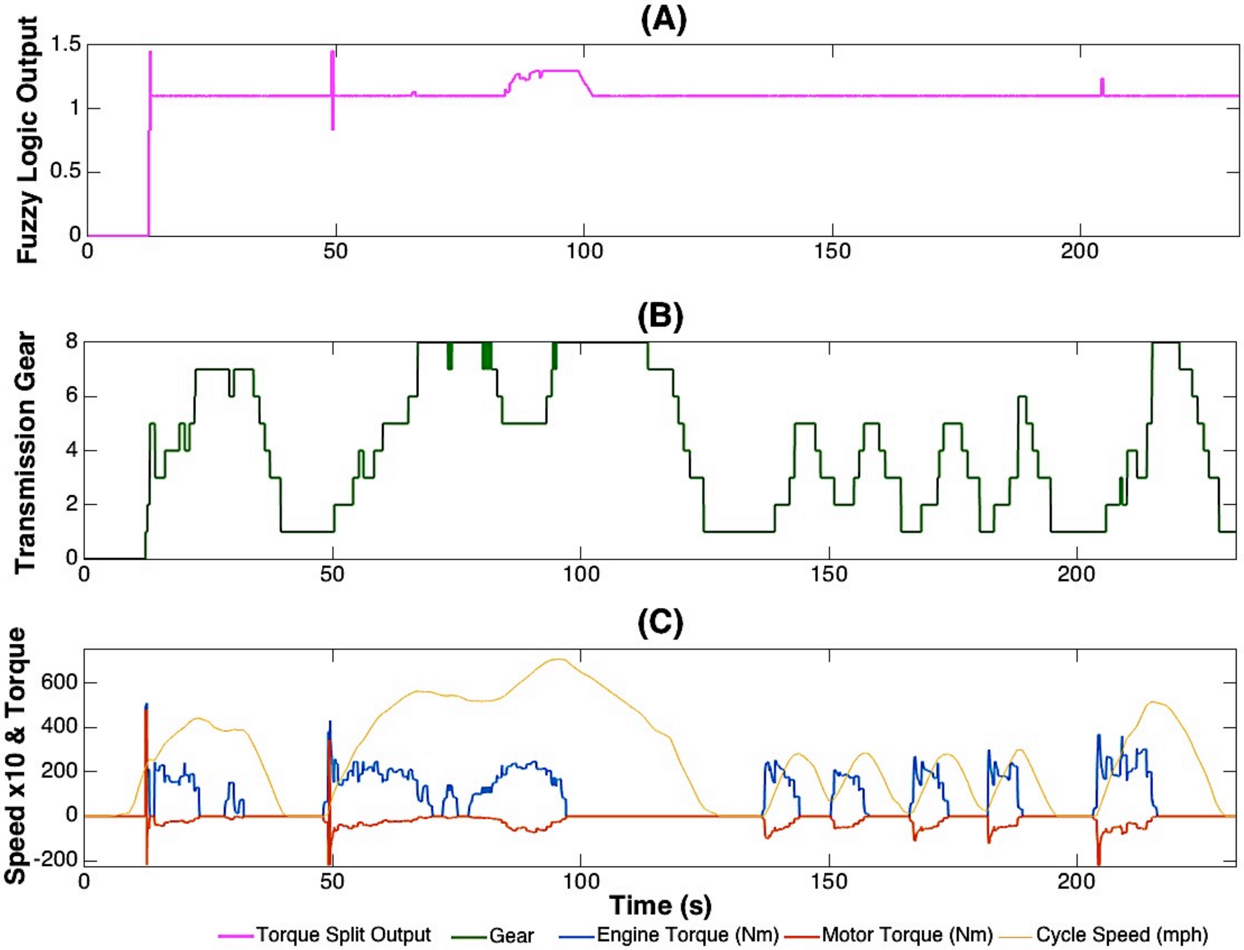


Figure 5-15: CS operation with Fuzzy control logic over US06 City cycle. (A) Fuzzy logic output. (B) Transmission gear. (C) Engine and motor torque with drive cycle speed trace.

The Fuzzy logic strategy yields acceptable behavior during drive cycle testing. The control strategy is also tested to generate performance results for 0 – 60 mph and 50 – 70 mph accelerations. Figure 5-16(A) shows that the 0 – 60 mph acceleration time is approximately 4.8 seconds with a limitation in tractive effort enforced to ensure realistic results based on the power to weigh ratio of the vehicle [21]. Figure 5-16 (B) and (C) show the engine and motor torque, and the engine speed over the acceleration test. The engine speed does not go to redline because the added motor torque forces the transmission to shift gears earlier as the upshift speed threshold is reached sooner.

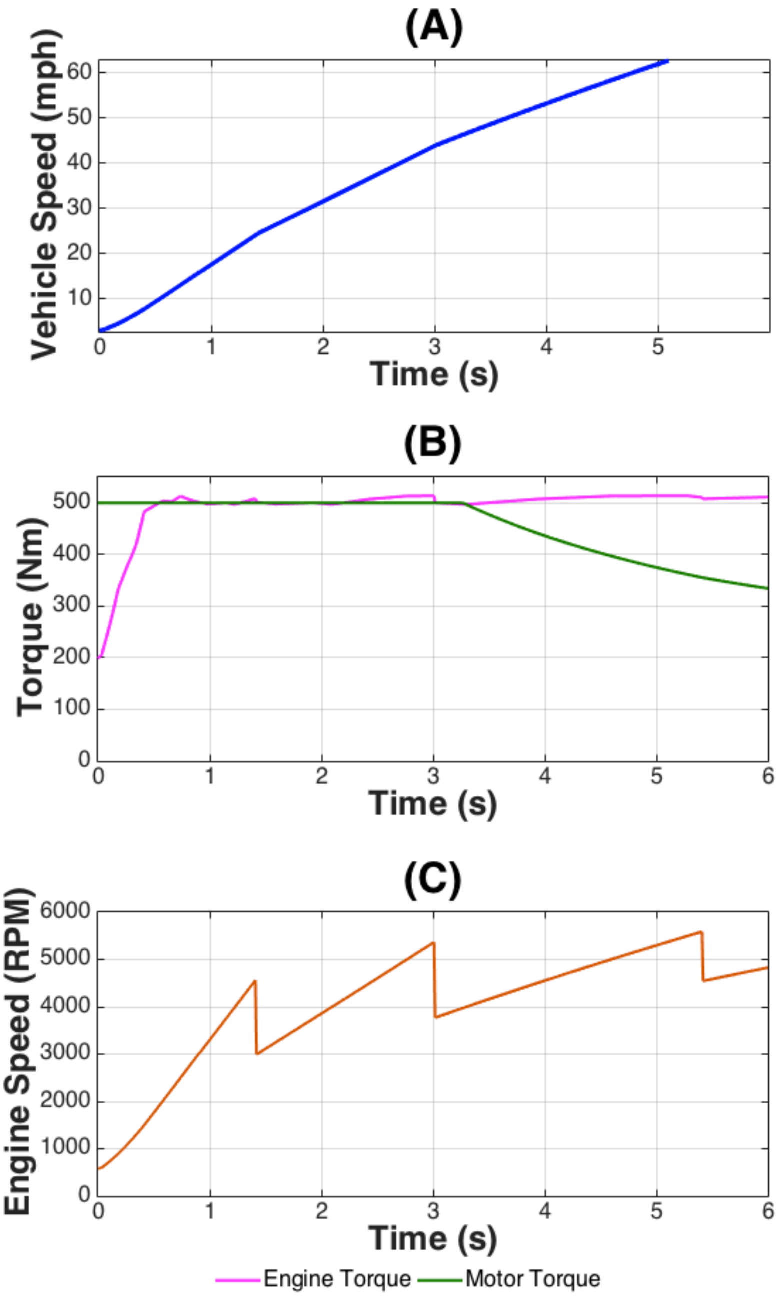


Figure 5-16: Results for 0 – 60 mph acceleration test. (A) Vehicle speed versus time. (B) Engine and motor torque versus time. (C) Engine speed versus time.

Figure 5-17 shows the analysis for the 50 – 70 mph acceleration time. The test shows that the vehicle initially cruises at 50 mph, then a step in desired speed (to 70 mph) is given. The CS control strategy commands the necessary engine and motor torques to reach the desired speed. The transmission downshift can be observed in Figure 5-17(C). The transmission downshifts to 3rd gear to increase the tractive torque available. Any delays in gear shifting are only associated with the fixed time step of the model because time steps are used to address shifting transients in the transmission SoftECU. The lack of torque converter and torque converter clutch models, and the fixed time step delays in the transmission SoftECU are possible causes for the small 50 – 70 mph acceleration time of 2.1 seconds.

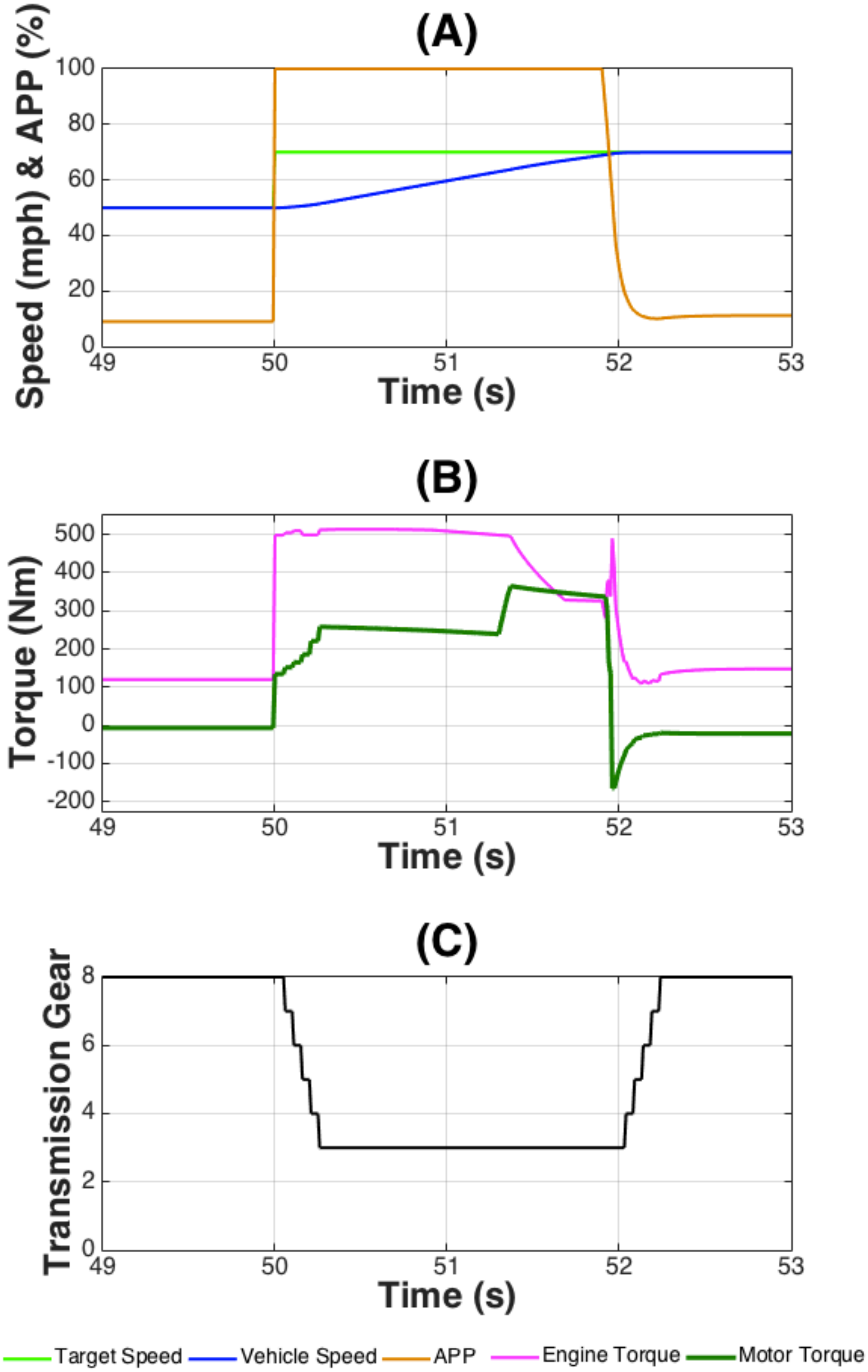


Figure 5-17: Results for 50 – 70 mph acceleration test. (A) Vehicle speed, target speed, and APP versus time. (B) Engine and motor torque versus time. (C) Gear versus time.

5.4.3. Comparison of Results – Autonomie Model & Fuzzy Logic

The performance, energy consumption, and emissions results obtained with the Fuzzy control strategy are detailed in Table 5-8. Comparing the results from the Fuzzy logic to the results from the Autonomie modeling (conducted in Year 1 of EcoCAR 3) shows very similar numbers for all categories. The 50 - 70 mph acceleration times are the very different potentially because the model used with the Fuzzy control strategy lacks torque converter and torque converter clutch models. Conversely, acceleration times from 0 – 60 mph only differ by 0.1 second with tractive force limitations enforced in both models to ensure realistic results based on a power to weight ratio study conducted in [21]. The Fuzzy control strategy increases the CD range by approximately half a mile from 25.5 miles to 26 miles, and slightly increases the CS fuel consumption from 810 Wh/km to 814 Wh/km. The utility factor weighted total energy consumption for the Fuzzy logic is 555 Wh/km compared to 560 Wh/km for the Autonomie model, which yields lower numbers for UF weighted PEU, GHG and criteria emissions. Although the Fuzzy control strategy does not yield significant improvements when compared to the Autonomie results, the Fuzzy strategy is a strong starting point that can be refined in Year 3 and Year 4 of EcoCAR 3 to yield better results and to incorporate system efficiency calculations.

Table 5-8: Performance, energy consumption, and emissions results for Autonomie and Fuzzy logic control strategies.

Vehicle Technical Specification	**HEVT Model P3 V8**	**HEVT Model P3 V8**
	(Autonomie)	**(Fuzzy Logic)**
Acceleration, IVM-60 mph, sec	4.9	4.8
Acceleration, 50-70 mph (Passing), sec	3.5	2.1
Braking, 60-0 mph, ft	< 128	<128
Curb Mass, lb (kg)	4277 (1939)	4277 (1939)
Total Vehicle Range*, mi (km)	170 (274)	170 (274)
CD Mode Range, mi (km)	25.5 (41.1)	26 (41.9)
Pure EV		
CD Mode Total Energy Consumption AC Wh/km (mpgge)	275 (76)	270 (77)
CS Mode Fuel Consumption, Wh/km (mpgge)	810 (26)	814 (26)
UF-Weighted Fuel Energy Consumption, Wh/km (mpgge)	430 (49)	426 (49)
UF-Weighted AC Electric Energy Consumption, AC Wh/km	130	128
UF-Weighed Total Energy Consumption, Wh/km (mpgge)	560 (37)	555 (38)
UF-Weighted WTW PEU, Wh PE/km	125	121
UF-Weighted WTW GHG Emissions, g GHG/km	170	167
UF-Weighted WTP Criteria Emissions g/km (score)	0.0418 (24.9)	0.0415 (25.1)

5.5. Execution

The execution subsystem is the final section of the HVSC algorithm that sends commands back to the vehicle I/O to be routed to components so they can operate as determined in the previous two blocks of the code. The execution subsystem is responsible for handling transient operations in the vehicle and performing logic for component commands. For instance, when the vehicle is in a transition from CD to CS operation, the execution subsystem makes sure that the starter motor relay is commanded to close. The execution block also disables the starter motor after the Engine_Active signal is determined to be active for a specified amount of time. Powertrain and non-powertrain components receive CAN commands, digital enables, and analog voltages from the HVSC. Table 5-9 shows a summarized list of relevant signal for the enabling sequence for the engine, motor, transmission, DC/DC converter, and HVBC. A full list of analog, digital, and CAN signals relevant for vehicle startup and enable operation is provided in Appendix C. The execution block is divided by component, and all necessary commands are determined for each component based on knowledge of documentation/operation and control requirements developed.

Table 5-9: Summarized list of relevant signals for startup and enable operation of engine, motor, transmission, DC/DC converter, and HVBC.

Component	Signal Type	Signal Name	Function
Engine	Digital	Battery Reconnect	Wakes up Body Control Module, which is needed to wake up ECM (Connector X1, pin 2)
	Digital	Any Door-Open Signal	Wakes up Body Control Module, which is needed to wake up ECM (Connector X7, pin 14 & 25)
	Digital	Headlamps On	Wakes up Body Control Module, which is needed to wake up ECM (connector X1, pin 16)
	Digital	Key-In	Wakes up Body Control Module, which is needed to wake up ECM (Connector X3, pin 15)
	Digital	Ignition On	Wakes up Body Control Module, which is needed to wake up ECM (connector X3, pin 6)
	Analog	Parking Lamps ON	Wakes up Body Control Module, which is needed to wake up ECM (Connector X1, pin 22)
	Analog	Keyless	Wakes up Body Control Module, which is needed to wake up ECM (Connector X1, pin 26)
Motor	Analog	ACH_Enable	Pin 7, wakes up the inverter motor system
Transmission	Analog	Accessory Wakeup Serial	Q8 connector pin 9, wakes up TCM
	Analog	Battery Positive Voltage	Q8 connector pin 4, provides power to the controller
	Analog	Transmission Park N Signal	Q8 connector pin 3, reports transmission position
DC/DC converter	Analog	12V connection	0-16V, 11-16 voltage high while active
Charger	Analog	Charging Station Presence	Control pilot signal from charging station (SAE J1772)
	Analog	Voltage Presence	Voltage across high/low pins
	Analog	Power ON	12 V power. Pin 3

5.6. CAN Architecture

The developed control code must effectively communicate commands to controllers in the vehicle. The vehicle CAN network is comprised of all component controllers, and it is divided in seven busses. Four out of the seven CAN busses are directly connected to the HVSC, as shown in Figure 5-18. The chosen HVSC is a dSPACE MicroAutoBoxII that has six CAN channels available. Two channels are reserved for future years of EcoCAR 3 in case issues with CAN bus load and excessive noise are found during component testing. HEVT has the ability to modify the physical CAN layer and re-program only the HVSC, but the stock and other third party controllers cannot be tuned or re-programmed.

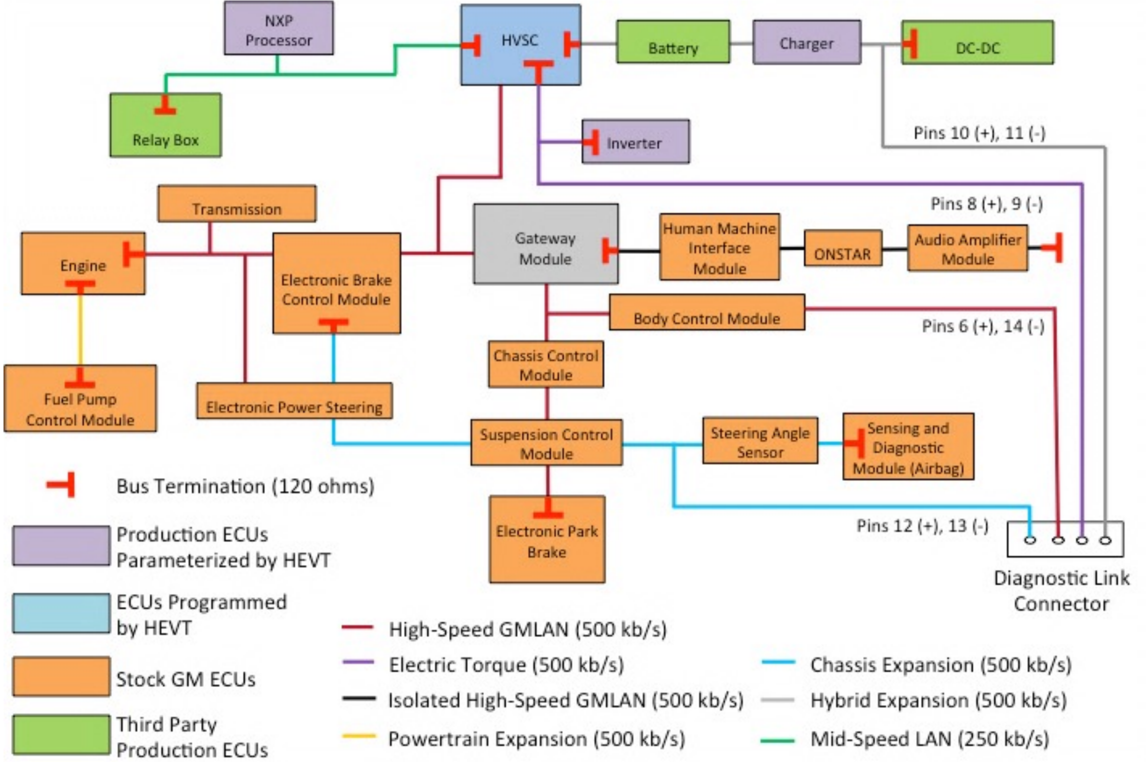


Figure 5-18: Diagram of proposed CAN architecture.

The Mid-Speed LAN, Hybrid Expansion, and Electric Torque busses are custom channels introduced to the vehicle network. The High Speed GMLAN, Powertrain Expansion, Isolated High Speed GMLAN, and Chassis Expansion busses remain in their stock configuration. Transferring information between different busses is sometimes necessary. Gateway modules communicate information between different CAN busses. In Figure 5-18, there are six acting gateway modules: the Electronic Brake Control Module (EBCM), the Gateway Control Module (GCM), the HVSC, the Engine Control Module (ECM), the Suspension Control Module (SCM), and the Electronic Power Steering Module (EPS). All of these gateway modules remain in their stock configuration with the exception of the HVSC. The Chassis Expansion bus, High-Speed GMLAN bus, Isolated ECM bus, and Hybrid Expansion bus are all connected to the Diagnostic Link Connector (DLC) for vehicle diagnostics. The Chassis Expansion and the High- Speed GMLAN bus preserve their stock connection to the DLC, while the Electric Torque bus and the Hybrid Expansion bus are connected to unused pins in the DLC, as specified in Figure 5-18.

Every bus interacting with the HVSC operates at a baud rate of 500 kb/s, except the Mid-Speed LAN, which operates at 250 kb/s. All mid-speed and high-speed busses are properly terminated by two 120 ohm resistors in parallel to provide an equivalent resistance of 60 ohm, thus preventing signal reflection within the busses. Controllers in the low-speed CAN bus do not interact with the HVSC as they do not have an impact on the hybrid control strategy. In addition, the low speed bus does not require terminating resistors because it uses a single wire at a rate of 33.3 kb/s [16].

The Mid-Speed LAN bus consists of a CAN controlled Relay Box to drive actuators, and an NXP Processor, both added by the team. This bus is set to a baud rate of 250 kb/s because information does not need to be transferred frequently between these modules. The innovation processor provided by NXP is placed on this bus as it requires data from multiple control modules to operate. The HVSC is used as a gateway module to forward messages from other controllers to the NXP Processor. This setup avoids bus-overloading that could occur if the NXP Processor was in one of the high-speed busses.

The Hybrid Expansion bus contains the majority of the team-added components. This includes the ESCM, the HV battery charger, and the DC-DC converter. This bus has a baud rate of 500 kb/s because the interactions between these components are critical to the operation of the vehicle. Bus loading is not expected to be an issue, but in case overloading occurs the components will be rearranged to available busses. CAN bus loading will be extensively evaluated through in-vehicle testing.

6. Vehicle & Component Bench Testing

Testing components individually in an isolated environment (bench) is valuable and provides information that facilitates control code development. Several powertrain and non-powertrain components are tested under different conditions before integrating those components in the HEVT Camaro. Bench testing is critical for understanding communication and interface requirements for component controllers. The 5.3L V8 engine and 8L90 transmission are tested on a 2014 Silverado (with load) and on a bench setup (no load). The ESS, and low voltage motor are tested under no load. Although no-load testing does not provide performance or thermal characteristics for the operation of a component, this type of testing does provide necessary information for establishing communication with controllers. Testing powertrain components before in-vehicle integration is one of the goals of this study. Similarly, studying stock vehicle operation (for the 2014 Silverado and 2016 Camaro) is important to characterize the expected behavior of the engine and other vehicle systems (body systems, immobilizer, etc.)

This section details testing activities conducted in the stock 2016 Chevrolet Camaro, 2014 Silverado and engine test stand. Moreover, tests for communication validation with the ESS and low voltage motor are covered in this section. The significance of bench testing resides in the information and knowledge gained to effectively interface with component controllers from the control code. Additionally, bench testing provides an opportunity to improve models for the engine, transmission, and ESS from experimental data. Component bench testing provides valuable information for starting and enabling procedures that complements manufacturer documentation. The information for startup and enable sequences of tested components is detailed in Appendix C.

6.1. Vehicle Baseline Testing

As a requirement of EcoCAR 3, every competing team must evaluate the stock behavior of their donated Camaro. The purpose of the evaluation is to ensure that data is gathered from the operational stock vehicle for future comparison with data from the hybrid powertrain. Although several aspects are evaluated in the stock vehicle, this section focuses on the results from on-road testing and the startup procedure.

6.1.1. On-Road Evaluation

On-road testing is performed to benchmark stock vehicle fuel economy and performance under normal driving conditions. All signals in the GM HSLAN, CE, and LS busses are recorded. The on-road evaluation is performed once in each vehicle mode: tour, sport and snow/ice. Since the focus for this section of testing is fuel economy, the results from Tour mode are shown. The stock vehicle fuel economy was evaluated according to Equation 6-1,

$$FE\ [mpg] = \frac{V\ [mph]}{FC\ [gal/h]} \qquad \text{Equation 6-1}$$

where V is vehicle speed, FC is instantaneous fuel consumption rate, and FE is fuel economy in. The equation is integrated over the full drive cycle to find an average fuel economy of 35 mpg. This value is higher than the EPA estimated combined number of 23 mpg, but matches within 10% with the unadjusted numbers found in the 2016 Test Car List Data of 38 mpg for highway cycles.

The test route is designed to emulate EPA standard drive cycles. This route is shown in Figure 6-1. The cycle is run clockwise from the perspective shown. The route distance is 15 miles traveled in 27 minutes, resulting in an average speed of 34 mph.

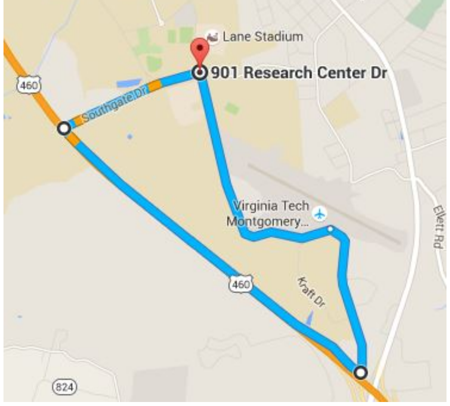


Figure 6-1: Route followed during vehicle testing.

A trace of vehicle speed for the route is shown in Figure 6-2, indicating that the first half of the cycle is city driving, and the latter half is highway.

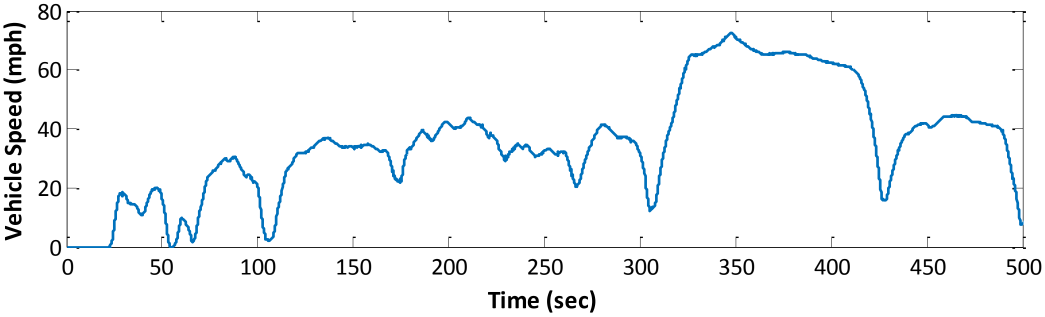


Figure 6-2: Vehicle speed trace recorded during in-vehicle testing.

Figure 6-3 shows the engine speed signal recorded from the HS GMLAN during testing. The drive cycle tested is mild compared to EPA cycles like the US06. The engine speed in the test peaks at 3500 RPM; the low engine speeds maintained throughout the cycle are a large factor in the relatively high average fuel economy.

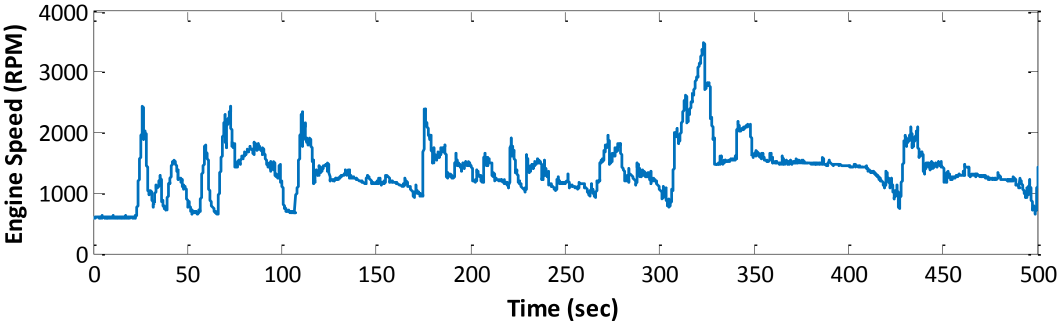


Figure 6-3: Engine speed signal recorded during stock Camaro testing.

6.1.2. Vehicle Startup and Shutdown Evaluation

The starting and enabling sequence followed by the subsystems in the car is a very important process that needs to be fully understood to hybridize the HEVT Camaro. During baseline testing of the stock Camaro, the vehicle startup and shutdown processes are extensively studied by looking, recording, and documenting relevant signals. Examples of analyzed signals include BPP, engine speed, Engine_Run_Active, and System_Power_Mode. CAN bus traffic recorded during several vehicle power states in sequence: completely off, demo mode, engine running, vehicle off (accessory), driver door open, and driver door closed. Figure 6-4 and Figure 6-5 show that run mode is enabled when the driver enters the vehicle. Furthermore, if the driver presses the start button with the foot on the brake, the crank state is entered until the Engine_Run_Active signal becomes true, as shown in Figure 6-6. During the short period that the vehicle is in crank, accessory power is cut for all modules and a crank command is sent to the starter motor.

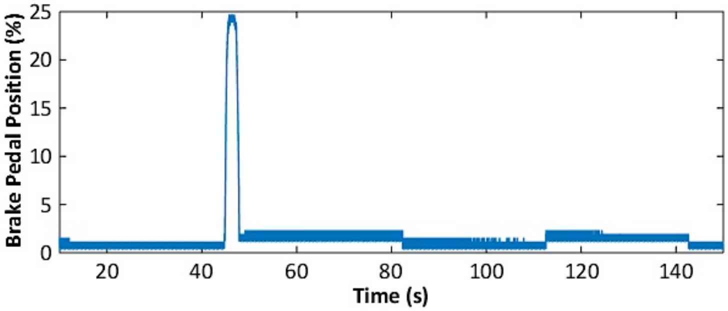


Figure 6-4: BPP versus time.

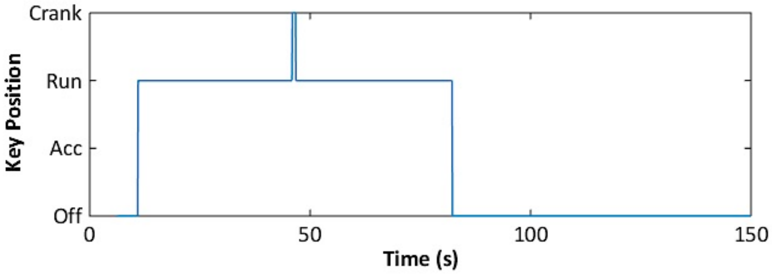


Figure 6-5: System power mode versus time.

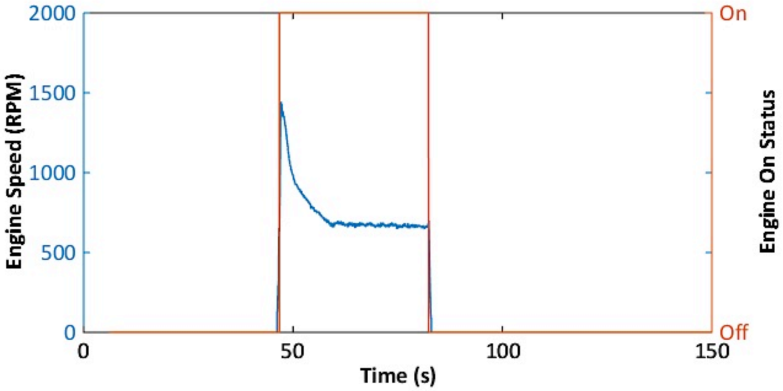


Figure 6-6: Engine speed and Engine_Run_Active signals.

Studying the stock startup procedure is key in the design of a sound enabling sequence for the hybrid powertrain that HEVT seeks to implement for EcoCAR 3. Specifics on other vehicle systems are developed from information learned through testing the startup process; for instance, in the EDS/HVIL system the fuel pump control and ignition relays are disabled when the EDS switch is pressed or when the HVIL circuit is open.

6.2. L83 Engine with 8L90 Transmission

The 5.3L V8 engine is tested in a 2014 Silverado and on a test stand. In-vehicle testing is conducted to evaluate AFM operation. Testing the engine in the original vehicle allows for engine loading during normal vehicle operation, and AFM can be enabled based on stock parameters. These same parameters are used to activate AFM when the L83 engine is run in the HEVT Camaro. Monitoring AFM activation under standard vehicle operation leads to a better understanding of how to improve the hybrid control strategy, which aims to augment time spent in AFM mode while effectively meeting driver demand.

The engine is also run on a test stand to evaluate startup procedures outside of the original vehicle. Connecting the L83 to the systems of the HEVT Camaro presents challenges with communication and immobilizer systems. Understanding the requirements of the immobilizer module and the environmental identifiers for engine startup is essential for successful integration in the HEVT Camaro. The engine is extracted from the 2014 Silverado with the stock 6L80 6-speed transmission. The 6L80 is swapped for an 8L90 8-speed transmission for bench testing, and communication/functionality is verified between the new TCM and existing ECM. Ensuring proper TCM functionality with the L83 ECM is critical for vehicle integration because the 8-speed transmission is used in the HEVT Camaro. Figure 6-7 summarizes the different testing environments for the engine system (blue), the conducted tests (yellow), and the benefits/understanding obtained from all testing environments and activities (orange).

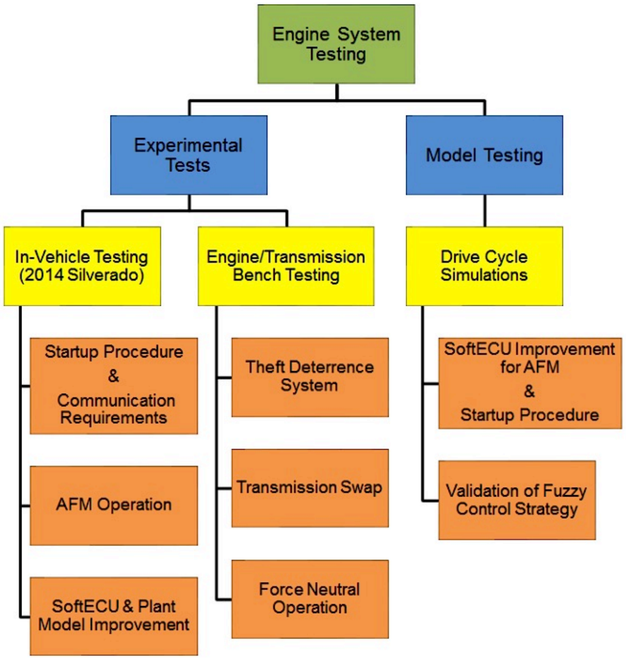


Figure 6-7: In-vehicle and model testing activities conducted for the engine system.

6.2.1. <u>In-Vehicle & Simulation Testing</u>

The main purpose behind in-vehicle testing is to analyze AFM operation under normal driving conditions. In addition, recorded data are analyzed to drive plant model and SoftECU improvement. During the test, the cylinder deactivation CAN signal is monitored with respect to four other CAN signals: APP, engine speed, transmission gear, and engine oil pressure. All of these signals are found on the HS GMLAN bus. A custom drive cycle is performed on a closed course in which elevation changes and ambient temperature are assumed not to have a relevant impact on the tests. The surface is a relatively flat road and the ambient temperature is recorded to be 25°C**.** The vehicle speed trace recorded from the Silverado is shown in Figure 6-8.

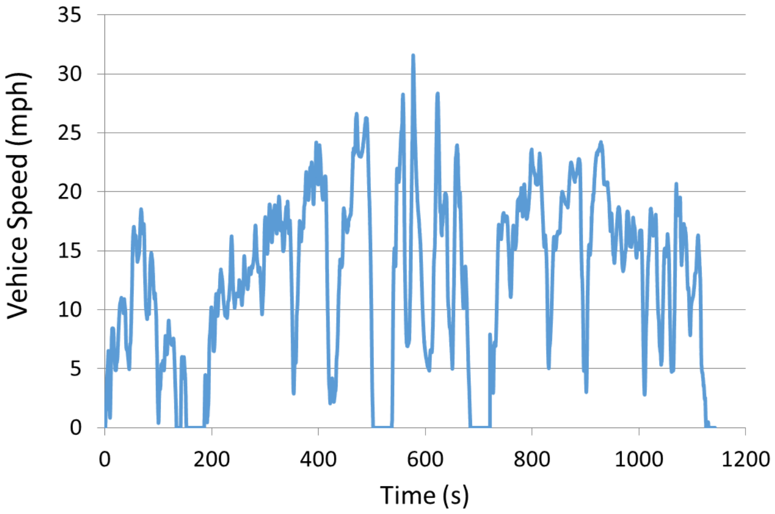


Figure 6-8: Speed trace recorded from in-vehicle testing.

Assumptions and limitations are present for the in-vehicle testing stage based on the driving conditions. The 2014 Silverado purchased by HEVT is wrecked, damage to the front bumper, front fenders, hood, driver and passenger front windows, and front suspension is assumed not to affect engine operation. The engine, transmission, and fuel system appear not to be damaged and run normally. Tests are only performed under normal vehicle operation and do not test the limits of cylinder deactivation. Vehicle speed does not exceed 35 mph, so AFM operation at high speeds cannot be analyzed. Assumptions and limitations are present for the in-vehicle testing stage based on the driving conditions. The 2014 Silverado purchased by HEVT is wrecked, damage to the front bumper, front fenders, hood, driver and passenger front windows, and front suspension is assumed not to affect engine operation. The engine, transmission, and fuel system are undamaged and run normally. Tests are only performed under normal vehicle operation and do not test the limits of cylinder deactivation. Vehicle speed does not exceed 35 mph, so AFM operation at high speeds cannot be analyzed.

A new "drive cycle" speed trace is created from the vehicle speed CAN signal recorded during in-vehicle testing. The Fuzzy logic control strategy is run with the full Camaro vehicle model on this speed trace to replicate the testing conditions in the 2014 Silverado. The results of the simulation testing are compared with in-vehicle results for APP, engine speed, transmission gear, and AFM state and are presented in the following subsections. The vehicle speed trace is also shown in reference to all parameters. Although the Camaro model uses a different vehicle mass, transmission, and final drive than the Silverado, the parameters examined for AFM remain in their expected ranges during vehicle operation.

APP is the primary driver input in all testing, determining engine torque and speed, and transmission gear. Thus, AFM state is strongly influenced by APP. Figure 6-9 shows the correlation between APP and AFM state for in-vehicle testing. AFM is not active when APP exceeded 15%, as expected from GM service information. Sharp increases in APP also deactivate AFM mode, due to the rapid increase in requested engine torque. This characteristic is likely meant to reduce engine wear by minimizing transient torque demand when in AFM mode. There are transition states for AFM activation once conditions are met, likely a safety factor to ensure vehicle operation is steady enough to enable AFM. The same is true for deactivation. For the AFM signal, a value 0 zero indicates V8 operation, 1 is AFM deactivation in progress, 2 is AFM, and 3 is AFM reactivation in progress.

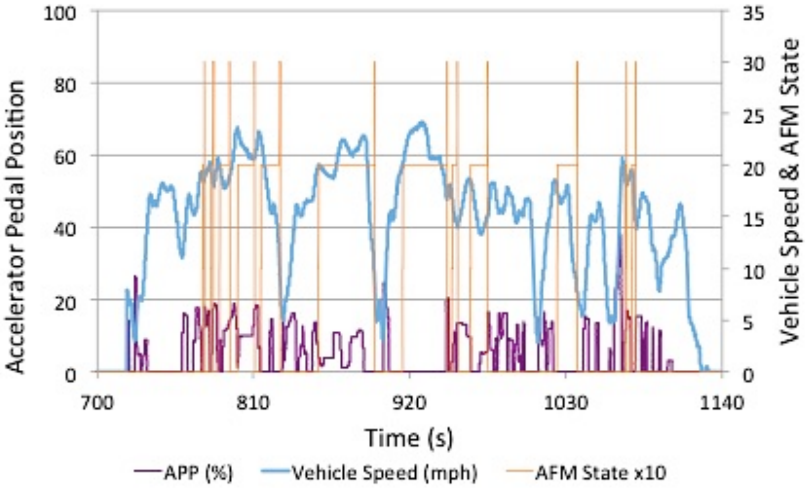


Figure 6-9: In-vehicle testing results for APP and AFM state.

Figure 6-10 shows the simulation results for APP and AFM state.

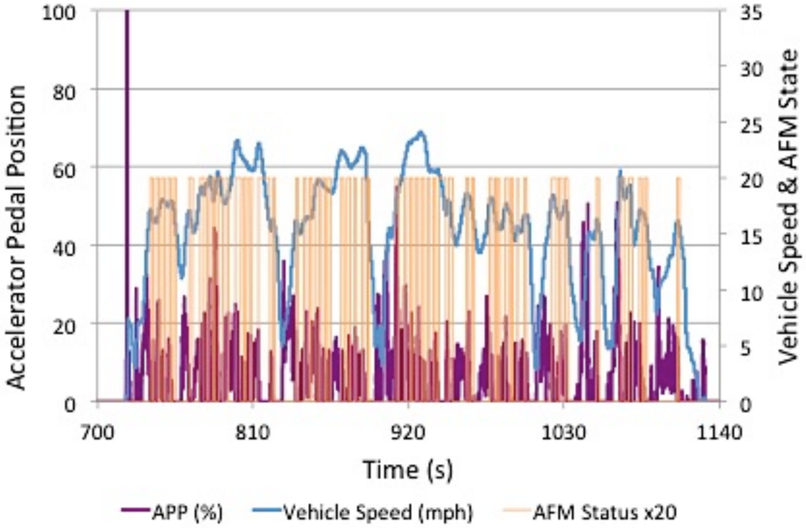


Figure 6-10: Simulation results for APP and AFM state.

Due to limits imposed on the model, AFM is not active for APP above 15%. However, the model only uses a single time step delay between meeting conditions and activating AFM, therefore transition states are not included. Combined with the low number of modeled parameters, AFM mode is active far more frequently in simulation testing than in the vehicle. This issue can be addressed by tuning the driver system that generates the APP in the model to better represent real vehicle operation and avoid transients in AFM. Time delays or hysteresis can be added to the engine SoftECU to further address transient problems with AFM operation.

The comparison of engine speed against AFM state for in-vehicle testing is shown in Figure 6-11. AFM is not active if engine speed is above 1500 RPM. Any time the engine activates or deactivates cylinders, there is a 500-1000 RPM spike before settling back to the original engine speed. The jumps in engine speed are likely used to smooth the change in engine torque with the activation/deactivation of 4 of the 8 cylinders. Sharp engine speed jumps are not directly tied to AFM state, unlike APP spikes. This result is expected because engine speed changes indirectly as APP input changes.

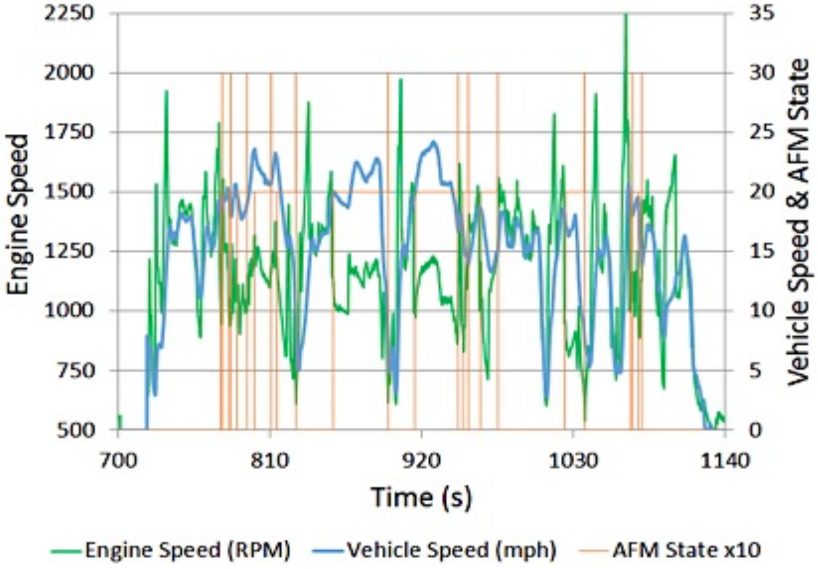


Figure 6-11: In-vehicle testing results for engine speed and AFM state.

Figure 6-12 shows the simulation results for engine speed and AFM state. The frequency with which the model switched AFM on and off does not accurately reflect in-vehicle performance. The model does not command an increase in engine speed when switching AFM on or off, allowing sharp torque spikes at transition points. There is not a good way to introduce this characteristic to the model, but implementing a longer delay before activation and deactivation will mitigate the effect. Engine speed differences between in-vehicle and simulation testing can be attributed to the lack of a torque converter and torque converter clutch in the vehicle model. The engine speed in the model follows the cycle speed trace to a large extent because there is no torque converter that represents the slip occurring in the actual vehicle. This limitation can be addressed by including a torque converter and torque converter clutch model in Year 3 of EcoCAR 3.

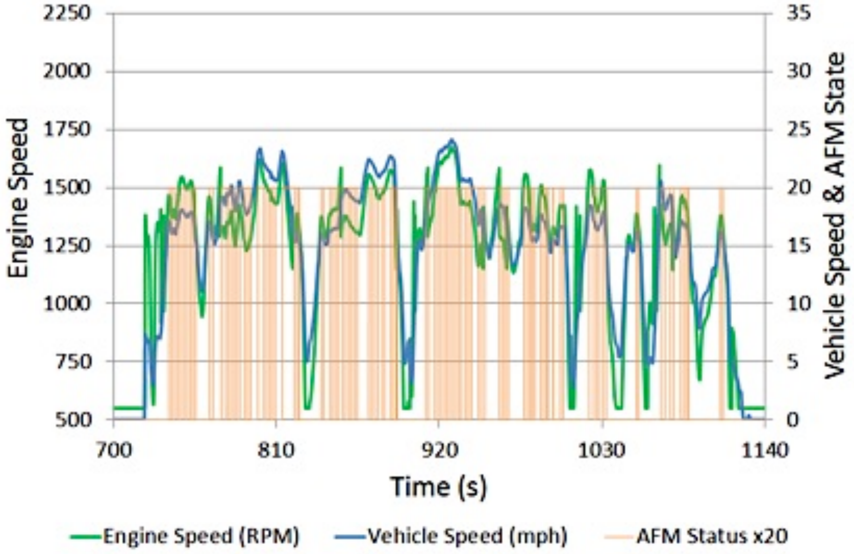


Figure 6-12: Simulation results for engine speed and AFM state.

Transmission gear and AFM state traces from in-vehicle testing are shown in Figure 6-13. Whenever the transmission is in third gear, the engine is able to activate AFM. Once activated, AFM mode can be maintained for a short time even if the transmission dropped into second gear. The noted delay in deactivation likely accounts for transients in transmission gear over short periods of time. Based on these findings, it appears that some of the parameter limits found in GM service information are only entry criteria, not hard requirements to remain in AFM mode. Note that the gear value of 15 is given when the transmission is set to Park.

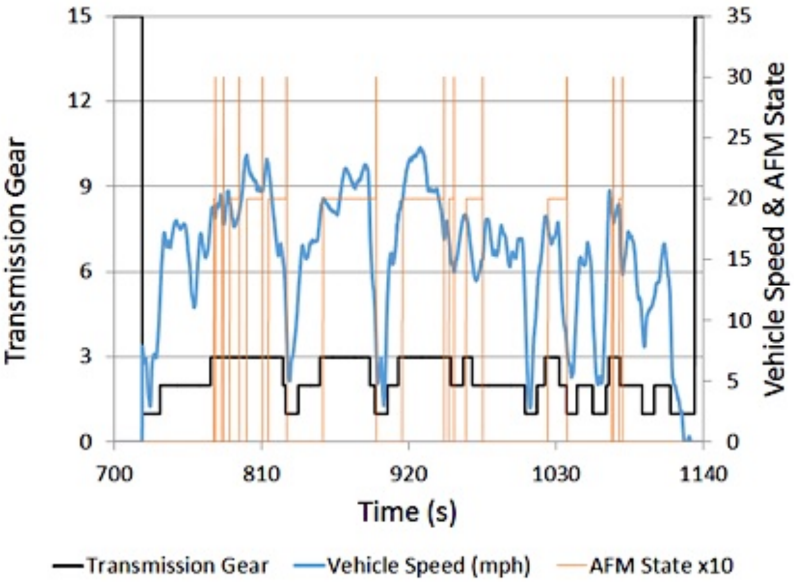


Figure 6-13: In-vehicle testing results for transmission gear and AFM state.

Figure 6-14 shows interaction between transmission gear and AFM state for simulation testing. Note that the model uses an 8-speed transmission with different shift points than the 6-speed transmission used in the vehicle. The model is thus able to enter fourth gear over the same drive cycle because the shift maps of the 8-speed transmission allow for upshifts much sooner than in the Silverado. As in the vehicle, AFM cannot be active if the transmission is not in third gear or higher. The model does not replicate the tolerance for entering second gear (from higher gear and speed) during AFM. A transition state in the engine SoftECU has the potential to minimize this discrepancy.

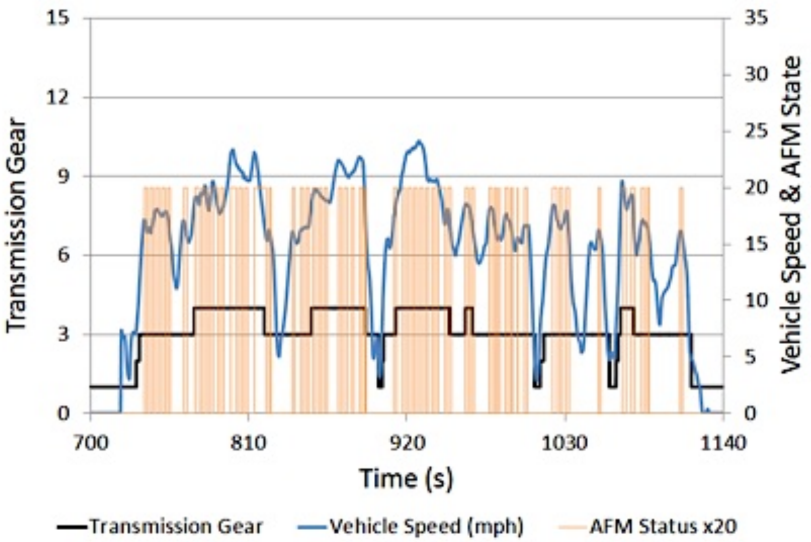


Figure 6-14: Simulation results for transmission gear and AFM state.

6.2.2. Bench Testing – Immobilizer System & Force Neutral

Requirements for the immobilizer system and forcing neutral are analyzed in detail on the test stand. Additionally, a transmission swap is performed, and communication between the ECM and TCM is validated. The test stand consists of a mounting structure, with no ability to load the engine-transmission system. While loaded tests are not possible, the test stand provides useful information on control of the engine. Figure 6-15 shows the engine, transmission, exhaust, and fuel tank mounted on the test stand. The test stand setup has additional controllers, an instrument panel, a gear selector, and key barrel that are not displayed in the figure.

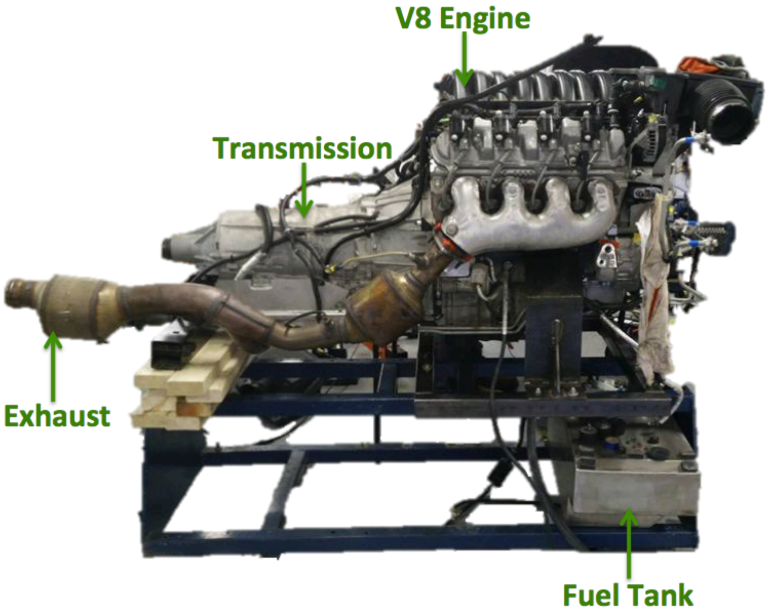


Figure 6-15: Bench testing setup for V8 engine.

The electronic control units (ECUs) involved in AFM testing are the ECM, BCM, transmission control module, and fuel pump control module (FPCM). The ECM monitors and controls engine parameters such as speed, torque, fuel rate, mass air flow, and oil pressure. The ECM communicates with all other involved ECUs. The TCM controls transmission shift patterns and broadcasts current gear state to the other ECUs. The BCM uses signals from the key barrel and the brake pedal position sensor to change vehicle operation mode (off, accessory, run, crank), and perform startup and shutdown procedures. The FPCM modulates fuel pressure and flow rate per request signals from the ECM. Figure 6-16 shows a block diagram containing all components, controllers, and data collection equipment on the test stand. Yellow blocks are driver inputs, blue blocks are physical components, controllers are orange, and data collection equipment is green. Data is recorded using the DLC located on the stock wiring harness of the engine.

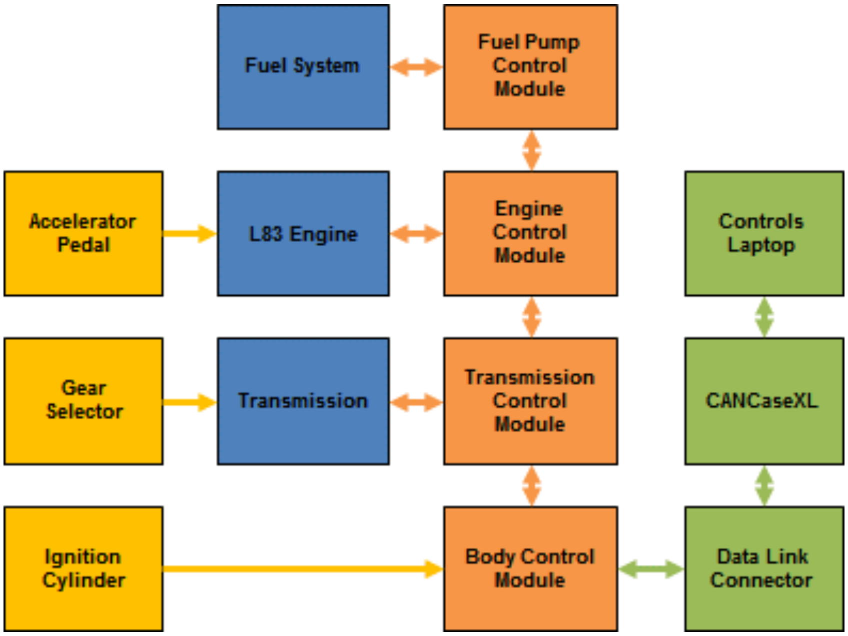


Figure 6-16: Block diagram of bench testing setup.

The 2014 Silverado counts with an immobilizer and environment identifiers. The ignition coil in the Silverado energizes and reads the transponder code (immobilizer code) in the vehicle key. The immobilizer code is sent to the BCM over low speed GMLAN. Then, the code is sent to the ECM with each controller running a set of challenges to verify that the identifier is correct.

The 2016 Camaro adds another layer of complexity to the system by incorporating a keyless entry system along with removing the regular ignition coil. The Camaro ignition coil is located in the rear cup holder. If the key remote battery is low (or depleted) the car does not receive communication to start the engine; if that is the case, the remote can be placed in the cup holder and the transponder code is read, and the engine can be started. For vehicle starts, the BCM sends a signal over LS GMLAN to the “Passive Entry Passive Start” control module, which confirm whether a valid key remote is present in the vehicle.

Testing the Camaro and the V8 engine on a test stand allowed for documentation of necessary identifiers for the immobilizer system of both vehicles. These codes are critical to satisfy the immobilizer system requirements for hybrid vehicle integration. Table 6-1 lists the recorded ID’s.

Table 6-1: Vehicle codes.

Vehicle	Immobilizer Identifier		Environment Identifier	
	Normal	Hex	Normal	Hex
Silverado	44386	AD62	49665	C201
Camaro	33934	848E	23219	5AB3

Forcing the transmission into neutral is required for the HEVT powertrain to run in EV mode. The transmission needs to be in neutral so that the torque produced by the motor is not fed to the engine. The driver must still be able to manipulate the gear selector, which is attached to the transmission via cable. The internal mode switch (IMS) reads the driver intent when this cable rotates the shaft through the sensor. The IMS outputs five signals and a signal reference. The stock vehicle then sends the sensor signals to the transmission control module to process and interpret the signals. The driver intent is then sent to the other ECUs over the high speed CAN network. The stock vehicle routes all signals from the IMS in the transmission to the TCM with a signal reference. The TCM then decides which clutch packs to engage or disengage, as well as the CAN messages to send. Figure 6-17 shows the configuration of the IMS circuit in the stock vehicle.

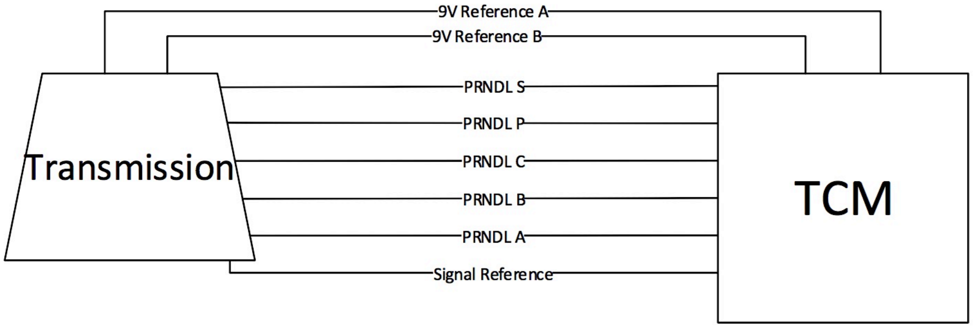


Figure 6-17: Stock IMS circuit.

The HVSC is introduced into the IMS circuit to be able to manipulate the switch voltages that the TCM receives. The HVSC reads the voltages that the IMS sends on the signal wires. The signal references are grounded from the transmission and to the TCM. This sets them to the same ground that the HVSC references the voltages. The signal wires are passed through 150-ohm resistors then grounded in parallel with the HVSC. These resistors have about the same resistance as the TCM and therefore generate the same voltage in the circuit so the HVSC readings are accurate. The HVSC then outputs the proper voltage in each wire to request the proper PRNDL position based on the control code developed. The TCM sends a message that communicates that the driver intent is neutral when neutral is forced. For EV operation, the HVSC has to override the signal from the TCM with the true driver intent. The override is accomplished by transmitting the same message as the TCM at a faster rate. Testing validates that the other ECUs accept the PRNDL position sent from the HVSC and not the TCM. In addition, the functionality of the forcing neutral method with the HVSC is confirmed through bench testing activities. Figure 6-18 shows the modified IMS circuit to force neutral with the HVSC.

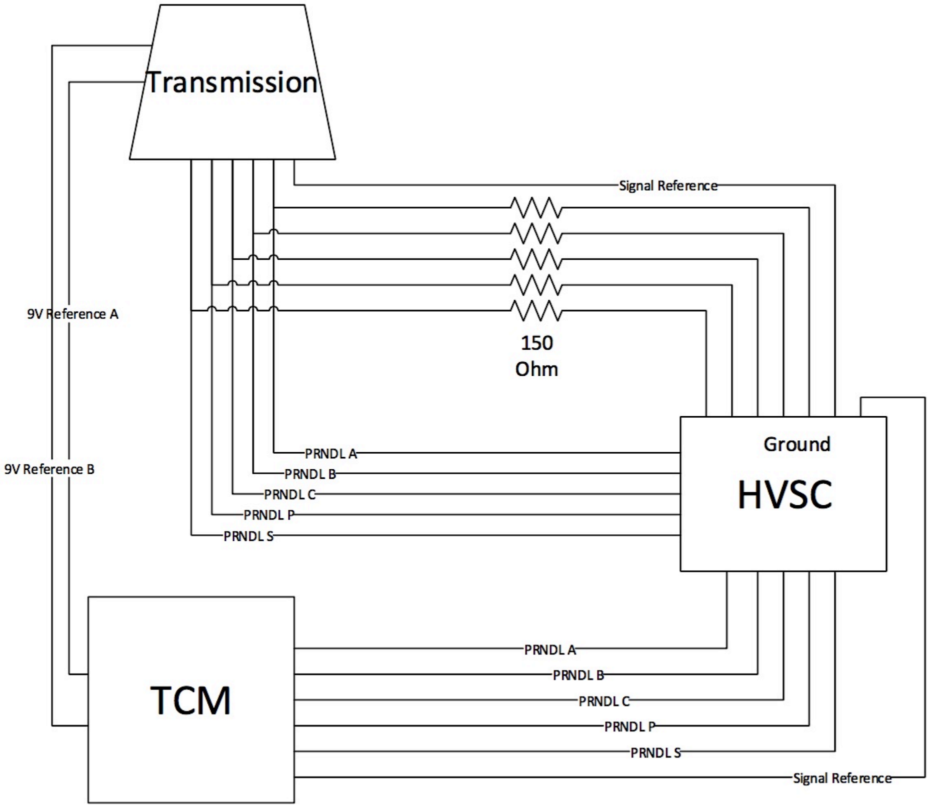


Figure 6-18: Modified IMS circuit to force the transmission into neutral.

6.3. Low Voltage Motor System

The P3 motor is the main source of torque in CD operation. The motor and inverter are sponsored by InMotion. Due to delays in manufacturing and delivery of these components, the sponsor company provided a low voltage setup (motor and inverter). The LV setup allowed for development of a serial communication structure and for bench testing to gain familiarity with the control interface of the system. The main benefit from having a LV motor system available is understanding all the requirements for communication, functionality, and safety before receiving a high voltage system capable of producing 100 kW of power. Guidance from InMotion was invaluable while developing a structure for the serial communication. The CANopen protocol (built on CAN) is chosen and a database is generated. The database file enables HEVT to communicate and control the motor with Vector and dSPACE hardware. Testing of the LV motor system provides information for the adequate enabling procedure of the inverter, where an global enable command (NMT command) must be sent to the nodes using CANopen and an enable command message has to also be sent to the inverter. Understanding these requirements allows for development of a control algorithm that can incorporate these commands in the execution logic.

Figure 6-19 shows the DC bus response of the system when the power stage is enabled (a value of 1). The DC bus current appears to drop to -1 Amp when the power stage is enabled and no command is active. The DC voltage, shown in purple, drops when a speed command is set. Similarly, the DC current goes up and oscillates between 6 Amps and 12 Amps. The behavior of these variables, as it could be similar with the high voltage setup in the vehicle.

Further testing at InMotion facilities is necessary to confirm the expected behavior of the DC bus during in-vehicle operation.

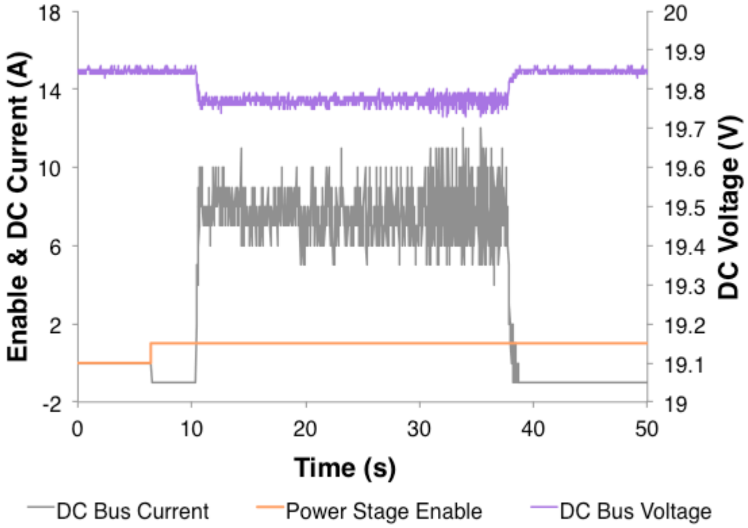


Figure 6-19: DC bus response of motor system after enable and speed commands.

The motor is bench tested in speed mode, which allows for testing without needing a load. In the vehicle, the motor is expected to be mainly controlled in torque mode. Speed mode may be beneficial for in-vehicle applications like EV cruise control, but most motor operation should be performed in torque control mode. Figure 6-20 shows the motor response to a speed command of 3000 RPM and the rotor acceleration. InMotion does the internal calibration of the motor controller. In the case of the high voltage motor system, extensive calibration is to be performed while dynamometer testing occurs. A refined calibration is necessary to ensure that the motor operates efficiently and meets requests effectively. Similarly, dynamometer testing is necessary to study thermal and performance characteristics and limits of the sponsored motor. Limits for acceleration, current, voltage, torque, speed and other relevant variables are to be determined from extensive testing at InMotion facilities after motor delivery.

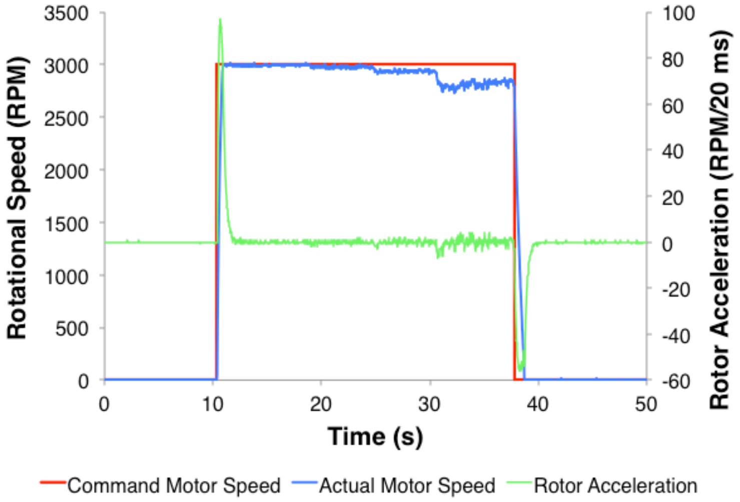


Figure 6-20: Motor response to a speed command.

6.4. A123 High Voltage Battery Pack

The ESS donated by A123 is assembled and tested in a safe high voltage are. Following and reading A123 procedures, schematics, and interface control documentation is necessary to understand all the functional and safety requirements associated with the ESS. The bench test performed with the battery pack validates control communication and the ability to close contactors. Testing is possible due to the construction of an effective low voltage wiring harness and due to live high voltage connections. The ESS is not loaded in the bench testing environment. To be able to close contactors, a high impedance device is placed across the battery terminals so that the EsCM internal diagnostics can be satisfied. Figure 6-21 shows the configuration of the test setup. M3(-) represents the negative side of battery module three. For quick reference, the pin connection numbers are provided in each module and controller in Figure 6-21. The ESS is comprised of the seven battery modules and several controllers. The CSM is the current sensing module, and it is a terminating element on one side of the module connections. The EsCM is the terminating element on the other side of the module connections. The EsCM acts as the supervisory controller for the ESS; this module commands contactors to close or open, and monitors SOC, SOH, and many other relevant variables for safe operation. The MSD is the manual service disconnect for serviceability. The HVIL circuit is open when the MSD is removed. Finally, the EDM is the electric distribution module and it houses the contactors. Communication is established with the EsCM using Vector hardware and software.

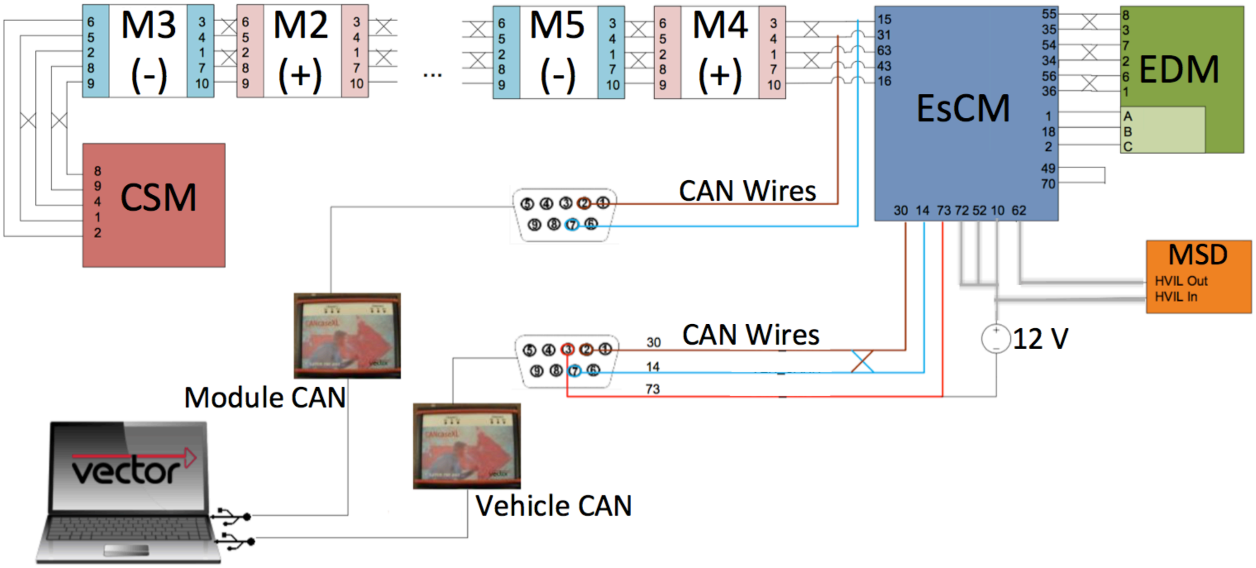


Figure 6-21: Test setup for ESS.

Using the module CAN channel is beneficial to troubleshoot and diagnose problems in the module network. On the other hand, the vehicle CAN network interfaces the ESS with the rest of the vehicle and the HVSC. All commands and ESS feedback are received and sent through the vehicle CAN. Figure 6-22 shows the module enable send via vehicle CAN to the EsCM, and it also the fault monitoring line. The fault monitor line connects the boards of all the battery modules, thus this line can provide internal diagnostics for faults. In addition, Figure 6-21 proves that the fault monitoring line acts as an alive rolling counter or keep alive when the battery modules are enabled.

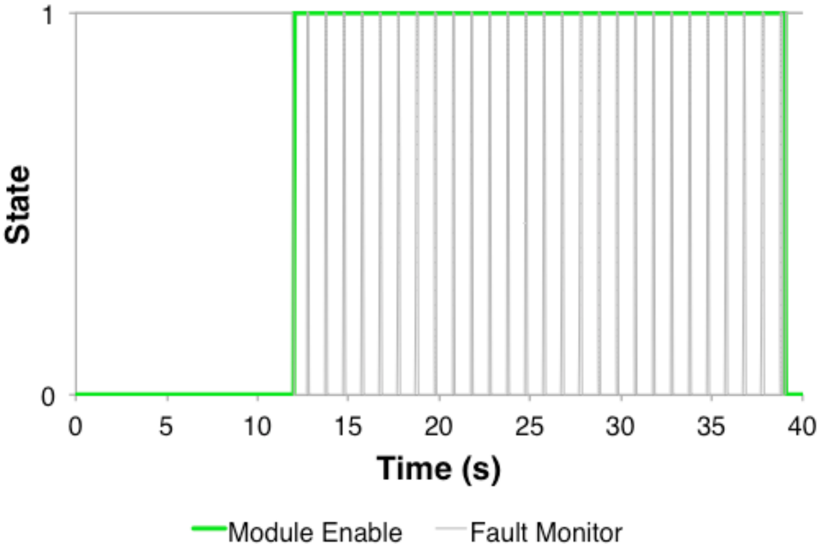


Figure 6-22: Module enable and fault monitoring line during ESS testing.

Finally, Figure 6-23 shows that the bench testing performed allows closing of the battery contactors, which proves that EsCM communication and safety requirements are met. The contactor command and contactor response have small delay, especially when contactor are commanded to close. This delay is the results of the procedure and checks that the ESS performs before closing contactors. Note that 1024 value in Figure 6-23 is a default voltage number for open contactors (1024 = 2^{10}), but when contactors close, the EsCM is able to report a value close to 347 V, which is what is measured with a multimeter in the bench test setup.

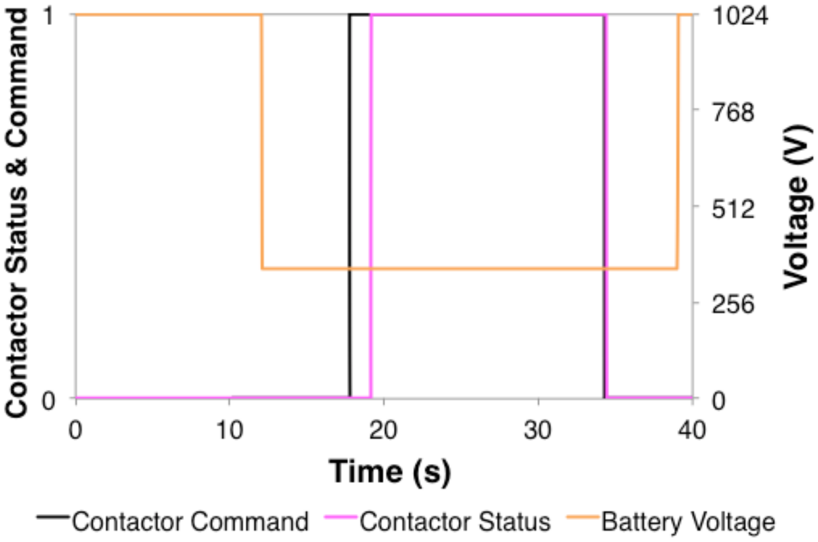


Figure 6-23: Contactor command, contactor status, and battery voltage for A123 ESS.

6.5. Relating Component Testing Information to Control Code & Model Development

Component bench testing provides valuable information that complements manufacturer documentation for starting and enabling procedures. The information for startup and enable sequences of the components tested is detailed in Appendix C. Engine testing yields information about the immobilizer system, force neutral operation, and AFM; the first two categories are relevant for control code development and AFM information is useful for model improvement. Similarly, motor testing yields information about the enable sequence, communication, and control for this component; the learned information is valuable for development of control code for the execution subsystem. Finally, ESS bench testing also yields information about the enable sequence, communication, and control for the battery management system (valuable for execution control logic).

The starting and enabling sequence followed by the subsystems in the car is a very important process that needs to be fully understood to hybridize the HEVT Camaro. During baseline testing of the stock Camaro, the vehicle startup and shutdown processes are extensively studied by looking, recording, and documenting relevant signals. On-road testing is performed to benchmark stock vehicle fuel economy and performance under normal driving conditions. The purpose of the evaluation is to ensure that data is gathered from the operational stock vehicle for future comparison with data from the hybridized Camaro.

Monitoring AFM activation under standard vehicle operation leads to a better understanding of how to improve the hybrid control strategy (in the future), which aims to augment time spent in AFM mode while effectively meeting driver demand. Engine testing also leads to a better understanding of startup procedures outside of the original vehicle. Connecting the L83 to the systems of the HEVT Camaro presents challenges with communication and immobilizer systems. Understanding the requirements of the immobilizer module and the environmental identifiers for engine startup is essential for successful integration in the HEVT Camaro. Testing the Camaro and the V8 engine on a test stand allowed for documentation of necessary identifiers for the immobilizer system of both vehicles. These codes are critical to satisfy the immobilizer system requirements for vehicle integration. Engine bench testing also validates that the other ECUs in the 2016 Camaro accept the PRNDL position sent from the HVSC and not the TCM when forcing neutral. In addition, the functionality of the forcing neutral method with the HVSC is confirmed through bench testing activities. The outcome of force neutral testing is valuable for control code development (CD mode). Additionally, engine simulation testing results are compared with in-vehicle results (2014 Silverado) for APP, engine speed, transmission gear, and AFM state to assess the behavior of the AFM logic and to identify areas of future improvement in the engine model SoftECU.

Testing of the LV motor system provides information for the adequate enabling procedure of the inverter; a global enable command (NMT command) must be sent to the nodes using CANopen and an enable command CAN message has to also be sent to the inverter. Understanding these requirements allows for development of a control algorithm that can incorporate these commands in the execution logic. Similarly, bench testing the battery pack validates control communication and the ability to close contactors, which is also valuable for control algorithm development. In Year 3 of EcoCAR 3, HIL simulations should be conducted to evaluate the behavior of the control code logic developed from bench testing information.

7. Iterative Platform Testing

SIL and HIL testing provide a strong foundation to ensure safety and functionality of the control algorithm. These testing platforms allow for model improvements and control code development from the observed result. The V-cycle implemented in this study requires testing and validation steps after control requirements are determined and algorithms are created. Figure 7-1 shows the process HEVT implements, inspired by the V-cycle, for development and refinement of the control code with SIL and HIL testing.

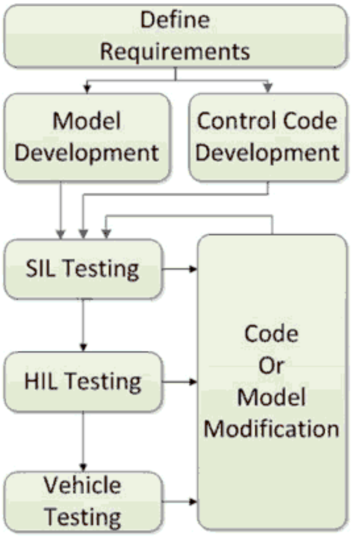


Figure 7-1: Control code testing process with SIL, HIL, and in-vehicle platforms.

This section covers the methods and techniques used by HEVT during EcoCAR 3 to migrate between testing platforms in a seamless manner, while allowing several team members to work on the control code and vehicle model in parallel.

7.1. Migration Between SIL & HIL Platforms

The ultimate objective of iterative testing is to develop a safe, functional, and robust control algorithms that can meet the control requirements set by the designer. In addition, iterative testing aims to produce constant improvement to the vehicle model to better represent the interactions observed from in-vehicle testing. Since the developed code and model go through different platforms, they are modified at every stage of the process. Keeping the control algorithm and vehicle concurrent between testing platforms is critical for the successful implementation of the V-cycle and the process shown in Figure 7-1. Code and model concurrency can be achieved by setting a compartmentalized structure for each element. The goal of the structure is to separate the I/O interface of the elements from the actual control logic and modeling operations. The I/O of the control code and vehicle model changes radically from SIL to HIL because the latter requires setting up variables in the software for use with hardware. The implemented structure in this study separates the control logic and model operations from their respective I/O's. Then, an I/O is created for each element (control code & vehicle model) in each testing platform (SIL & HIL). The logic of the control algorithm and the operations of the vehicle model are implemented using libraries to isolate those two elements and to enable parallel development. The SIL model is broken down into several blocks, shown in Figure 7-2.

Input signals are first scaled to the values required by the control code. In SIL testing, scaling does not require changing the signal values for hardware conditioning, however is it necessary to account for scaling in SIL due to the scaling changes from HIL. Control logic is then performed, and the signals are mapped to proper names and sent to the SoftECUs. After component-level calculations are performed, the SoftECU output signals are scaled back to the control code.

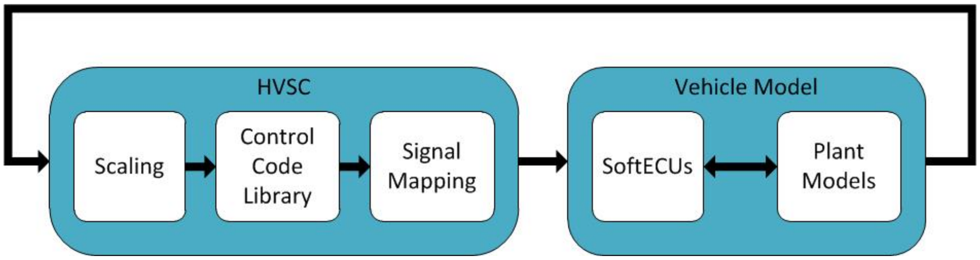


Figure 7-2: SIL testing structure.

The separation of each subsystem allows any block to be easily replaced with a new subsystem which performs the same functions. Thus, the SIL control code block will be moved outside of the I/O structure and replaced with a real-time interface CAN multi-message (RTICANMM) library block for HIL, as shown in Figure 7-3. A similar structure is implemented for the simulator side of HIL testing, with another RTICANMM block receiving signals from the controller and mapping the signals to commands read by the SoftECUs. The RTICANMM blocks allow the HVSC (left side of the figure) to interface with the dSPACE simulator (right side of the figure). Appropriate signal scaling is necessary on both the controller side and simulator side of the HIL setup, and signals must be mapped differently for both the HVSC and simulator.

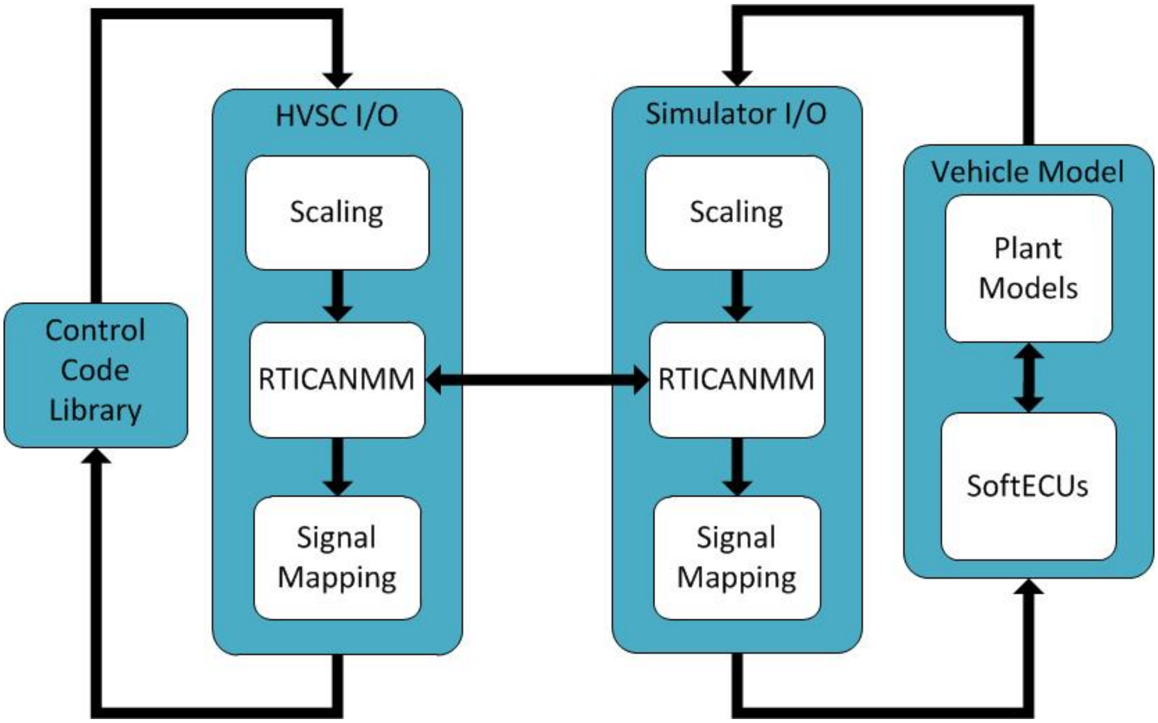


Figure 7-3: HIL testing structure.

Although the proposed structure and method facilitate keeping the control code and model concurrent in SIL and HIL testing, effort is required to update changes in the I/O structure of the control code and vehicle model. The scaling and mapping subsystems have to be updated manually to reflect changes driven by testing results in different platforms. The great advantage of the method implemented is that using libraries for the control code logic and vehicle model automatically updates changes made to any of those two elements from testing

7.2. Libraries in Transition from SIL to HIL

Libraries in Simulink are helpful to define the compartmentalized structure of SIL and HIL testing platforms. Libraries are subsystems with specific functionality that can be recreated several times in a Simulink model or many models. The master block has the ability to modify all instances of a specific library. Similarly, every instance of the library block has the ability to propagate local changes to all other samples. The ability to keep all instances up to date is what enables concurrency of the control code and vehicle model in SIL and HIL testing.

The structure of the vehicle model uses one library block per modeled component and SoftECU. For instance, a library exists for the engine plant and SoftECU, and a different library exists for the battery model and SoftECU. Several team members can work on the vehicle model because of the modular structure. Members working on the vehicle model can be responsible for one or more component subsystems. In this setup, development conflicts are reduced because each developer works on an isolated subsystem. Communication is necessary between different members to make subsystems compatible (battery and motor, engine and transmission, transmission and motor, etc) and to put together the whole vehicle model. Conversely, the control code development group is limited to three people working in series. The small number of people in control code development ensures that effective communication and adequate version control methods are used. The control code group is responsible for internal communication to drive safety and functionality in the algorithm. Additionally, this group is responsible for establishing strong communication with the modeling group assembling the full vehicle model.

Strong communication between the development groups (control code and vehicle model) is necessary to successfully implement the proposed structure in this study. A major advantage of this development setup is the ability to spread tasks and responsibility throughout a large number of workers. Thus, team dynamics and a collaborative environment is maintained and encouraged. The chosen development structure is based on previous team experience, EcoCAR training, and methods highlighted in [17]. The implementation of the methods described in this section is currently in progress, and a future study could build up on the foundation provided by this work.

8. Conclusions

This research study presents and documents several stages of the control system development process for a parallel plug-in hybrid electric vehicle. The goals of this work and the EcoCAR 3 competition are introduced. Advanced methods and techniques like the vehicle development process, model based design, and rapid control prototyping are briefly explained and related to the process of creating the hybrid control system for the P3 parallel powertrain. The use of advanced engineering techniques has a tremendous educational value, and these techniques aid in the accelerated development of a functional and safe hybrid control system.

Then, the process for selecting a hybrid powertrain to meet specific design goals is detailed. Several viable powertrains and the associated VTS are evaluated. The significance of the VTS sets for each combination of components is analyzed in depth. In addition, the impact of several fuels on emissions and energy consumption is carefully analyzed. The P3 parallel configuration with a V8 engine is chosen because it generated the set of VTS that best meet design goals and EcoCAR 3 requirements. The V8 engine also preserves the heritage of the Camaro, which is attractive to the established target market. In addition, E85 is chosen as the fuel for the powertrain because of the decreased impact it has on emissions and energy consumption compared to other fuels like E10 and gasoline. After a powertrain is determined, detailed component plant models and SoftECUs are developed to represent and evaluate the expected hybrid vehicle performance. The sole purpose of the developed vehicle model is to assist the development of the control algorithm for the hybrid powertrain. Learning about component operation, and advanced modeling techniques and theories was a requirement for creating a full hybrid powertrain model that can represent the expected operation of the P3 Parallel vehicle. Consequently, the details of the control system are explained. The expected interactions between components are analyzed to determine control requirements. The structure for these requirements is presented, and their significance resides in their ability to guide the creation and development of the control algorithm that is ultimately deployed in the HVSC target hardware. The structure of the control algorithm is explained in detail. The code has the ability to perform diagnostics to detect faults, then it can select a hybrid operating mode based on SOC, and finally the code sends commands to specific controllers in the vehicle network. The Fuzzy control strategy for torque management in CD and CS modes is covered. In CD operation, the Fuzzy controller effectively commands a motor torque based driver demand and vehicle speed. The Fuzzy controller for CS operation evaluates driver demand, vehicle speed, SOC, and SOC direction to output a number between 0 and 1.5 that determines the proportions of the torque split between motor and engine. This CS Fuzzy controller is validated over the EcoCAR 3 drive cycles (example is provided for US06 City). The developed strategy proves to be functional without having a negative impact of the energy consumption characteristics of the hybrid powertrain (when compared to initial modeling results from Section 3). Finally, testing of the engine, LV motor system, and ESS is conducted and documented. Testing facilitated learning about communication, safety, and functionality requirements for the three components. In some instances, component testing was a driver for improvements to the plant model and the control code. Finally, the processes and tools used to develop control code logic and models in parallel (with several team members) are presented. The method for seamless migration between SIL and HIL while keeping all models up to date is documented and has proved to work effectively during Year 2 of EcoCAR 3.

Opportunities for future work that could build on this research include developing Fuzzy control logic with efficiency considerations. The existing logic provides a strong foundation for improvement in many aspects. Adding new input and output variables can be beneficial to address drivability. Similarly, changing the characteristics and number of membership function could have an impact on the response of the torque split. Furthermore, the developed Fuzzy logic strategy is validated in software, so additional work can be done to aim for hardware and in-vehicle implementation. In Year 3 and Year 4 of EcoCAR 3, drivability calibration should be performed to integrate the engine and motor operation in a seamless manner. Tools, like AVL Drive, and training are available to analyze vibrations in the driveline and decrease their impact with the goal of improving drivability. Finally, engine start/stop capability can be incorporated in the HEVT Camaro. Engine calibration modifications may be necessary. Newly available tools to HEVT provide the ability to re-calibrate stock ECUs. Therefore, engine start/stop capability is more feasible than it has ever been in the past. However, HEVT may face limitations when attempting to quantify the impact that reprograming stock ECUs may have on emissions.

9. References

[1]. Greene, D. L., and Plotkin, S. E., “Reducing Greenhouse Gas Emissions from U.S. Transportation,” Pew Center for Global Climate Change, January 2011.

[2]. “U.S. HEV Sales by Model,” U.S. Department of Energy Alternative Fuels Data Center, http://www.afdc.energy.gov/data/10301, August 2014.

[3]. Haraf, John. "Introduction to the Vehicle Development Process." EcoCAR 3, Kickoff Workshop, Year 1. 2014. Lecture.

[4]. Davenport, Thomas H., and James E. Short. "The New Industrial Engineering: Information Technology and Business Process Redesign." MIT Sloan. Management Review, 15 July 1990. Web. 6 Mar. 2016.

[5]. Edelstein, Stephen. "Ford’s New GT Has More Lines of Code than a Boeing Jet Airliner." Digital Trends. 21 May 2015. Web. 06 Mar. 2016.

[6]. Benoy, Brian. "Introduction to Systems Modeling and Simulations and Controls." EcoCAR 3, Kickoff Workshop, Year 1. 2014. Lecture.

[7]. Dvorkin, William N. "Virginia Tech – Consumer Market Research Report." Scholar. 2014. Web. Mar. 2016.

[8]. “Non-Year-Specific Rules, Revision B”. *EcoCAR 3 Advanced Vehicle Technology Competition.* Argonne National Laboratory. Section L, pg. 102-109.

[9]. “2015 Camaro Performance Catalog”. Chevrolet. August 2014.

[10]. “Where Energy Goes: Gasoline Vehicles”. US Department of Energy. Web. Nov 18, 2014. http://www.fueleconomy.gov/feg/atv.shtml

[11]. Shidore, Neeraj and Icked, Andrew. “Evaluation of Ethanol Blends for PHEVs using Simulation and Engine-in-the-Loop.” Argonne National Laboratory. pg. 11. http://energy.gov/sites/prod/files/ 2014/03/f10/vss049_shidore_2011_o.pdf

[12]. Jo, Y., Lewis, R., Bromberg, L., and Heywood, J., "Performance Maps of Turbocharged SI Engines with Gasoline-Ethanol Blends: Torque, Efficiency, Compression Ratio, Knock Limits, and Octane," SAE Technical Paper 2014-01-1206, 2014, doi:10.4271/2014-01-1206.

[13]. Moawad, A., P. Sharer, and A. Rousseau. "Light-Duty Vehicle Fuel Consumption Displacement Potential up to 2045." (2011). Web.

[14]. Ziegler, J. G., and N. B. Nichols. "Optimum Settings for Automatic Controllers." *J. Dyn. Sys., Meas., Control Journal of Dynamic Systems, Measurement, and Control* 115.2B (1993): 220. Web.

[15]. Ord, David A. *Advanced Vehicle Powertrain Design Using Model-Based Design*. Thesis. Virginia Polytechnic Institute and State University, 2014. *VTechWorks*. Web. 8 Mar. 2016.

[16]. "General Motors Service Information." ACDelco TDS. Web. 12 Apr. 2016

[17]. T. Crain, T. Fayer, B. Fabien and P. Reinhall, "Structuring a Hybrid Vehicle Supervisory Control System Simulink Model for Simpler Version Control with Multiple Software Developers," *SAE Technical Paper Series.* Paper 2014-01-1923, 2014.

[18]. Salmasi, Farzad Rajaei. "Control Strategies for Hybrid Electric Vehicles: Evolution, Classification, Comparison, and Future Trends." IEEE Trans. Veh. Technol. IEEE Transactions on Vehicular Technology 56.5 (2007): 2393-404. Web.

[19]. Manning, Peter C. *Development of a Series Parallel Energy Management Strategy for Charge Sustaining PHEV Operation*. Thesis. Virginia Polytechnic Institute and State University, 2014. *VTechWorks*. Web. 2 Mar. 2016.

[20]. Gao, Dawei, Zhenhua Jin, and Qingchun Lu. "Energy Management Strategy Based on Fuzzy Logic for a Fuel Cell Hybrid Bus." Journal of Power Sources 185.1 (2008): 311-17. Web.

[21]. HEVT. "Virginia Tech – Architecture Selection Proposal." *Scholar*. 15 Apr. 2015. Web. 20 Apr. 2016. <https://scholar.vt.edu/portal/site/baddd21d-b21a-4d4c-8022-c22583bdfac1/page/05260435-0eb6-494f-0020-85cd40ba8f56>.

[22]. A. Rousseau, “Plug&Play Architecture for System Simulation” SIA System Modeling Conference, Paris, May 2015.

[23]. Tamaro, Courtney A. *Vehicle Powertrain Model to Predict Energy Consumption for Ecorouting Purposes.* Thesis. Virginia Polytechnic Institute and State University, 2016. Print.

Appendix A: Control Requirements

Table A-1: Vehicle level control requirements.

Number	Vehicle Requirements
1	Vehicle shall warn and shut off the engine during critical failures
2	Vehicle shall warn and shut off the ESS during critical failures
3	Vehicle shall warn and shut off the motor during critical failures
4	Vehicle shall warn and shut off components that may be malfunctioning critically
5	Vehicle shall slow down when the brake pedal is depressed
6	Vehicle shall be drivable within 10 seconds of startup
7	Vehicle shall not move when Park is selected on the PRNDL.
8	Vehicle shall move forward when Drive or Low is selected.
9	Vehicle shall go backwards when Reverse is selected in PRNDL
10	Vehicle shall use accessories when directed by the driver (wipers, lights, horn)
11	Vehicle shall stop operation when emergency stop switch is pressed
12	Vehicle shall have basic consumer features (AC, heat) during both CS and CD
13	Vehicle shall notify driver of low fluid levels in fuel, oil, and other essential fluids
14	Vehicle shall turn cockpit lights on when driver opens doors
15	Vehicle shall inform the driver with an LED when electric powertrain is ready
16	Vehicle shall provide cooling to all required components

Table A-2: Subsystem level control requirements.

Subsystem	Components	Requirement ID	Requirement
I) Ignition	Keyless entry, 12 V battery, starter motor, engine, high voltage bus, motor, ESS, pedals, gear selector, DC/DC converter	I.A	The ignition system shall only be capable of ignition when the brake pedal is pressed and the gear shift is in park
		I.B	The ignition system shall turn on when the driver presses the brake and start button
		I.C	The ignition system turn off when the driver presses the start button with the PRNDL in P and vehicle on
		I.D	The ignition system shall turn on the accessory circuit when the driver presses the start button without foot on brake
		I.E	The ignition system shall enable the high voltage bus for CD and CS operation
		I.F	The ignition system shall enable the DC/DC to support the 12V system in all operational modes
II) HV Traction System	ESS, inverter, motor, high voltage bus	II.A	HV traction system shall provide power to motor to provide traction to wheels
		II.B	HV traction system shall provide braking torque in regenerative braking
		II.C	HV traction system shall receive power from motor during regenerative braking
		II.D	HV traction system shall stop all delivery of power to powertrain when EDS circuit is open
		II.E	HV traction system shall have an intertia switch that activates at >8g lateral acceleration
III) Mechanical	Engine, transmission,	III.A	The mechanical traction system shall provide tractive force to the vehicle

Subsystem	Components	Requirement ID	Requirement
Traction System	differential, wheels, brakes, alternator	III.B	The mechanical traction system shall find the most suitable transmission gear
		III.C	The mechanical traction system shall maintain charge for 12 volt battery
		III.D	The mechanical traction system shall provide braking effort as required by the vehicle
		III.E	The mechanical traction system provide torque to the drive shaft
		III.F	The mechanical traction system shall provide torque to the rear differential
		III.G	The mechanical traction system shall provide torque to the wheels
IV) Safety Critical Systems	Airbags, seatbelts, EDS, HVIL, HVSC, pedals, power steering	IV.A	The vehicle shall provide the driver with a adequate response time to driver inputs
		IV.B	The vehicle shall have fully functional seat belts
		IV.C	The vehicle shall not have any airbag diagnostic trouble codes
		IV.D	The vehicle shall have proper fault detection for all components
		IV.E	The vehicle shall shut off faulted components when operation is not safe
		IV.F	The driver inputs shall be handled quickly by the HVSC (with latency below SAE limits)
		IV.G	HVIL circuit shall disable electric powertrain if open
		IV.H	EDS shall cut all powertrain operation if switch is pressed
V) Low Voltage System	12V battery, DC/DC converter, starter motor	V.A	LV system shall provide power to starter to crank engine
		V.B	LV system shall power accessory loads inside vehicle
		V.C	LV system shall provide power to headlights and signal lights
		V.D	LV system shall provide power to wipers
		V.E	LV system shall enable the ESCM
VI) Braking System	ESS, motor inverter, ABS, brakes	VI.A	The braking system shall use regen braking when SOC<98%
		VI.B	The braking system shall use brake pad braking when SOC>=98%
		VI.C	The braking system shall providing stopping force to the vehicle
		VI.D	The braking system shall use ABS braking when required
		VI.E	The regen braking shall use the motor to slow the vehicle and charge the ESS
VII) Cooling System	Engine cooling system, ESS, transmission, DC/DC converter, inverter, HVBC	VII.A	The cooling system shall cool the engine
		VII.B	The cooling system shall cool the DC-DC
		VII.C	The cooling system shall cool the inverter
		VII.D	The cooling system shall monitor the transmission, ESS, and HVBC temperatures
VIII) Charging System	HVBC, charging station, ESS, high voltage bus	VIII.A	The charging system shall charge the ESS to a 98% SOC when charge balancing is not desired
		VIII.B	The charging system shall not over charge the ESS
		VIII.C	The charging system shall stop charging operation when an SOC of 98 is reached
		VIII.D	The charging system shall balance the ESS when commanded

Subsystem	Components	Requirement ID	Requirement
		VIII.E	The charging system shall wake the HVSC when operation starts to close ESS contactors
		VIII.F	The charging system shall enable the DC/DC converter to support 12V systems
		VIII.G	The charging system shall enable the HV bus through the HVSC

Table A-3: Component level control requirements.

Component	Category		Component Requirement	
1. Engine	1.1	Starting	1.1.1	Engine starter module should respond to ignition
			1.1.2	ECM shall initiate engine idle after starting
			1.1.3	ECM shall differentiate between hot and cold start procedures
	1.2	Communication Input	1.2.1	The engine ECU shall Receive CAN/Analog/Digital signals from the HVSC
			1.2.2	The engine ECU shall Receive CAN/Analog/Digital signals from the Engine Wiring Harness
			1.2.3	The engine ECU shall Receive CAN/Analog/Digital Signals from the Body Wiring Harness
	1.3	Communication Output	1.3.1	The engine ECU shall Send CAN/Analog/Digital signals from the HVSC
			1.3.2	The engine ECU shall Send CAN/Analog/Digital signals from the Engine Wiring Harness
			1.3.3	The engine ECU shall Send CAN/Analog/Digital Signals from the Body Wiring Harness
	1.1	Normal Operation	1.4.1	Engine shall output torque through the driveshaft
			1.4.2	Engine shall combust fuel internally
			1.4.3	Engine shall control the air to fuel ratio in the cylinders
			1.4.3	Engine shall respond to different modes of driving (CD/CS)
			1.4.4	Engine shall respond to different vehicle operation settings (Performance mode, economy mode)
			1.4.5	Engine shall incorporate active fuel management
			1.4.6	Engine shall provide torque to the alternator
			1.4.7	Engine shall power the air conditioning compressor
	1.5	Diagnostics	1.5.1	The engine ECU shall determine if engine coolant temperature is above 110 degrees C
			1.5.2	ECM shall determine if engine Knock is within acceptable limits
			1.5.3	ECM shall determine if engine speed is between 300 RPM to 6800 RPM
			1.5.4	Engine ECU shall determine if air fuel ration is within appropriate limits
			1.5.5	Engine ECU shall determine if an engine component is not sending an expected signal (not functioning correctly)
	1.6	Stopping	1.6.1	Engine shall stop after key is removed from vehicle or when key is in "off" position
			1.6.2	ECM shall have no power supply when the vehicle is "off"
2. Transmission	2.1	Communication	2.1.1	Vehicle speed sensor shall send signal to TCU (transmission control unit) to determine gear shift changes

			2.1.2	Wheel speed sensor shall send signal to TCU to determine gear shift changes
			2.1.3	Throttle position sensor shall send signal to TCU to determine gear shift changes
			2.1.4	Input speed sensor shall send signal to TCU to determine slippage
			2.1.5	Kick down switch shall send signal to TCU for going into an appropriate lower gear in order to use the full power of the engine
			2.1.6	Traction control system shall send signal to TCU to evaluate road conditions
			2.1.7	Cruise control module shall send signal to TCU
			2.1.8	TCU shall send and receive messages from CAN
			2.1.9	TCU shall send signal to ECM to reduce ignition timing or fuel quantity to reduce the transmission load during heavy throttle
	2.2	Diagnostics	2.2.1	Transmission fluid sensor shall send signal to TCU to downshift if overheating
			2.2.2	Brake light switch shall send signal to TCU to lock shift solenoids
	2.3	Operation	2.3.1	The shift solenoids shall change gears
			2.3.2	Pressure control solenoids shall maintain transmission pressure within steady operation range
			2.3.3	Torque converter clutch solenoid shall regulate the optimal torque that should be applied by torque converter
	2.4	Force Neutral	2.4.1	The HVSC must monitor the Internal Mode Switch C, P, and ECM P/N signals
			2.4.2	In park, reverse, neutral, and manual mode, the HVSC must send the IMS C and P signals to the transmission
			2.4.3	When the gear shifter is in drive and the HVSC requires the transmission to be in neutral, IMS C and P must be switched and sent to the transmission
			2.4.4	When the gear shifter is in drive and the HVSC requires the transmission to be in gear, IMS C and P must be passed through to the transmission
3. Motor	3.1	Starting	3.1.1	The HVSC shall enable the power relay to provide 12V power to the inverter
			3.1.2	The HVSC shall verify that the inverter is sending CAN messages
			3.1.3	The HVSC shall start sending the CAN command message with the Inverter_Enable signal equal to 0
			3.1.4	The HVSC shall verify that there are no faults being transmitted by the inverter, if there are, they must be identified, resolved, and cleared before proceeding
			3.1.5	The HVSC shall enable the inverter through both the CAN command message and the digital enable signal
	3.2	Communication Input	3.2.1	The inverter shall receive a torque request from the HVSC
			3.2.2	The inverter shall receive a motor speed signal from motor speed sensor
			3.2.3	The inverter shall receive a voltage signal from the BCM
			3.2.4	The inverter shall receive a current signal from the BCM
			3.2.5	The inverter shall receive a temperature signal from the inverter temperature sensor
	3.3	Communication Output	3.3.1	The inverter shall send a current demand to the BCM
			3.3.2	The inverter shall output a temperature signal from the inverter temperature sensor

			3.3.3	The inverter shall output a fault signal to the HVSC if CAN messages are corrupted
			3.3.4	The inverter shall output a fault signal to the HVSC if any motor constraints are not met
			3.3.5	The inverter shall output the current motor speed
			3.3.6	The inverter shall output the current motor torque
	3.4	Constraint	3.4.1	The motor shall not exceed the maximum rated torque
			3.4.2	The motor shall not exceed the maximum rated speed
			3.4.3	The motor shall not exceed the maximum rated temperature
			3.4.4	The motor shall decay from peak torque to continuous torque based on thermal considerations
			3.4.5	The motor shall not exceed the torque rate limit
			3.4.6	The motor shall not exceed the speed rate limit
	3.5	Shutdown	3.5.1	The HVSC must command a zero speed to the inverter through the CAN command message
			3.5.2	The HVSC must disable the inverter through both the CAN command message and the digital enable signal
			3.5.3	The HVSC must disable the power relay that provides 12V power to the inverter
4. ESS	4.1	Communication	4.1.1	The BCM shall send CAN messages, to include information on battery state, limits, and faults.
			4.1.2	The BCM shall send and receive messages from the HVSC, such as power demand, wake up, shutdown, operating mode, and operating conditions.
			4.1.3	The BCM shall send and receive messages from other component ECUs, such as the motor.
			4.1.4	The BCM shall send and receive messages from the various ESS controllers, which include the MSD, EMI, EDM, and CSM.
			4.1.5	The BCM shall include a vehicle interface to include vehicle and module CAN connections. The signals will include vehicle and charge wake and enable, as well as CAN connections with the HVSC and control modules.
			4.1.6	The BCM shall include a CAN connection for fault identification which include the EDM and LEM.
	4.2	Diagnostics	4.2.1	The BCM shall conduct failure testing and communicate warnings and diagnostics to the HVSC.
			4.2.2	The BCM shall regulate battery voltage within acceptable limits of 263 V and 378 V.
			4.2.3	The BCM shall monitor the battery SOC to prevent it from dropping below the accepted limit, as this would damage the battery. This limit is currently set at 20% SOC.
			4.2.4	The BCM shall regulate the battery current to not exceed the limit specified by the manufacturer. This limit is 120 A for continuous operation, and 408 A for a 10 sec discharge. There may be slight variation in these numbers based on the ESS SOC and temperature.
			4.2.5	The BCM shall communicate to the HVSC if any of the ESS controllers are disconnected or if power demand cannot be met.
			4.2.6	The BCM shall properly shutdown the ESS if the HVSC commands to do so.
			4.2.7	The BCM shall receive the maximum and minimum cell temperatures and emit a fault if the temperature is above 60 degC.
	4.3	Operation	4.3.1	The BCM shall monitor the temperature, voltage, SOC, internal resistance, coolant flow, and current for the battery.

			4.3.2	The BCM shall account for charge depleting, and charge sustaining by receiving command signals from the HVSC.
	4.4	Safety	4.4.1	The BCM shall protect the battery from operating outside of its safe operating area such as over-current draw, over-voltage draw during charging, under-voltage draw during discharging, over/under temperature, and ground faults.
			4.4.2	The BCM shall communicate with HVSC if battery is over 60 degC and shutdown the ESS to prevent overheating.
			4.4.2	The BCM shall prevent the battery from operating outside of its safety range by implementing a switch, sending signals to various component controllers (such as the MCU) to request not to use the battery, or increase cooling of the surrounding environment.

Appendix B: CS Fuzzy Logic Rules

1. If (APPprcnt is VerySmall) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

2. If (APPprcnt is VerySmall) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

3. If (APPprcnt is VerySmall) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

4. If (APPprcnt is VerySmall) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

5. If (APPprcnt is VerySmall) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

6. If (APPprcnt is VerySmall) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

7. If (APPprcnt is VerySmall) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

8. If (APPprcnt is VerySmall) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

9. If (APPprcnt is VerySmall) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

10. If (APPprcnt is VerySmall) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

11. If (APPprcnt is VerySmall) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

12. If (APPprcnt is VerySmall) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

13. If (APPprcnt is VerySmall) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

14. If (APPprcnt is VerySmall) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

15. If (APPprcnt is VerySmall) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

16. If (APPprcnt is VerySmall) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

17. If (APPprcnt is VerySmall) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

18. If (APPprcnt is VerySmall) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

19. If (APPprcnt is VerySmall) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

20. If (APPprcnt is VerySmall) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

21. If (APPprcnt is VerySmall) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

22. If (APPprcnt is VerySmall) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

23. If (APPprcnt is VerySmall) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

24. If (APPprcnt is VerySmall) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

25. If (APPprcnt is VerySmall) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

26. If (APPprcnt is VerySmall) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

27. If (APPprcnt is VerySmall) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

28. If (APPprcnt is VerySmall) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

29. If (APPprcnt is VerySmall) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

30. If (APPprcnt is VerySmall) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

31. If (APPprcnt is Small) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

32. If (APPprcnt is Small) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

33. If (APPprcnt is Small) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

34. If (APPprcnt is Small) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

35. If (APPprcnt is Small) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

36. If (APPprcnt is Small) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

37. If (APPprcnt is Small) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

38. If (APPprcnt is Small) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

39. If (APPprcnt is Small) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

40. If (APPprcnt is Small) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

41. If (APPprcnt is Small) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

42. If (APPprcnt is Small) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

43. If (APPprcnt is Small) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

44. If (APPprcnt is Small) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

45. If (APPprcnt is Small) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

46. If (APPprcnt is Small) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

47. If (APPprcnt is Small) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

48. If (APPprcnt is Small) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

49. If (APPprcnt is Small) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

50. If (APPprcnt is Small) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

51. If (APPprcnt is Small) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

52. If (APPprcnt is Small) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

53. If (APPprcnt is Small) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

54. If (APPprcnt is Small) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

55. If (APPprcnt is Small) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

56. If (APPprcnt is Small) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

57. If (APPprcnt is Small) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

58. If (APPprcnt is Small) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

59. If (APPprcnt is Small) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

60. If (APPprcnt is Small) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

61. If (APPprcnt is Medium) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

62. If (APPprcnt is Medium) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

63. If (APPprcnt is Medium) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

64. If (APPprcnt is Medium) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

65. If (APPprcnt is Medium) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

66. If (APPprcnt is Medium) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

67. If (APPprcnt is Medium) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

68. If (APPprcnt is Medium) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

69. If (APPprcnt is Medium) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

70. If (APPprcnt is Medium) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

71. If (APPprcnt is Medium) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

72. If (APPprcnt is Medium) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

73. If (APPprcnt is Medium) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

74. If (APPprcnt is Medium) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

75. If (APPprcnt is Medium) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

76. If (APPprcnt is Medium) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

77. If (APPprcnt is Medium) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LittleLoad) (1)

78. If (APPprcnt is Medium) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

79. If (APPprcnt is Medium) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

80. If (APPprcnt is Medium) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

81. If (APPprcnt is Medium) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

82. If (APPprcnt is Medium) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

83. If (APPprcnt is Medium) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

84. If (APPprcnt is Medium) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

85. If (APPprcnt is Medium) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

86. If (APPprcnt is Medium) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

87. If (APPprcnt is Medium) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

88. If (APPprcnt is Medium) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

89. If (APPprcnt is Medium) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

90. If (APPprcnt is Medium) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

91. If (APPprcnt is Large) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

92. If (APPprcnt is Large) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

93. If (APPprcnt is Large) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

94. If (APPprcnt is Large) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

95. If (APPprcnt is Large) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

96. If (APPprcnt is Large) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

97. If (APPprcnt is Large) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

98. If (APPprcnt is Large) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

99. If (APPprcnt is Large) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

100. If (APPprcnt is Large) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

101. If (APPprcnt is Large) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

102. If (APPprcnt is Large) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

103. If (APPprcnt is Large) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

104. If (APPprcnt is Large) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is SmallAssist) (1)

105. If (APPprcnt is Large) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeLoad) (1)

106. If (APPprcnt is Large) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is SmallAssist) (1)

107. If (APPprcnt is Large) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

108. If (APPprcnt is Large) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

109. If (APPprcnt is Large) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MidLoad) (1)

110. If (APPprcnt is Large) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

111. If (APPprcnt is Large) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

112. If (APPprcnt is Large) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

113. If (APPprcnt is Large) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

114. If (APPprcnt is Large) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is AllElectric) (1)

115. If (APPprcnt is Large) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

116. If (APPprcnt is Large) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

117. If (APPprcnt is Large) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

118. If (APPprcnt is Large) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

119. If (APPprcnt is Large) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

120. If (APPprcnt is Large) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

121. If (APPprcnt is VeryLarge) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

122. If (APPprcnt is VeryLarge) and (VehSpdMPH is CitySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

123. If (APPprcnt is VeryLarge) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

124. If (APPprcnt is VeryLarge) and (VehSpdMPH is CitySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

125. If (APPprcnt is VeryLarge) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

126. If (APPprcnt is VeryLarge) and (VehSpdMPH is CitySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

127. If (APPprcnt is VeryLarge) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

128. If (APPprcnt is VeryLarge) and (VehSpdMPH is CityFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

129. If (APPprcnt is VeryLarge) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

130. If (APPprcnt is VeryLarge) and (VehSpdMPH is CityFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

131. If (APPprcnt is VeryLarge) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

132. If (APPprcnt is VeryLarge) and (VehSpdMPH is CityFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

133. If (APPprcnt is VeryLarge) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

134. If (APPprcnt is VeryLarge) and (VehSpdMPH is MidRange) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

135. If (APPprcnt is VeryLarge) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

136. If (APPprcnt is VeryLarge) and (VehSpdMPH is MidRange) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is MediumAssist) (1)

137. If (APPprcnt is VeryLarge) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

138. If (APPprcnt is VeryLarge) and (VehSpdMPH is MidRange) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

139. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is SmallAssist) (1)

140. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwySlow) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

141. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

142. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwySlow) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

143. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeAssist) (1)

144. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwySlow) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

145. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is BottomTop) then (EngTransTrqFraction is MediumAssist) (1)

146. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwyFast) and (SOCprcnt is Low) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

147. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeAssist) (1)

148. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwyFast) and (SOCprcnt is Window) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

149. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is BottomTop) then (EngTransTrqFraction is LargeAssist) (1)

150. If (APPprcnt is VeryLarge) and (VehSpdMPH is HwyFast) and (SOCprcnt is High) and (SOCdirection is TopBottom) then (EngTransTrqFraction is LargeAssist) (1)

Appendix C: Enable Signals in Execution Subsystem

Table C-1: List of relevant signals of major components for startup and enable sequence.

Component	Transmitted Or Received	CAN Signal	Full Signal Name or Action	Function
Engine	Received	SysPwrMd	Power enabled	System Power Mode is used as an input in determining whether to initiate cranking in the Starter Control algorithm.
	Received	BatVlt	Battery Voltage	Receives battery voltage from BCM
	Received	ImmblzrPreRelPswd [password]	Immobilizer Password	Provides immobilizer password
	Received	EngIdlAtv	Engine Idle Activated	See Full Name
	Received	EngFuelCntlState	Engine Fuel Control State	See Full Name
	Received	any		Any signal on the HSCAN line will wake up the Body Control Module
	Received	AccActPos[%]	Accelerator Actual Position	See Full Name
	Received	ThrPos[%]	Throttle Position	See Full Name
	Received	AccPos[%]	Accelerator Position	See Full Name
	Transmitted	ImmblzrPreRelPswrdStat	Immobilizer Password Status	Checks to see if the immobilizer password occurs
	Transmitted	ImoInf	Immobilizer Information	Information about the immobilizer
	Transmitted	PTImmblzrInfo	Powertrain Immobilizer Information	Identical to above, but transmitted to the transmission
	Transmitted	Eng12vStrtrMtrCmmdOn	Engine 12 Volt Starter Motor Commanded On	See Full Name
	Transmitted	ETC_MinRunTorq[Nm]	Engine Transmission Minimum Run Torque	See Full Name
	Transmitted	PT_CrnkAct	Powertrain Crank Active	When the ECM has activated the starter relay. This signal may be used for Key Crank and BAS+ High Voltage Contactor Control in hybrid vehicle applications.
	Transmitted	PTCrnkAbrted	Powertrain Crank Aborted	When the ECM is not attempting to crank the engine or has aborted the crank while in the Crank Request power mode and the engine is not yet started.
	Transmitted	PTRUnAbrt	Powertrain Run Aborted	When the ECM has disabled fuel during remote engine running operation initiated by the Remote Start System and the fuel disable is for conditions that do not allow a re-start.

Component	Transmitted Or Received	CAN Signal	Full Signal Name or Action	Function
	Transmitted	PTO_PT_RunAbrtd	Power Take Off Powertrain Run Aborted	When the ECM has disabled fuel during PTO remote engine running operation initiated by the PTO Remote Start System and the fuel disable is for conditions that do not allow a re-start.
	Transmitted	EngCntrlRunCrnkTrmSt	Engine Controller Run Crank Terminal Status	This signal is needed to perform the Run Crank input rationality to fulfill an OBD requirement.
	Transmitted	ETC_MinTorq [Nm]	Engine Transmission Minimum Torque	See Full Name
	Transmitted	ParkNeutralSwStat	Park Neutral Switch Status	See Full Name
	Transmitted	BsTrgtEngIdlSpd	Target Engine Idle Speed	See Full Name
	Transmitted	EngColdHiIdleActive	Engine Cold Idle Active	See Full Name
	Transmitted	EngRunng	Engine Running	See Full Name
	Transmitted	EngRunAtv	Engine Run Active	Indication from powertrain that the engine is running (has RPM > 0)
Transmission	Received	BatVlt	Battery Voltage	See Full Name
	Received	PMMACCTrmSt	Terminal Status	See Full Name
	Received	PMMRunCrkTrmSt	Run Crank Terminal Status	See Full Name
	Received	CmndAxlTrqImm[Nm]	Command Axle Torque Immobilizer	See Full Name
	Received	EngPrdctStdyStTorq	Engine Predicted Steady State Torque	See Full Name
	Received	TransNtrlCntrlMdStat	Transmission Neutral Control Status	See Full Name
	Transmitted	ImoInf	Immobilizer Information	See Full Name
	Transmitted	TrnsShftLvrPos	Transmision Shift Lever Position	See Full Name
	Transmitted	TrnsEngdState	Transmission Engaged State	See Full Name
	Transmitted	EngRunng	Engine Running	See Full Name
Battery	Received	hsc_escm_enable	ESCM Enable	ESCM waits for hsc signal
	Received	hsc_escm_mainc_close	Close Contactors	escm waits for hsc contactor close command
	Transmitted	escm_veh_mon	Ignition On	Performs wakeup tests
	Transmitted	escm_hvil_mon	HVIL Confirmed	Confirms HVIL power present
	Transmitted	escm_cpwr_cmd	Power Contactors	ESCM enables contactor power
	Transmitted	escm_cpwr_mon	Power Contractors	ESCM confirms contractors are connected
	Transmitted	escm_ready	Detect Contactor Power	ESCM confirms contactor power relay closed

Component	Transmitted Or Received	CAN Signal	Full Signal Name or Action	Function
	Transmitted	escm_mainc_step	Contactor Status	Indicates status of contactor closing, including weld check, precharge contactor, negative contactor check, precharge, positive contactor close, open precharge, voltage check
	Transmitted	escm_mainc_stat	Drive State	Indicates that the pack is ready for HV loads
Motor	Received	CommandEnable	Enables Motor	The HVSC will send this signal to enable and wakeup the motor
Battery Charger	Received	Control	NLG5_C_EnablePowerStg	I/U target value guidelines, control bits
	Transmitted	Errors	Errors	Fault causes and warnings
	Transmitted	Status	Status	Status of the regulator and the limiter
	Transmitted	Internal Values	Internal Values	I/U current values of the NLG5
	Transmitted	External Values	External Values	Current external values
	Transmitted	Temp	Temp	Internal and external temperatures
DC/DC converter	Received	0x1D7	Request_VirtualNetworkControl	Setpoint tells the DC/DC whether to operate in buck/boost mode and instructs it what output we desire. This must be sent every 25 ms