Novel Electric Vehicle Architecture Selection, Model Development, and Design of Controls Testing Framework and Workflow.

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Dominic Dallago

Graduate Program in Mechanical Engineering

The Ohio State University

2024

Thesis Committee

Dr. Shawn Midlam-Mohler, Advisor

Dr. Giorgio Rizzoni, Committee Member

Abstract

The United States has made a commitment to de-carbonization of the transportation sector. Around seventeen percent of all US greenhouse gas emissions come from light-duty passenger vehicles [1]. Electric vehicles (EVs) provide a promising pathway to achieving that goal. EVs are an emerging technology in the automotive sector and are now just being produced at the same scale as internal combustion engine vehicles. During this time, rapid innovation in the EV space is possible, and unique EV architectures and designs can be explored. Further, a standardized testing and development process is needed for vehicle models and controllers. This work presents an architecture selection process that explores a large range of unique EV architectures and tabulates all of the data. Next, from that architecture selection process, a physical vehicle is constructed, and a new drivetrain controller is developed. The processes used for requirements generation and management, model development, and controller testing are presented in this work. The presented processes are all done to an industry-level standard and present a future steppingstone for further exploration of the EV design space.

Acknowledgments

I would first like to acknowledge my mother, grandma, and grandfather for always supporting me. If not for them, I would not be here today and have accomplished all of my education goals.

Next, I would like to thank my advisor, Dr. Shawn Midlam-Mohler, for giving me the opportunity to come to The Ohio State and work on the EcoCAR team and for all the support he has given me over the past two years.

Finally, I would like to thank Dax Chhatralia, Ian Campbell, and Pranay Ketineni, who have been by my side throughout this entire project. This work is only possible due to their collaborative efforts and support.

Vita

2021 - 2022	Undergraduate Research Assistant, USU Aspire Project, Colorado State University
2022 – 2024	Graduate Research Assistant, EcoCAR Propulsion Controls and Modeling, The Ohio State University Center for Automotive Research
2023	Controls Systems Intern, Hydrogen Fuel Cell Team, Caterpillar

Fields of Study

Bachelors of Science, Colorado State University, 2018 - 2022

Major Field: Mechanical Engineering

Masters of Science, The Ohio State University, 2022 - 2024

Major Field: Mechanical Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Motivation

The US is producing greenhouse gasses (GHG) and other pollutants at an unsustainable rate. The transportation sector is the highest GHG producing sector in the US [1]. However, transportation is necessary as it drives the US economy and allows people to connect and grow relationships. In the US, an automobile is the main form of transportation for an individual. Light-duty vehicles make up the most significant amount of GHG emissions by the transportation sector. In order to reduce the emissions in the transportation sector, light-duty vehicles must become cleaner and more efficient. Electric vehicles provide a pathway to increase the efficiency of light-duty vehicles, thus reducing overall energy consumption and GHG emissions.

Electric vehicle production has been steadily on the incline in the US and around the world. As the automotive sector becomes increasingly electric, rapid innovation is necessary. The model-based design will play an even larger role in development to cut down on costs and turnaround time. This work aims to present a unique take on this process and apply it directly to a novel electric vehicle and controller strategy.

EcoCAR Electric Vehicle Challenge

This project was completed in parallel with the EcocCAR Electric Vehicle Challenge. This challenge is a four-year competition between 13 North American Universities and is a part of the Advanced Vehicle Technology Competition (AVTC) series. These competitions are managed by Argonne National Lab and sponsored by the Department of Energy, General Motors, and MathWorks. The competition challenged teams to apply innovative drivetrain designs and autonomous features to a 2022 Cadillac LYRIQ to increase energy efficiency without sacrificing vehicle performance or drivability. The competition happens over a four-year cycle; this project coincided with years one and two of the competition.

Year one was focused on architecture selection and vehicle and system design. An in-depth architecture selection process was completed in order to determine the vehicle architecture that best fit the team’s vehicle technical specifications (VTS). These can be seen below in Table 1. The team's vehicle technical specifications were determined based on an in-depth market study of the current electric vehicle market. With an emphasis on electric luxury SUVs since that is the specific market category the Cadillac LYRIQ falls into.

Table 1: Team VTS Table

Specification	Units	Market Average		Team VTS Targets	
Acceleration (0 – 60 MPH)	s	5.5		<= 6	
Acceleration (50 - 70 MPH)	s	2.8		<= 3	
Unadjusted AC Energy Consumption	Wh/mi	UDDS	HWFET	UDDS	HWFET
		240	290	310	325
Adjusted Combined Range	mi	250		>= 207	
Braking (60 - 0 MPH)	ft	NA		<= 51	
Lateral Acceleration	g	NA		>= 0.79	
Cargo Capacity (Rear Seats Up/ Seats Down)	ft^3	28 / 65		15 / 45	
Passenger Capacity	seats	5		5	
Curb Mass, maximum	kg	2,600		< 2,850	
Ground Clearance	in	7		7	

The architecture that was chosen is a dual motor all-wheel drive (AWD) configuration with one Dana iS4500 integrated drive unit on each axle, as shown in Figure 1. This permanent magnet electric machine can produce 180 kW of peak power and 380 Nm of peak torque. On the front axle, it is paired to a single speed reducer gearbox with a gear ratio of 9.1:1. On the rear axle, the Dana iS4500 is paired to a single speed reducer gearbox with a gear ratio of 14.0:1. There is a spline disconnect on this drive unit which is used to decouple the rear half shafts to the gearbox. It can be actuated in cases where the vehicle does not require the rear axles' torque and improves efficiency. The vehicle will generally operate both electric machines with an efficiency priority control strategy. This optimally engages the rear axle for higher torque and acceleration. In the AWD mode, the vehicle prioritizes performance and keeps the rear axle connected. In FWD mode, the rear electric machine is not utilized.

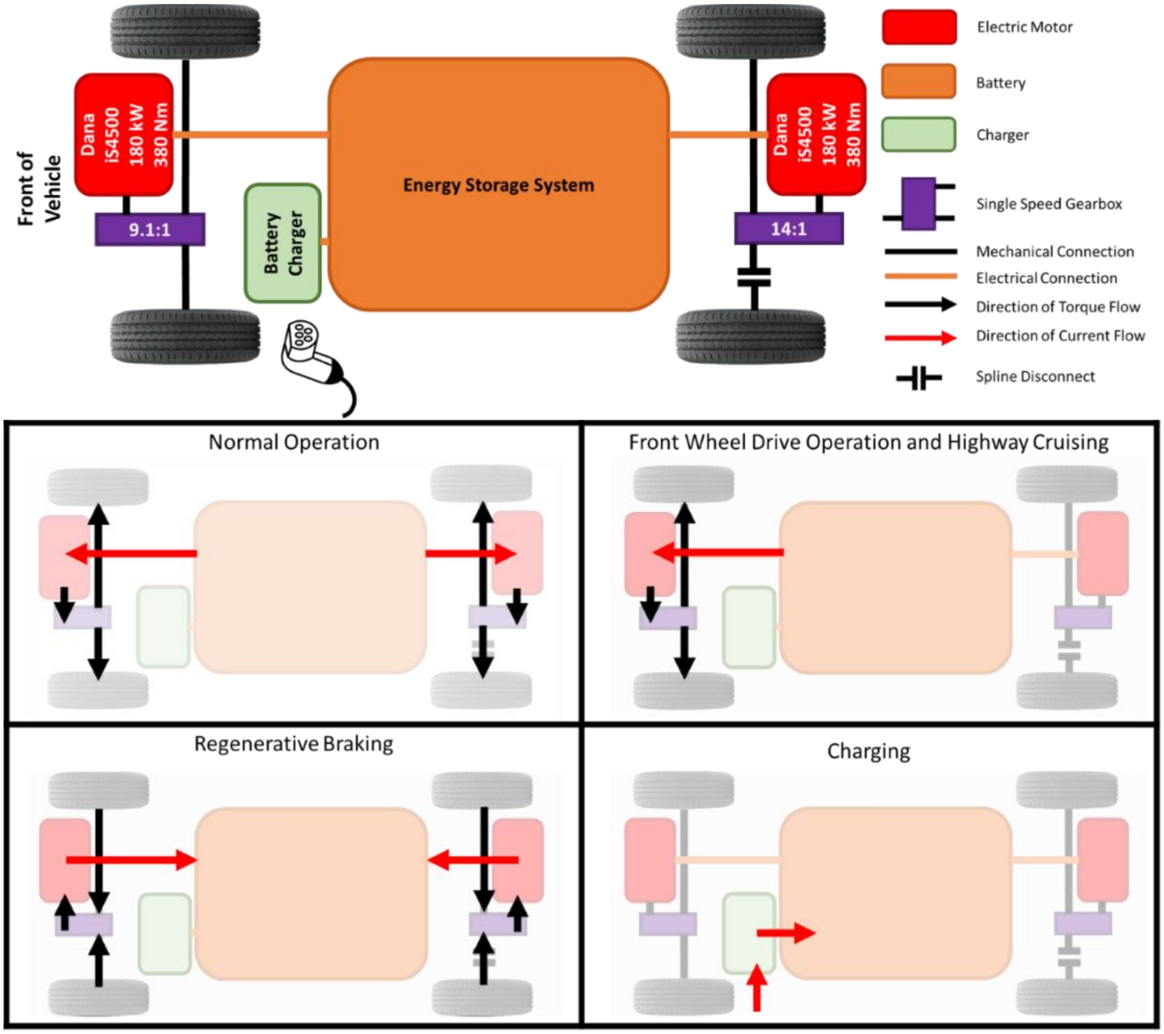


Figure 1: Team Chosen Architecture and Power Flow

Furthermore, in year one, the team also decided on the data communications architecture of the team-added systems in the vehicle. The team employed CAN, LIN, digital, and analog communications to run their components in conjunction with the GM stock system. The communication architecture can be seen below in Figure 2.

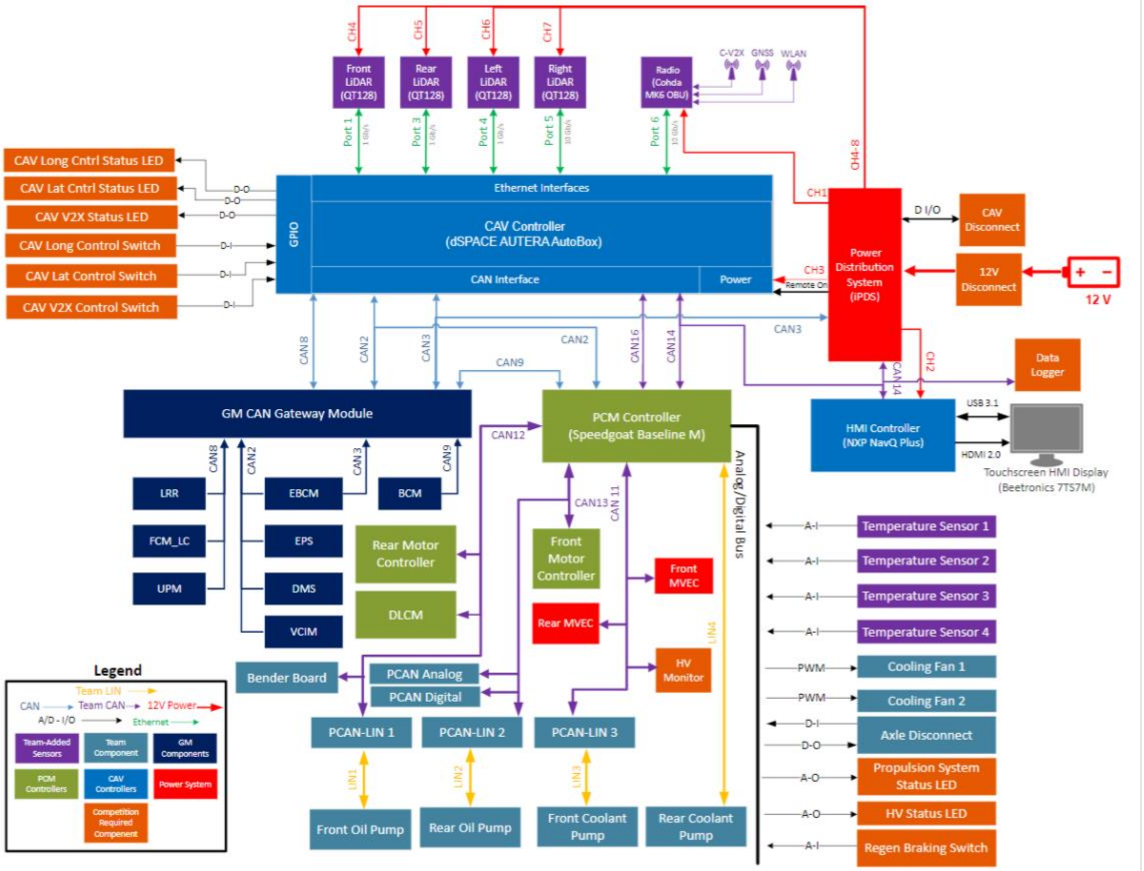


Figure 2: Serial Network Diagram

In the second year of the competition, the team focused on mechanically and electrically integrating the new components with the stock vehicle. In parallel, a controls strategy was developed for the electric power train. The controls strategy was developed using a model-based design approach. This approach utilized a plant model, which was originally given by MathWorks and was very heavily modified to accurately model the team's chosen architecture. The model was also adapted in order for it to be used in controls development in different testing environments. The plant model can be used for controls strategy testing in Model, Processor, and Hardware in the Loop Testing. This allows the control strategy to be stringently tested before it is applied in the vehicle. This workflow allows for safer and faster vehicle testing and will be used in years three and four of the competition to refine the controls strategy.

Project Objectives and Significance

This document aims to elaborate on the work completed during the first two years of the EcoCAR EV Challenge and details the process undertaken by the Propulsion, Controls, and Modeling (PCM) team. It begins with the architecture selection and proceeds to the design and development of the drivetrain plant model. Then, how soft ECUs were created to simulate communications with the stock system and team-added components. These soft ECUs and models were based on component documentation, past team knowledge, and test data from an extensive literature review. Next, it details how the plant model and controller were compartmentalized and integrated to create a plant model that can be used in different testing environments. This allows the controller to be tested in machine, software, and hardware in the loop environments before it is applied in the vehicle.

The plant model was applied using a model-based design approach. The development process followed a standard V-diagram process, which is standard in the automotive industry. The model is based on competition, safety, and functional requirements. The thesis details how the model was used to verify controller performance and functionality, including the workflow used by the PCM team. Further, an overview of the test cases run using the plant model to verify controller functionality is included. Moreover, the paper elaborates on how the results of the model and test cases were verified and validated and how requirements were generated and tracked. Finally, the thesis will discuss future work that can be applied to improve model accuracy and efficiency.

Confidential Information Disclaimer

This project was completed in partnership with General Motors who gifted The Ohio State with a 2023 Cadillac LYRIQ. Because of the nature of this work a great amount of confidential information was shared by General Motors for use in this project. Due to this some information will be excluded from this thesis and naming conventions will be given generic descriptions. The excluded confidential information from this document is not necessary for a complete understanding of the work and has been deemed not necessary.

Chapter 2. Literature Review

2.1 Introduction

Chapter two details the major portions of the literature review completed to enable the completion of this project. The literature review details the model-based design process and the ways in which it is executed. Further, the literature review contains the details of the market study completed by the team to develop vehicle technical specifications and guide vehicle design decisions. Finally, the literature review will cover different drivetrain modeling techniques in detail.

2.2 Model-based Design Process

2.2.1 Overview

In the current market, vehicles have become as much of a software product as a physical machine. Many major automotive manufacturers, including GM, have made a push for a “software-defined vehicle”. With such a software-heavy product, it is essential to have a well-defined and efficient process for development. The model-based design process allows for an inexpensive and fast development process, as most testing can be done at an accelerated pace on a desktop station through machine-in-the-loop testing (MIL). After controller feature functionality has been established through MIL testing, controller run time can be tested in a processor in the loop testing (PIL). Finally, real-time testing can be completed in hardware in-loop testing (HIL). HIL testing allows for software feature testing and signal processing testing. This ensures that the controller will run correctly when applied on the vehicle or in a vehicle in-the-loop testing (VIL). Finally, an important part of the model-based design process is generating, verifying, and tracking requirements. These requirements drive model development and are the deciding factor in test case creation. The development of requirements, model components, and test cases is also covered below. All of these steps can be summarized in the V-diagram diagram below.

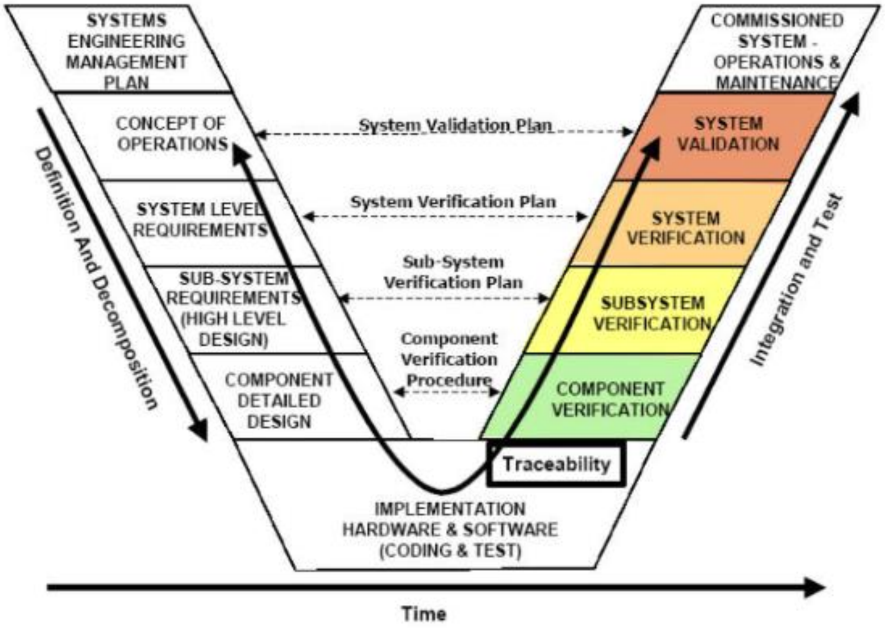


Figure 3: V – Diagram [2]

The V-software development process is standard in the automotive industry and guides model development during a controller’s model-based design process. The process can be split into two main portions: 1) definitions and design, 2) integration and testing.

2.2.2 Definition and Design

Requirement and Test Case Definition

Requirements are “a desired feature or behavior of a system” [3]. Functional and non-functional requirements drive the model design process. Functional requirements are generated using a process that takes into account business requirements and stakeholder interests. Business requirements are high-level goals in the V-diagram above. They can be compared to the concept of operation. The goal of these requirements is to set out high-level goals for the product. Next, non-functional requirements are constraints that are placed on the product or system. These requirements begin to form as the process moves down the left side of the v-diagram. Going from product to system and then to component-level requirements. Both of these sets of requirements come from stakeholders in the project. The main stakeholders in a project are consumers, suppliers, and external governing bodies. Figure 4 below shows a complete breakdown of stakeholders in a project.

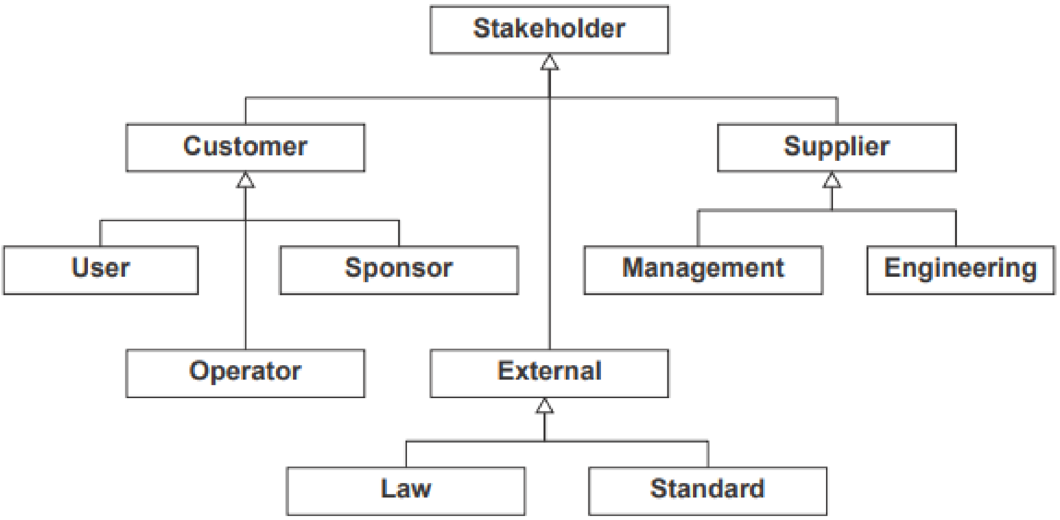


Figure 4: Stakeholder Chart [4]

As stated above, business requirements and stakeholder wants and needs all contribute to the development of functional requirements. Functional requirements define the necessary functionality of the software and can range from system level requirement all the way down to component level requirements of the software [4]. These requirements are present in all area of the v-diagram and define the test cases developed.

The test case process follows the inverse of the requirement development process. Test cases are first run for the software components. Then, as the system becomes more defined, system-level test cases can be run to properly ensure system functionality. Even though tests are run at the component level first, system-level test cases are defined at the start of the process. This is only possible because of the earlier requirement definitions. Following this process allows for an iterative design process that allows for easy software modification if higher-level requirements are not met after model development

Model Component Design

The model components fall within three major categories white, grey, and black box models. These models use different levels of physics based modeling and experimental data. Table 2 below details the main attributes of these models.

Table 2: Types of Models [5]

Model Name	Description
Black Box Model	This model has no physics based equations. It is based entirely off of experimental data. A good example of this is a look up table in a plant model. These are highly computationally efficient but are only as accurate as the experimental data collected.
Grey Box Model	This model contains physical parameter and physics based equations. The main difference between this and white box models is that all physical parameters are not know. So, some physical parameters are estimated based on experimental results in order to closely emulate system behavior. This is the mostly commonly used model in the plant.
White Box Model	This model is created from physics-based equations and parameters of the system.

Model Integration

In order to execute requirements testing and validation in different steps of the v-diagram, the model must be setup and integrated in a way that allows for easy switching between testing environments. This is done using model compartmentalization. This separates the input and output layers of the plant model and controller. An example of this can be seen applied to a hybrid supervisory controller in Figure 5 below. The interfaces use variant control in Simulink to switch between testing environments quickly.

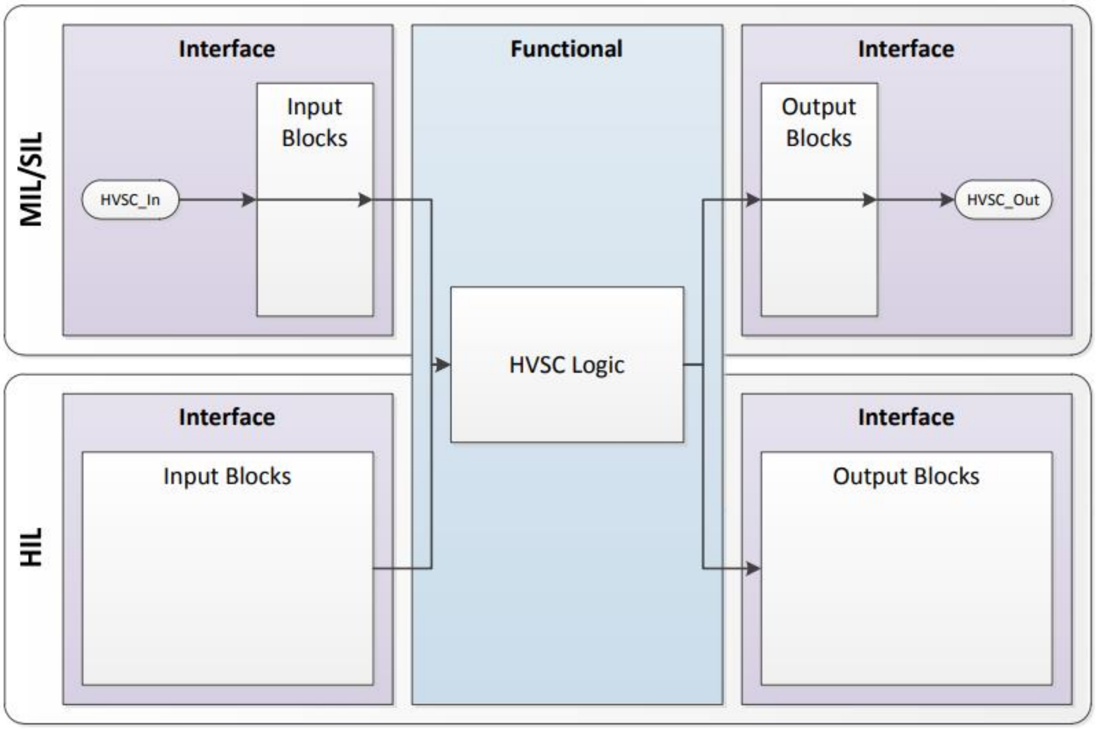


Figure 5: Plant Testing Environment Interface Variants [6]

2.2.3 Model Testing

As stated above, testing can be done in many different environments. Below are detailed testing goals in each environment and the process in which it is standardly done in the industry. A summary of each testing environment's goals can be seen in Table 3 below.

Table 3: Controller Functions and Testing Environments

Main Controller Function	MIL	PIL	HIL	VIL
I/O Signal Processing		X	X	X
I/O Fault Detection	X	X	X	
Torque Request	X	X	X	X
Mode Selection	X	X	X	X
Mode Operations Algorithm	X	X	X	X
Actuation Commands				X

Signal processing is only done in HIL and VIL environments because a physical connection is needed to ensure that the team’s hardware and signal processing algorithms are properly functioning. Input-output fault detection is tested in MIL, PIL, and HIL. The test cases in MIL and PIL are manually made in extreme cases where faults in signal values are given. For the HIL test cases, the fault detection for I/O is both extrema cases, and physically losing signals are tested with the hardware. Next, all of the overall algorithms, such as torque request, mode selections, and mode operation algorithms, are tested in every environment, to the extent to which each is tested will be discussed in Chapter 5. Finally, actuation commands are tested purely on VIL since actual components are needed.

Machine in the Loop

MIL testing is the most time efficient form of testing as the controller and plant can be run at an accelerated time on the same system. MIL testing can be done both on a system and component level. Component-level MIL testing allows for faster testing and component-level requirement verification throughout the design process. System-level MIL testing tests the functionality of the controller and is the first step in the system verification process. MIL testing can be automated and integrated with git to automatically test the system anytime a change is pushed into the repository. This allows for rapid iterations by a large team. Figure 6 below shows how the controller, plant, and driver are all encompassed in a single virtual environment.

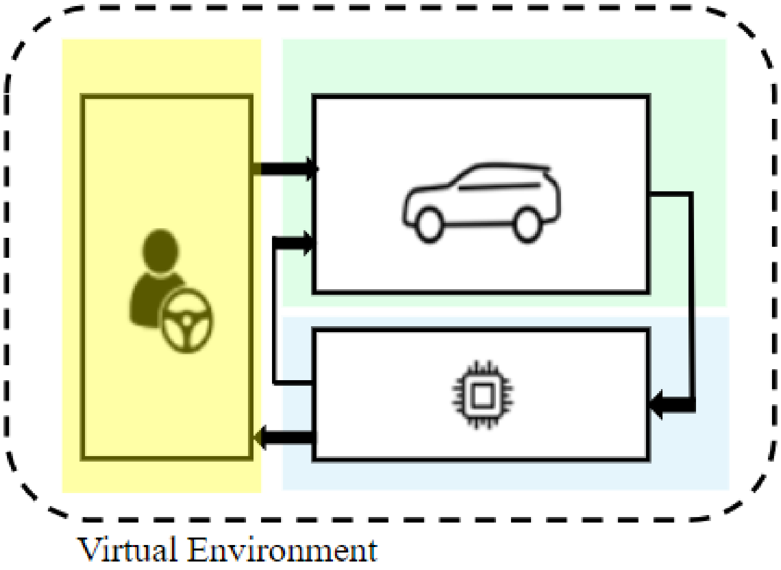


Figure 6: Machine in the Loop Overview

Processor in the Loop

PIL testing is the next step in the testing process. In this step the controller and plant model are compiled in C using an embedded coder or other code generation techniques. This software is then flashed onto the target hardware. Now the code will run at real-time. This testing phase, verifies that the software is functional at real-time and does not lose functionality when compiled on to the target machine. Figure 7 below shows an overview of the PIL system.

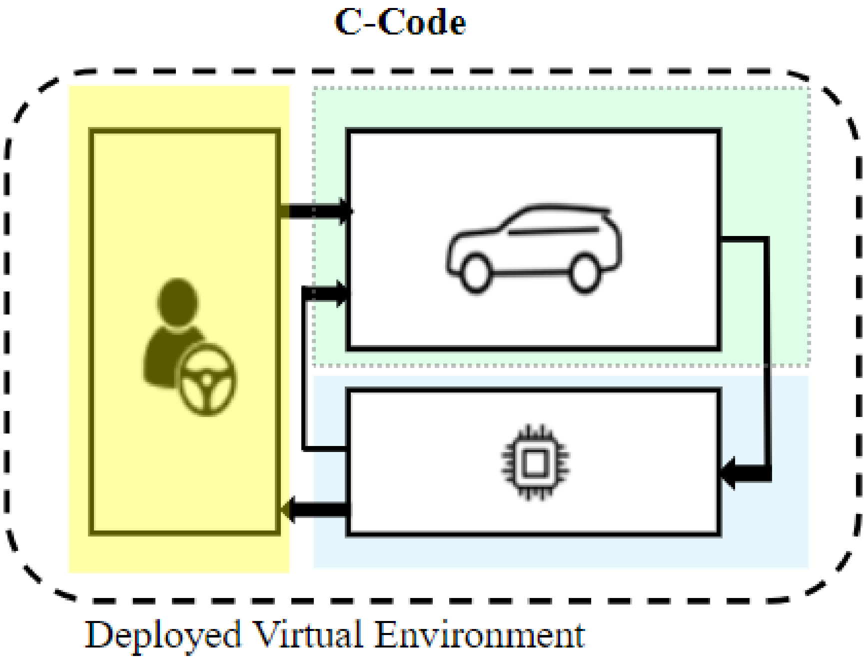


Figure 7: Processor in the Loop Overview

Hardware in the Loop

HIL testing is the final step before the code is greenlit to be tested on the vehicle. In HIL testing, the plant model and controller are run simultaneously on two different pieces of hardware. The models are connected with CAN, LIN, and A/D interfaces. Controller functionality, processing time, and signal processing can all be tested in this environment. Further controller reactions to physical faults, such as loss of communications or incorrect signals, can be tested. Adding another layer of safety and verification before the code is ever placed on the vehicle. Figure 8 below shows a representation of the HIL system.

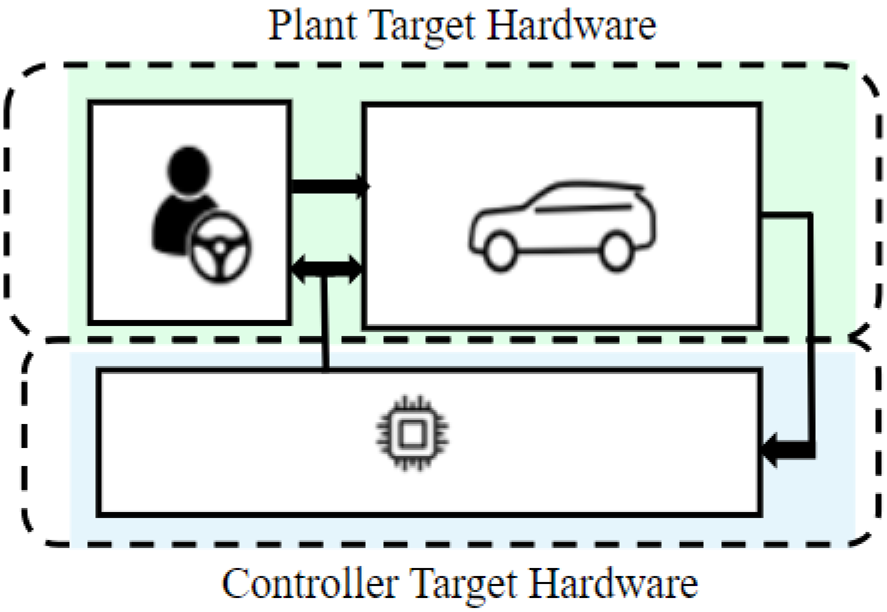


Figure 8: Hardware in the Loop Overview

Vehicle in the Loop

VIL testing is the last step in testing. The main goal of VIL testing is to calibrate already made software features. The above methods of testing are all based on a model that will be inherently flawed no matter how much time and effort was put into accurately developing it. VIL testing allows for these small software changes to be made and ensures a robust code. Figure 9 below shows a common VIL setup.

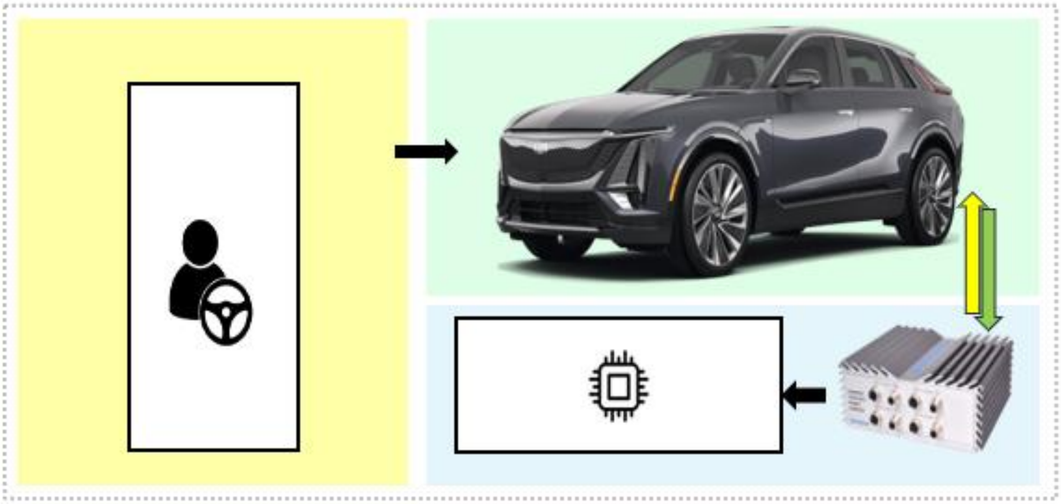


Figure 9: Vehicle in the Loop Overview

2.3 EV Drivetrain Architectures

To assess these objectives, the team completed a market research study to view the current state of the ESUV market, develop vehicle technical specifications, and compare the team’s architectures to the market. The team researched the ESUV market and emphasized vehicles with around the same mass as the Cadillac LYRIQ (± 200 kg) and referring it to as the focus market. The ESUVs included in the market research study can be seen in Appendix A.

2.3.1 Summary of ESUV Market

The ESUV market can be broken into two categories front/rear wheel drive vehicles (FWD/RWD) and all-wheel drive (AWD) vehicles. Most of these vehicles use synchronous permanent magnet (SPM) motors. A few of the AWD vehicles use an induction motor (IM) in parallel with an SPM. A limited number of vehicles offer three or four motor configurations, which are produced by Tesla, Audi, and Rivian. These options have a high-power output when compared to most single and dual motor configurations and have increased performance metrics. Because of the limited number of production vehicles with these characteristics, they were grouped in the AWD portion of this market. Table 4 summarizes the market average specifications and performance metrics of different ESUV architectures, the stock RWD LYRIQ, and vehicles with a mass like the LYRIQ.

Table 4: Market Research Summary

Drivetrain	# of Sampled Vehicles	Total Power [kW]	% Of Power in Rear	Mass [kg]	Battery Capacity [kWh]	0 – 60 [s]	Range [mi]	City [Wh/mi] Unadj	HW [Wh/mi] Unadj	Comb [Wh/mi] Unadj
LYRIQ RWD	NA	250	100%	2626	102	5.7	312	243	288	268
RWD	19	182	100%	2036	78	6.5	267	213	250	230
FWD	3	150	0%	1860	66	6.0	253	184	223	201
AWD	42	374	54%	2395	88	4.5	267	258	278	267
Entire Market	64	306	65%	2270	84	5.1	266	241	268	253
Focus Market	**17**	**397**	**45%**	**2600**	**94**	**4.5**	**281**	**280**	**284**	**282**

The averages of the entire market are important to note; however, the market has a vast range of vehicles that have different design goals. To help narrow down the scope of the market study, the team decided to focus on vehicles that have a similar mass to the RWD LYRIQ. Figure 10 shows the vehicle mass effect on energy consumption and the team’s focus market. From the research, it can be observed that most vehicles over 2400 kg or with a power rating over 275 kW are AWD to distribute the power and improve the handling characteristics. This is further shown in Figure 11, which compares the vehicle's power to acceleration times.

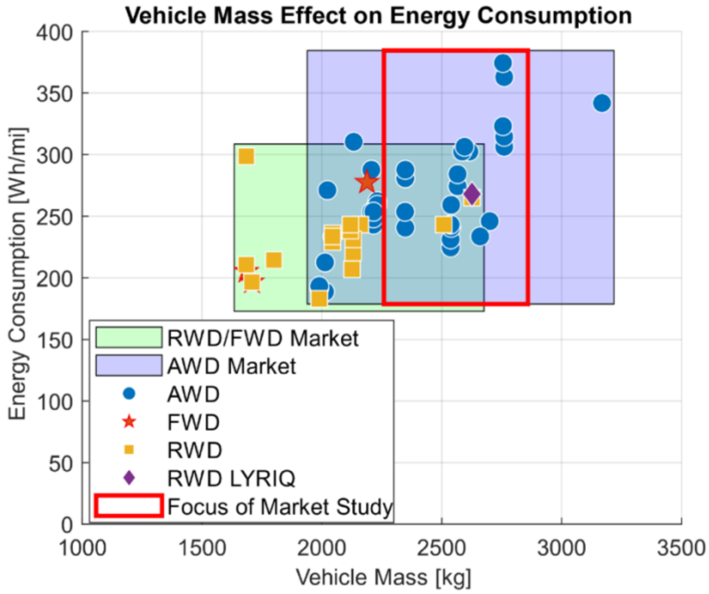


Figure 10: Market Research Energy Consumption

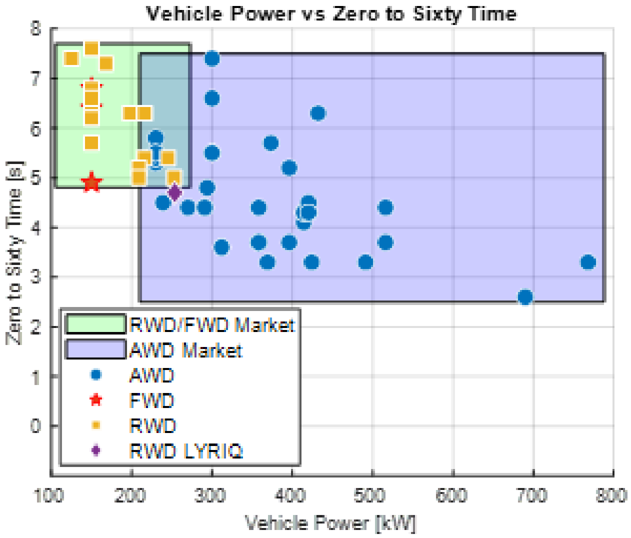


Figure 11: Market Research Acceleration Performance

2.3.2 Focus Segment Market Study

The focus segment market study is made up of 17 vehicles. The majority of these 17 vehicles are different iterations of the Mercedes EQ450+, Tesla Model X, BMW iX xDrive, and Audi e-Tron. These vehicles are classified as high-end luxury ESUVs placing them in the exact same market as the Cadillac LYRIQ. In addition to the similar mass, these vehicles are roughly the same size as the Cadillac LYRIQ.

Segment Market Vehicle Architectures

Fifteen of the seventeen vehicles in the segment market are dual motor AWD drive architectures. The remaining two are RWD architectures. The reason most of these vehicles are AWD is because of the larger size of the vehicles. The two RWD architectures use a single motor of around 250 kW, like the stock RWD Cadillac LYRIQ. Furthermore, all these vehicles have an above average battery capacity. In the segment market, there is no correlation between battery capacity and vehicle mass. The vehicle's battery pack capacity is limited by the volumetric size of the battery or by the manufacturer to reduce vehicle costs. Furthermore, the power of these vehicles ranges from 230 kW to 500 kW. From the vehicle power, drivetrain, and battery capacity, it can be concluded that the objectives of the vehicles in the segment market are to provide the consumer with a vehicle with quick accelerations, an above market range, and a large interior. The objectives of the market segment align with the objectives of the OSU-WU team.

Segment Market Longitudinal Acceleration and Energy Consumption

Figure 12 below equates the zero to sixty of each vehicle to the power to mass ratio. To have a vehicle with best-in-class longitudinal acceleration performance, the team expects the vehicle to have a zero to sixty time of 4.5 seconds or less. Figure 13 below shows how energy consumption and vehicle mass are proportionally related. Because of this, the team is expecting higher energy consumption from their proposed architectures because the overall mass of the architectures will be greater than the market average. The team will have designed an energy efficient vehicle that is competitive in the market if the proposed architectures fall below the line of best fit in Figure 13.

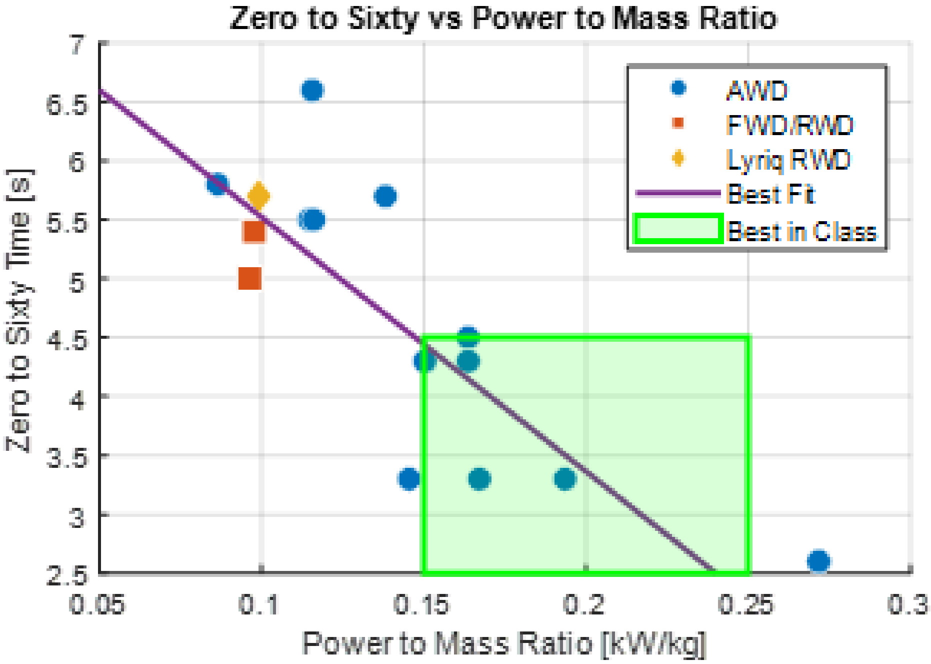


Figure 12: Target Market Power to Mass Ratio vs Zero to Sixty

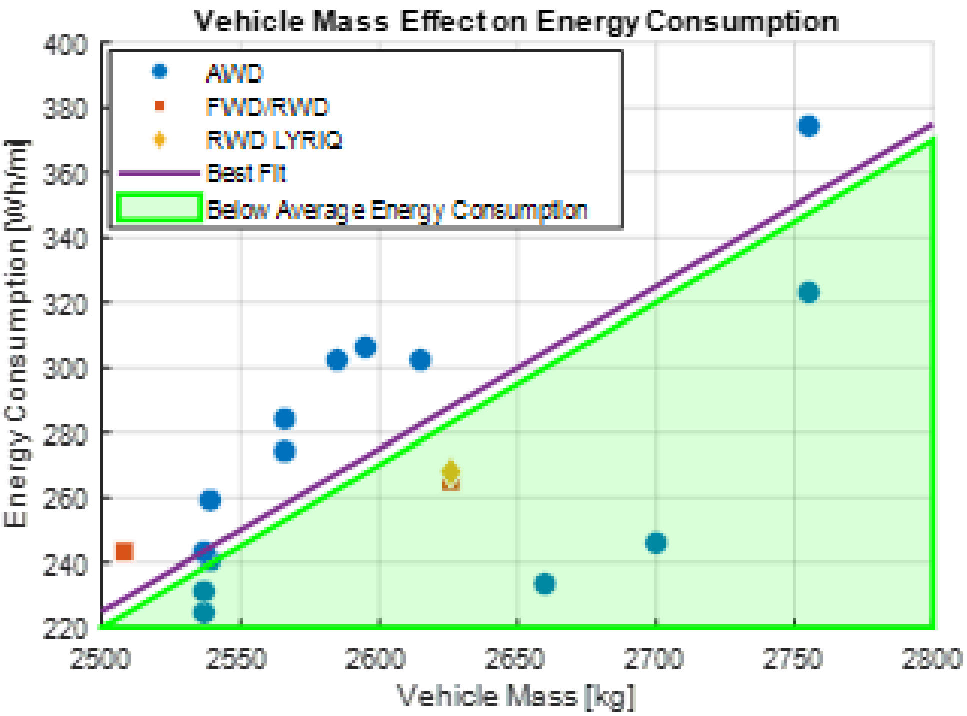


Figure 13: Target Market Energy Consumption vs Mass

Range

Closely related to energy consumption is range. The main factors determining an ESUV's range are battery capacity and energy consumption. The segment market has an above average range (281 miles) and battery capacity (94 kWh) compared to the entire ESUV market (266 miles, 84 kWh). Since the competition constrains the teams to use only 80% of LYRIQ's battery capacity, this must be factored into the market average. The realistic segment market average range is now 225 miles.

Segment Market Conclusion

In conclusion, the segment market study provides valuable insights into the design and performance characteristics of high-end luxury ESUVs. The market is dominated by vehicles with dual motor AWD drive architectures, high battery capacities, and power ranging from 230 kW to 500 kW. These vehicles aim to provide consumers with quick acceleration, an above-market range, and spacious interiors. The OSU-WU team developed objectives aligned with those of the market segment, as they aimed to design a vehicle with best-in-class longitudinal acceleration performance while maintaining energy efficiency. Further, the team expected higher energy consumption due to the overall mass of their proposed architectures but still strived to stay competitive in the market by falling below the line of best fit in the energy consumption and vehicle mass relationship.

Market Research, Team Objectives, and the Development of Team VTS

The above market research and the stock RWD LYRIQ specifications shaped the team's objectives and the VTS. The team expected the vehicle to be competitive in the segment market and to be competitive in specific categories within the overall market. The vehicle's range needed to be above the market average to make the LYRIQ competitive in the segment market. However, the vehicles' energy efficiency should have been below the overall market average because of the larger size and mass of the LYRIQ. Ultimately, the team's objective for acceleration performance was for the vehicle to be at the market average of the eSUV market, meaning the proposed acceleration VTS is well below the average of the segment luxury eSUV market.

Chapter 3. Vehicle Architecture Selection

3.1 Introduction

This section describes the vehicle architecture selection process taken by the OSU-WU PCM team. First, a description of the process will be given, and then each section will be expanded upon in the following subsections.

3.2 Architecture Selection Process

The architecture selection process was a five phase process. An overview and the flow of these five processes can be seen below in Figure 14 and Table 5 below.

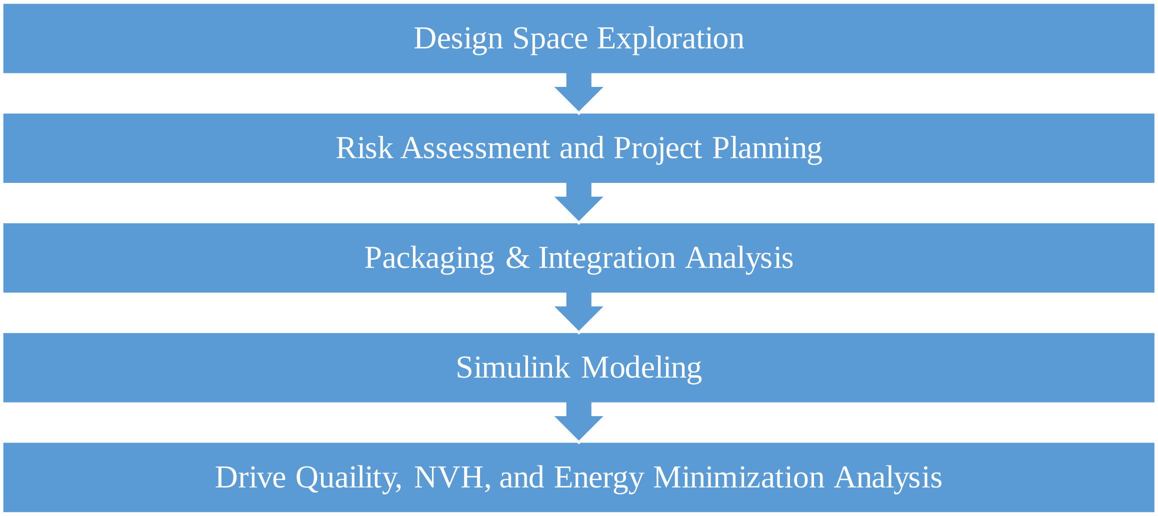


Figure 14: Architecture Selection Process Chart

Table 5: Architecture Selection Process Description

Phase	Tool	Results	Testing Criteria	Design Space
Phase 1 Design Space Exploration	MATLAB script to analyze energy required for UDDS and HWFET cycle with a minimized energy consumption strategy	All possible architecture combinations with FWD, RWD, and AWD configurations and a range of gear ratios (appendix for combinations)	Architectures that are within the team's VTS targets (e.g., eliminate architectures with IVM-60 times greater than 6s)	Tested: 151,080 Feasible: 4,000
Phase 2 Risk Assessment and Project Planning	Timeline analysis and risk matrix analysis	Architectures that are achievable in the competition timeline and components that are acquirable	Projects under the team's risk appetite and components that are available or manufacture-able	Tested: 4,000 Feasible: 250
Phase 3 Packaging & Integration Analysis	NX and space claim	Architectures that are feasible with respect to packaging space on the vehicle	Front and rear cradle modification and stock component relocation requirement	Tested: 250 Feasible: 175
Phase 4 Simulink Modeling	Simulink model with pre-computed energy split strategy	Narrowed down the combinations to most desirable with respect to competitive VTS targets, integrate-ability, and controllability	VTS targets test cases, modeling validity like speed trace error and current limits	Tested: 175 Feasible: 5
Phase 5 Simulink NVH and Drive Quality Modeling	Hamiltonian minimization strategy, along with team added backlash compensator and disconnect controller.	The best architectures were tested with added controller features to account for drive quality torque smoothing and, disconnect delays & speed syncing, relative comparison to real life data.	VTS targets test cases, and validity should be passed with the compensation models.	Final Selection

3.3 Phase 1: Design Space Exploration

The design space exploration was a sweep through of over 150,000 electric vehicle architectures. Each architecture's energy efficiency and zero to sixty mph acceleration were tested in order to narrow down the possible options. This was done using a physics-based model that was developed using a Matlab script and data from component suppliers. Before the start of the design space exploration, there were already constraints on the design based on budget and project planning factors. The major constraint was that architecture with more than two motors would be too costly for the team and would require too much integration work to get done within the project timeline. Because of these, only architectures with two or fewer motors were considered. Table 6 below details all the architecture combinations considered. An architecture was determined to be any unique combination of electric motors, transmissions, transmission gear ratios, and vehicles with an electric motor/axle disconnect feature.

Table 6: Tested Architecture Breakdown

Architecture Layout	Gear Combinations	Motor Combinations	Total Architectures
AWD (2 Speed Gearbox or Disconnect on one axle)	376	157	59,032
AWD (Fixed Gear)	144	157	22,608
AWD (2 Speed Gearbox on front and rear Axles)	434	157	68,138
RWD/FWD (Fixed / 2 Speed Gearbox)	93	14	1,302
Total			151,080

3.2.1 Simulation Overview

Energy Consumption

Each architectures energy consumption was evaluated using the EPA urban and highway driving cycle. An equation estimated the necessary torque at the wheel at each second of the drive cycle was used. The equation can be seen below. Where A, B, and C are constants in the road load equation; v is the vehicle velocity, m is the vehicle mass, a is the vehicles acceleration, and R is the radius of the wheel and tire.

$$T_{wheel} = (A + Bv + Cv^2 + m * a\ ) * R$$

Equation 1[8]

Next based off of the drive cycle trace the motor speed was estimated from these and other inputs the power consumption at every second of the drive cycle was calculated and integrated to find the overall energy usage over the cycle. The block diagram below illustrated the inputs and outputs of the defined Matlab function. To simplify the function, the disconnect was modeled as a gear box with a neutral gear which applied no load to the motor.

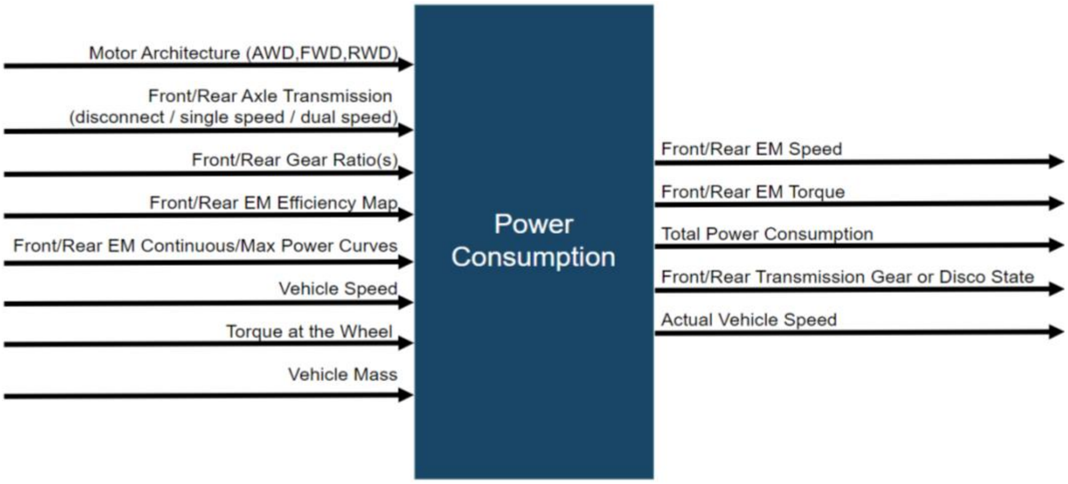


Figure 15: Power Consumption Calculation I/O

The torque split and shifting/EM engagement strategy for each architecture used the same function. The goal of the function is to maximize energy efficiency. However, penalties were introduced to avoid constant operation outside of the electric motors' continuous operating zone and to limit shifting or disconnect/reconnect events. An overview of the function can be seen in Figure 16 below.

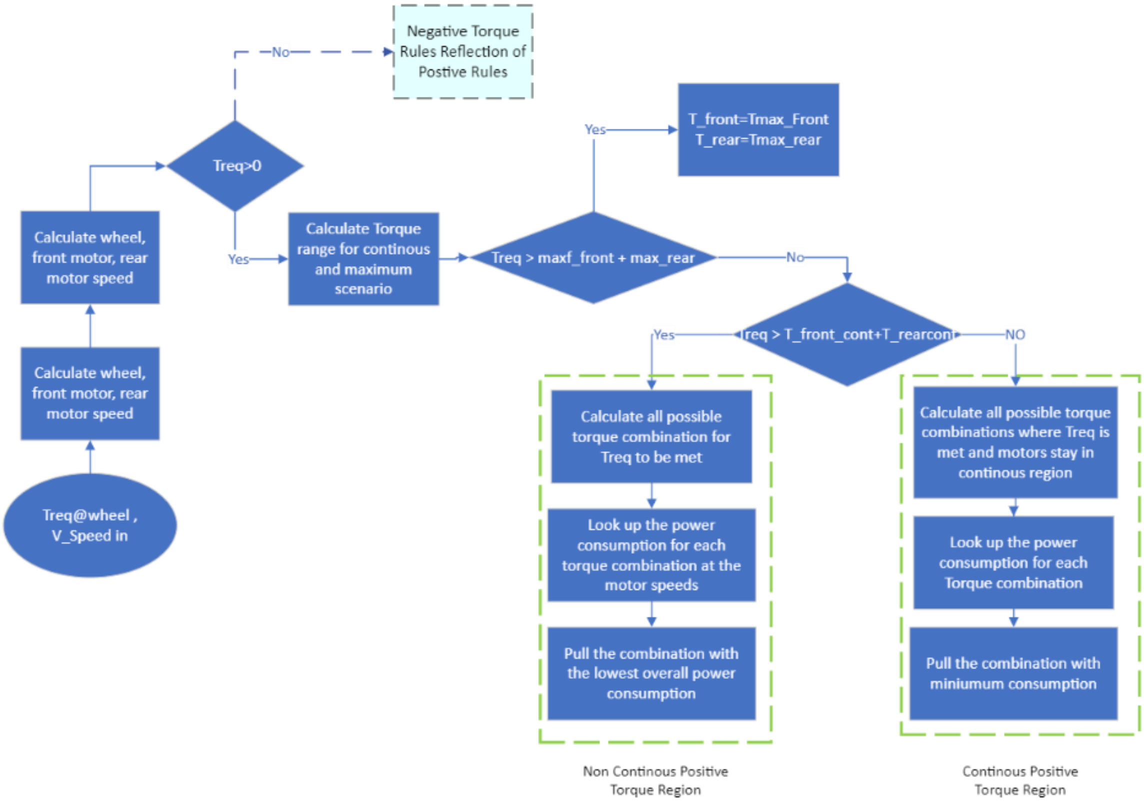


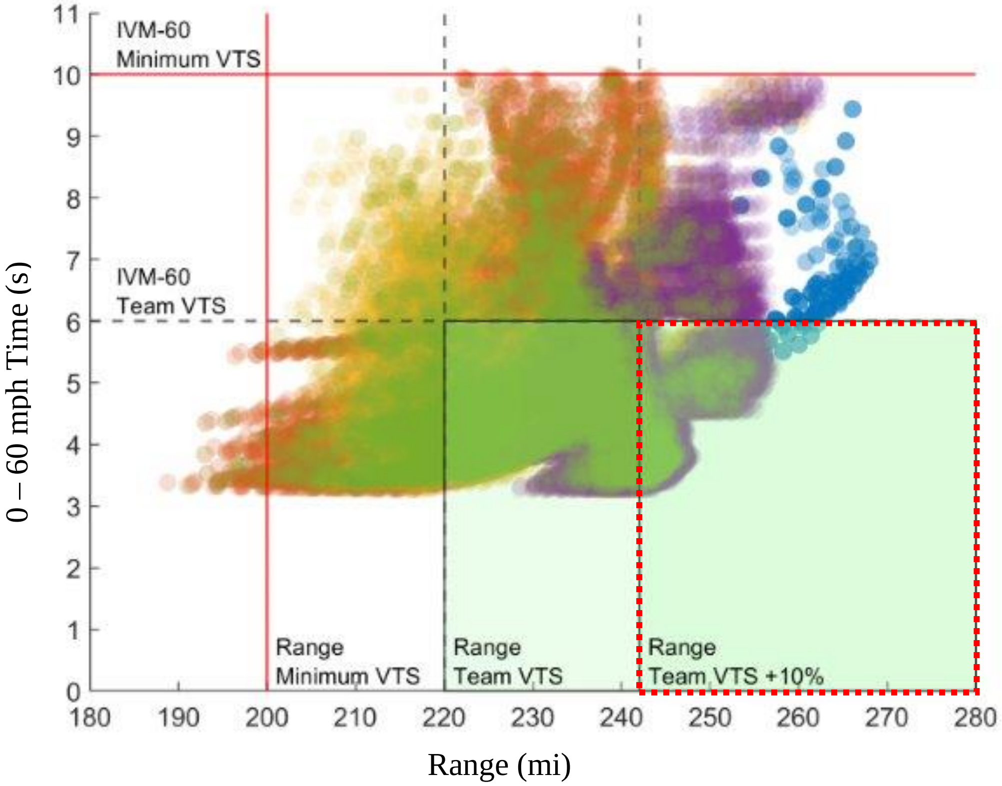
Figure 16: Torque Split Algorithm Flow Chart

Zero to Sixty Acceleration

The zero to sixty acceleration uses the same torque split strategy and function above. The only difference in inputs is the torque at the wheel, which is not estimated based on a drive cycle trace. A torque curve was developed for the zero to sixty test. This torque curve emulates the maximum torque at the wheel the vehicle can experience while keeping traction to the road. This torque curve allows the motors to either operate at their maximum torques or at the maximum torque, which would keep the wheels from slipping. Further, the model step time was reduced to 0.01 seconds for increased accuracy. With these changes, the vehicle's zero to sixty time can be estimated.

3.2.2 Results & Conclusions

Figure 17 below shows a summary of the results obtained from architecture exploration simulations. In Figure 17 it can be seen where the team VTS targets are marked, range and zero to sixty time. The range is directly related to the vehicle's energy efficiency since the battery pack capacity is fixed. The architectures highlighted in the red box were considered moving forward. In order to compensate for the lack of fidelity in the energy consumption model (lack of accessory load), architectures that were 10% above the team requirement for range were considered moving forward.



🟦 Single Motor – FWD and RWD
🟧 Dual Motor – AWD – Single Speed Transmissions
🟨 Dual Motor – AWD – Single and Dual Speed Transmission
🟪 Dual Motor – AWD – Two Dual Speed Transmissions
🟩 Dual Motor – AWD – Single Speed Transmission and Disconnect on one Axle

Figure 17: Architecture Sweep Results Summary

3.2.3 Model Confidence and Accuracy

The overall fidelity of this model is low. However, it provides a computationally inexpensive method to fairly compare a large number of vehicle architectures and determine relationships and patterns between similar architectures. Motor efficiency maps were obtained from suppliers or were modeled using experimental data and the Williams Line Model estimation [9]. Using this and other industry standard estimation practices does allow for fairly accurate comparisons. Further, because all architecture have all the same assumptions the error for all simulations is similar.

3.4 Phase 2 & 3: Risk Assessment, Project Planning, and Integration Analysis

This section of the architecture selection process was carried out mainly by the project manage, electrical leads, and mechanical leads on the project. Based on the architectures narrowed down in the above portion the leads contacted suppliers to receive project timeline information, detailed dimensions, and electrical information. Using this information the team excluded the architecture which would not be executable based on external constrains. The main takeaways from these phases are listed below.

- Qualified single motor architectures are not acquirable.
- 2-Speed gearboxes with required ratios not available in the project timeline.
- Dual motors from same manufacturers are practical and advantageous for integration.
- Larger electric machines were out of bounds in the workable region.
- Some drive units not compatible with cradle modifications restrictions.
- Thermal components placements and restrictions limited the use of some drive units.

These takeaways helped narrow down the plausible architectures to 175.

3.5 Phase 4 and 5: Simulink Modeling

Now that the architecture are narrowed down to a smaller number. More detailed modeling can be completed within a reasonable time frame. First acceleration, fuel efficiency, grade-ability, and braking tests were run for all remaining 175 architectures. After these simulations the architectures were narrowed down to five. Once down to five more detailed energy efficiency and acceleration simulations were completed. Also vehicle drive quality and NVH were estimated in other simulations. This was done with added controller features to account for drive quality torque smoothing and disconnect delays and speed syncing using a relative comparison to real life data. Each controller was custom tuned for each of the five remaining architectures.

3.5.1 Modeling Test Cases

Various test cases were considered while simulating the architectures. The team's VTS targets were translated into software requirements and test cases to verify these requirements. These test cases were followed in various forms through the phases. In phase 1, the VTS targets were hard cut-off and the same targets in phases 4 and 5 were implemented as automated test cases. Here are the high-level tests that determined the team's direction towards the proposed architectures.

Simulation Validity

Each simulation that is performed for an architecture is checked for its validity. In any simulation where the virtual vehicle is tracing an EPA rated cycle, it is subjected to EPA dynamometer driving schedule allowance. The EPA allowance states that the vehicle should stay within the upper limit of 2 mph higher than the highest point on the trace within 1 second of the given time and the lower limit of 2 mph lower than the lowest point on the trace within 1 second of the given time. Along with the speed trace, the electric machine’s DC current is verified with the maximum (datasheet specified) current limit of the component. The battery current is also verified with the GM specified current limits. This test case fails if all combinations of the above test condition fail. This ultimately voids the results of the simulation.

UDDS Energy Consumption

This test case calculates the energy consumption of the simulation in an urban driving condition on the EPA specified cycle. The Urban Dynamometer Driving Schedule (UDDS) cycle is 7.45 miles long and lasts 1369 seconds. This test case calculates the energy consumed over the drive cycle (DC kWh) and with additional data processing, checks if the architecture meets the team’s VTS maximum allowable 310 Wh/mi. However, this is not a failing condition as the combined consumption is used to determine the expected range of the vehicle.

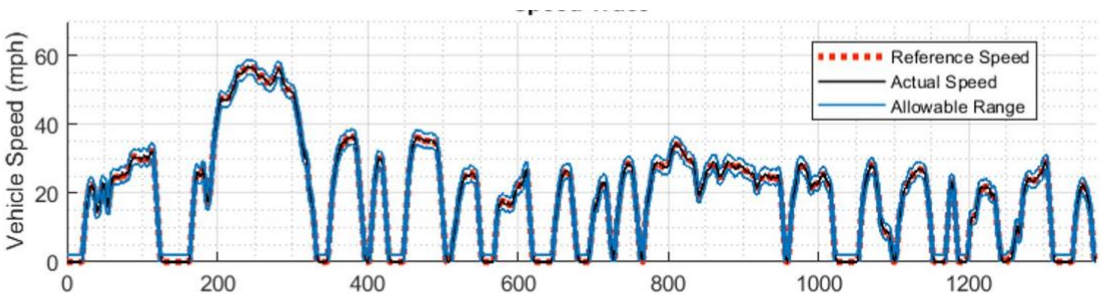


Figure 18: Urban Drive Cycle Speed Trace

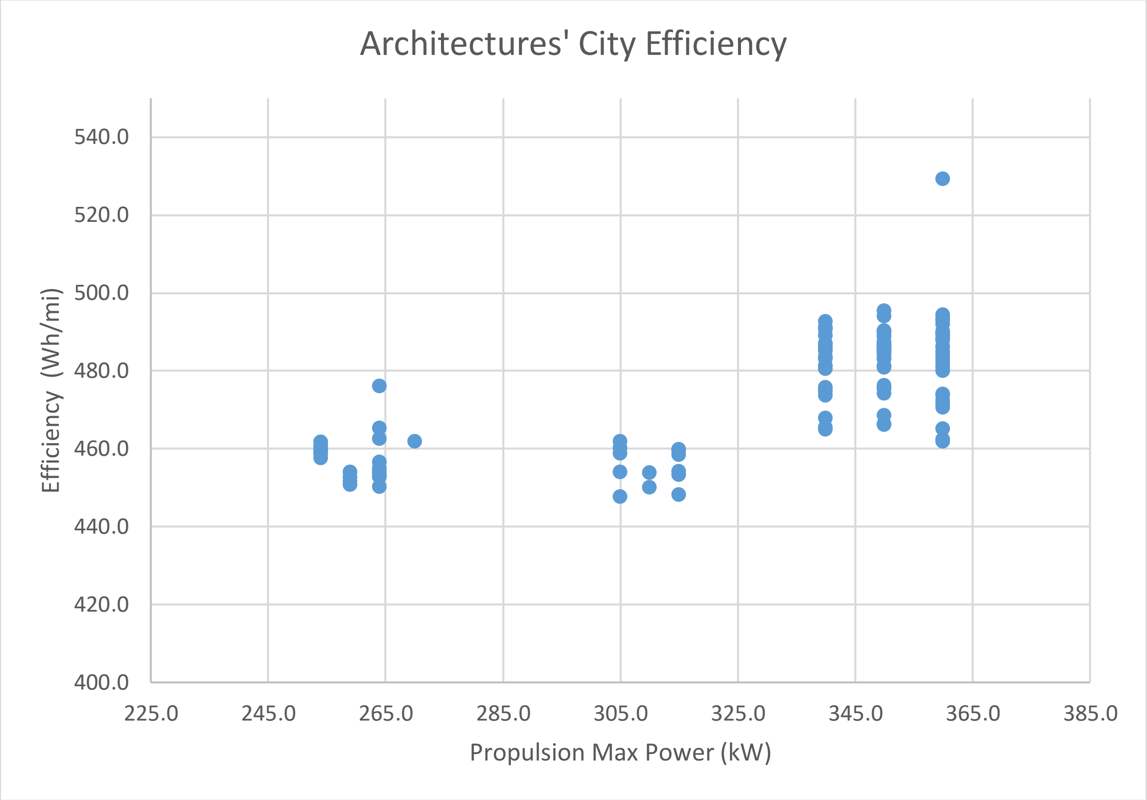


Figure 19: Top Architectures City Efficiency

HWFET Energy Consumption

This test case calculates the energy consumption of the simulation in an urban driving condition on the EPA specified cycle. The Highway Fuel Economy Driving Schedule (HWFET) cycle is 10.26 miles long and lasts 765 seconds. This test case calculates the energy consumed and checks if the architecture meets the team’s VTS maximum allowable 325 Wh/mi. A comparison of all top architectures is represented in Figure 20. The segment market is compared along with the acceptable team VTS target on both drive cycles.

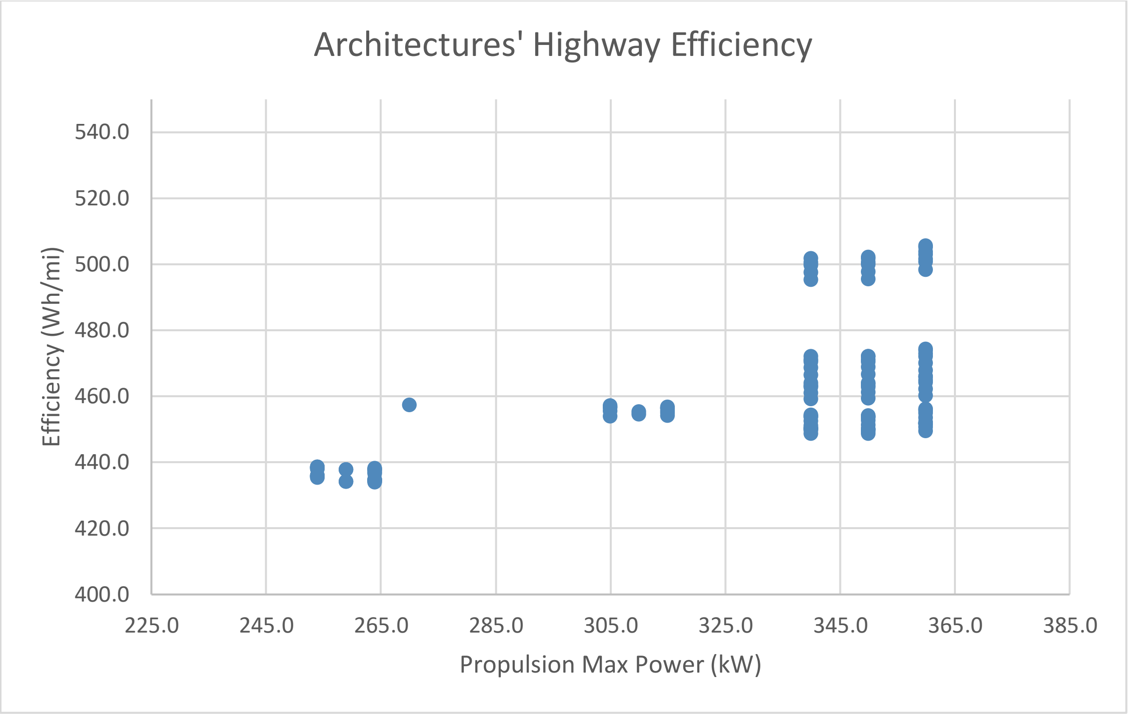


Figure 20: Top Architectures Highway Efficiency

Performance (Acceleration and Braking)

One of the consolidated test cases to verify three VTS targets is a custom-made drive cycle. First, in the drive cycle, a zero to sixty acceleration is performed. The test case calculates the zero to sixty time. This metric should be under the team VTS target of six seconds. Further, the vehicle is slowed to a steady fifty mph, and a fifty to seventy mph test is performed at thirty-five seconds. Similarly, the test case computes the time from initial acceleration till the speed of the vehicle is greater or equal to seventy mph. In the end, the simulation is subjected to hard braking from sixty mph at sixty seconds. The distance accumulated between the start of deceleration and the vehicle coming to a stop is the braking distance. The test checks if this distance is less than the maximum allowable fifty-two meters. A drive trace of the test case can be seen in Figure 21 below.

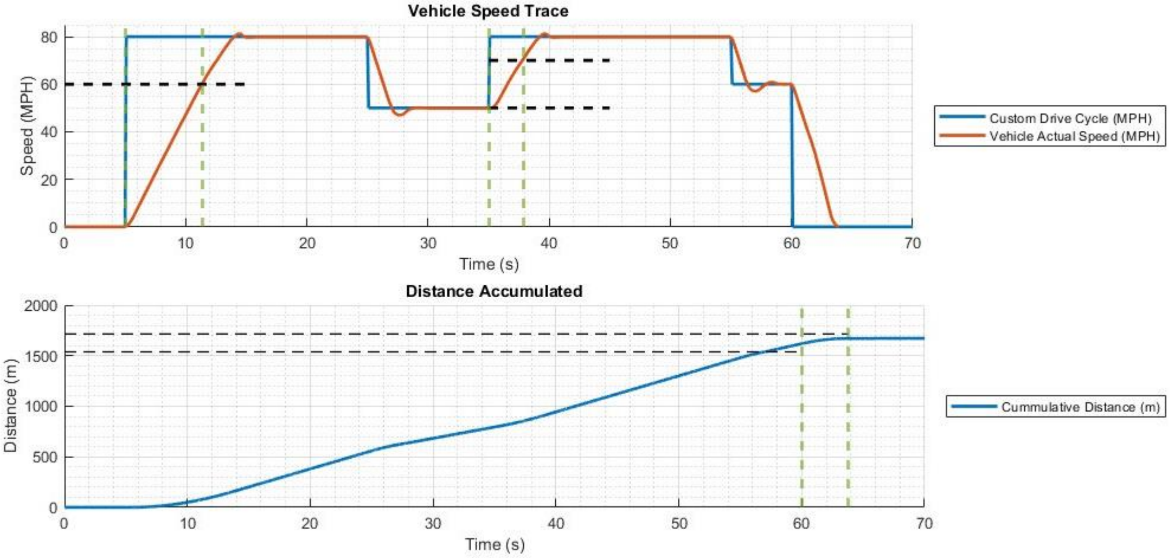


Figure 21: Acceleration Performance Drive Cycle Trace

Further, the results for the zero to sixty and fifty to seventy mph test can be seen in Figure 22-25 below. As expected, acceleration performance increases as motor power and torque at the wheel increase. However, the relationship is asymptotical as acceleration performance does seem to peak out at around four seconds for the zero to sixty, and the fifty to seventy acceleration peaks out at a little over two seconds. This begins to happen because the vehicles are reaching their traction limits.

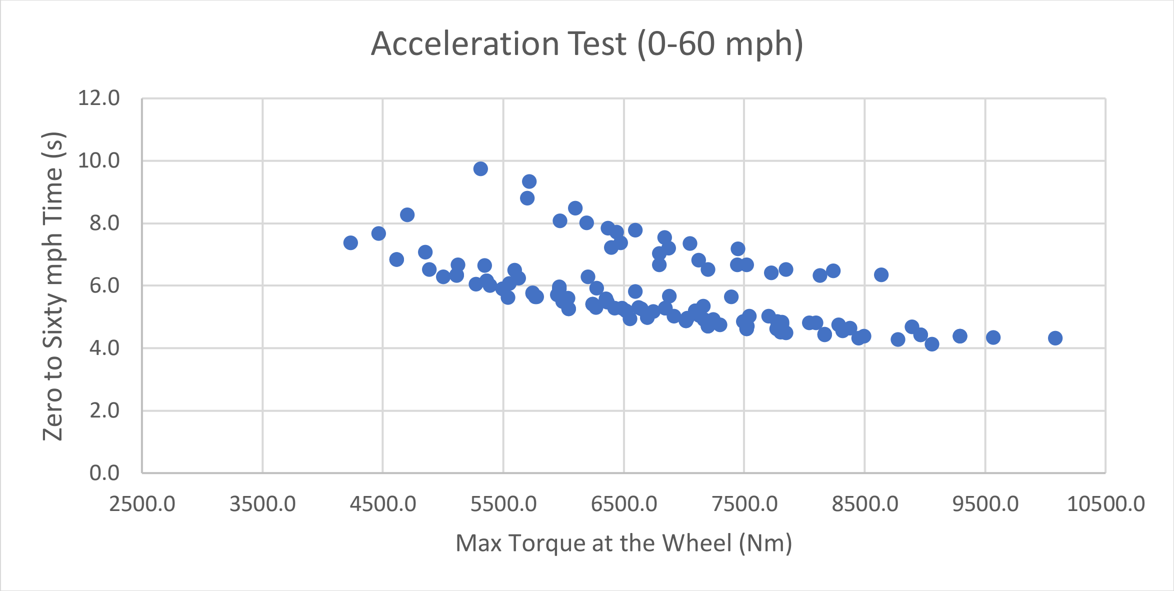


Figure 22: Zero to Sixty Time vs Vehicle Max Wheel Torque

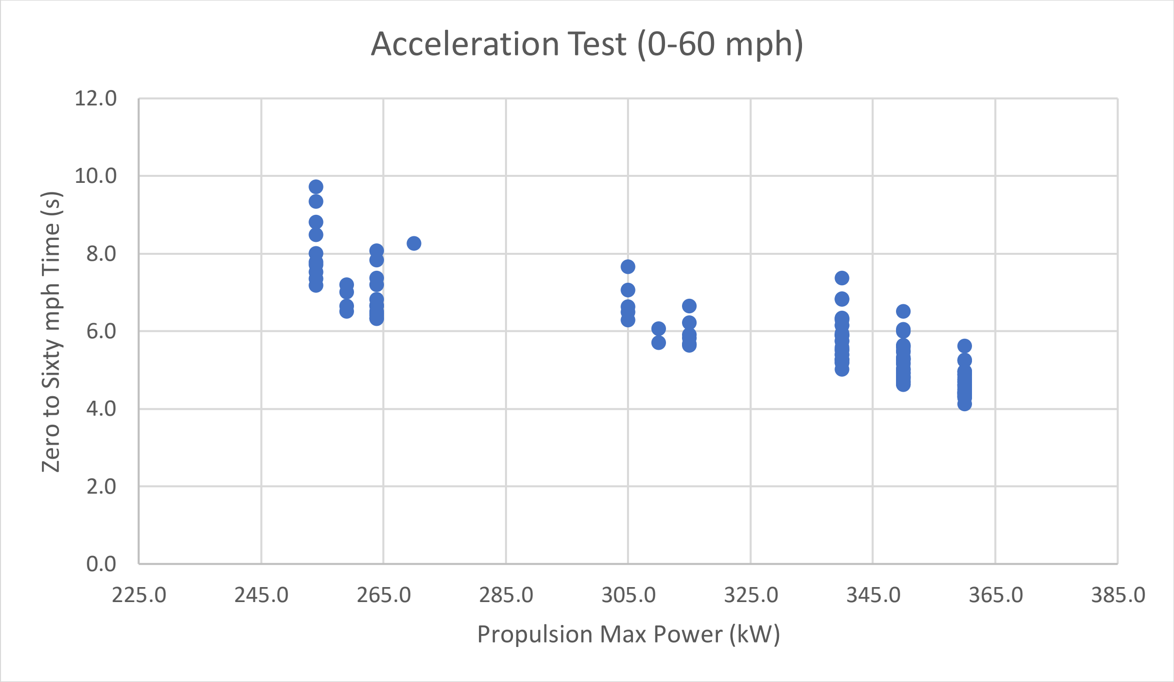


Figure 23: Zero to Sixty Time vs Vehicle Max Power

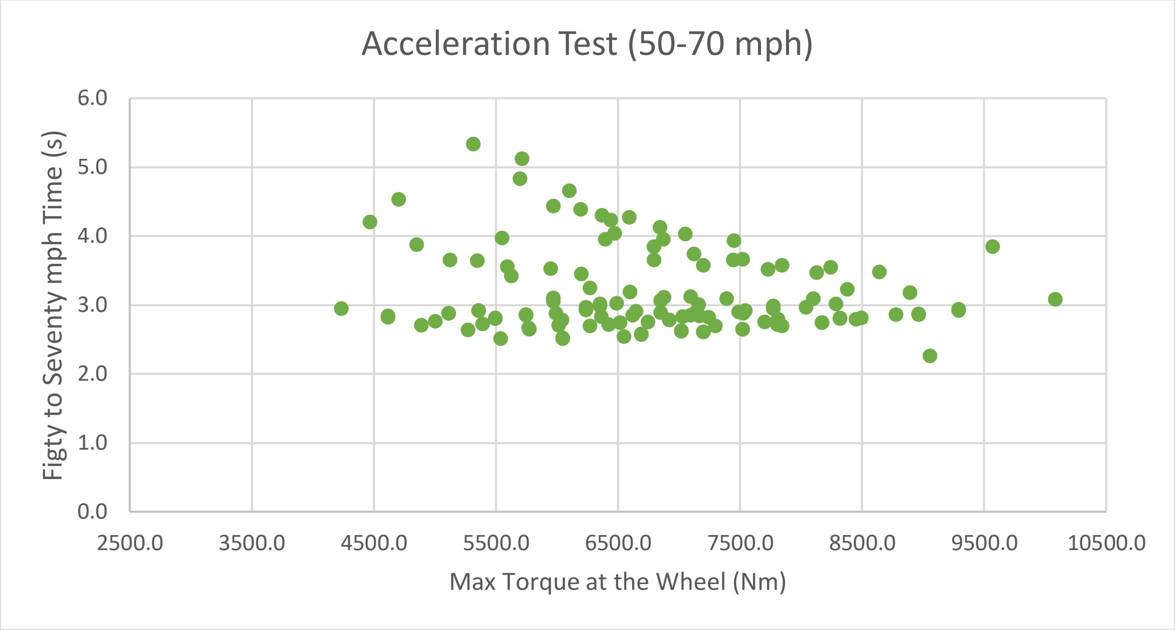


Figure 24: Fifty to Seventy Time vs Vehicle Max Wheel Torque

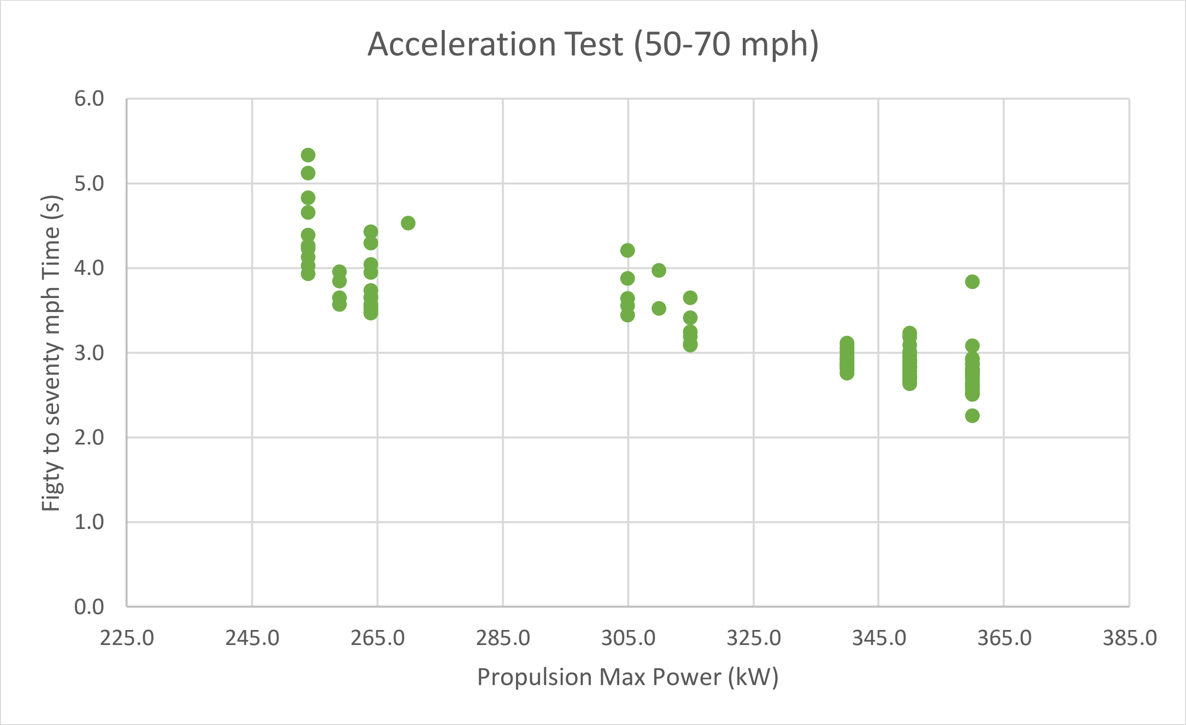


Figure 25: Fifty to Seventy Time vs Vehicle Max Power

Grade-ability Test

One of the test cases is to verify whether architecture can operate in a safe continuous zone while performing a gradient ascent. The condition for the test is to accelerate the vehicle to highway speed, which is sixty-five mph, and perform a constant speed drive for three hundred seconds. Here, the two passing conditions are to allow the motor to operate in the peak region for ten seconds or less and then operate in the continuous region for the remaining three hundred seconds. This complete test is performed on a grade of six percent, which is an inclination angle of 3.43 degrees. This test case is treated as a pass/fail test to eliminate options that had to operate in the peak regions to perform regular highway driving. Further, the maximum possible grade at sixty-five mph is tested by repeating this test with an incrementally increasing grade till the vehicle cannot trace within one mph of the reference speed.

Disconnect Logic NVH

In phase five of architecture simulation and testing, a few of the architectures were equipped with a disconnect clutch to control the torque flow on one axle. This, in theory, should reduce driveline resistance and improve the overall efficiency of the drive cycle. This testing process is conducted on the model modified using an axle disconnect, which is modeled using a gearbox with a 1:1 drive gear and a neutral gear. To actuate this disconnect system, a rudimentary control logic was implemented. This control logic can be seen below in Figure 26.

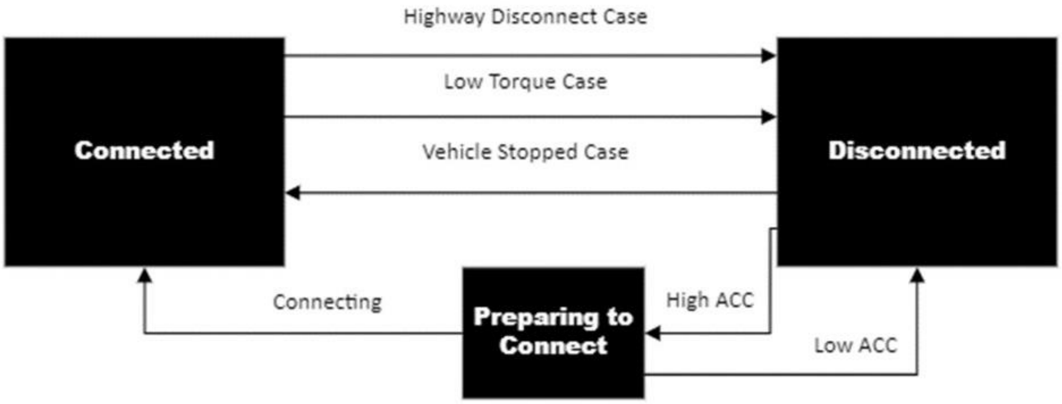


Figure 26: High-Level Disconnect Logic

The test case is to check the change in energy consumption and the driveline jerk estimated from the model. The passing condition is to have longitudinal jerk spikes to be less than +/- 4 m/s^3 and disconnect/reconnect events to happen less 25 times an urban drive cycle**.** An example of an acceptable drive cycle case can be seen in Figure 27 below.

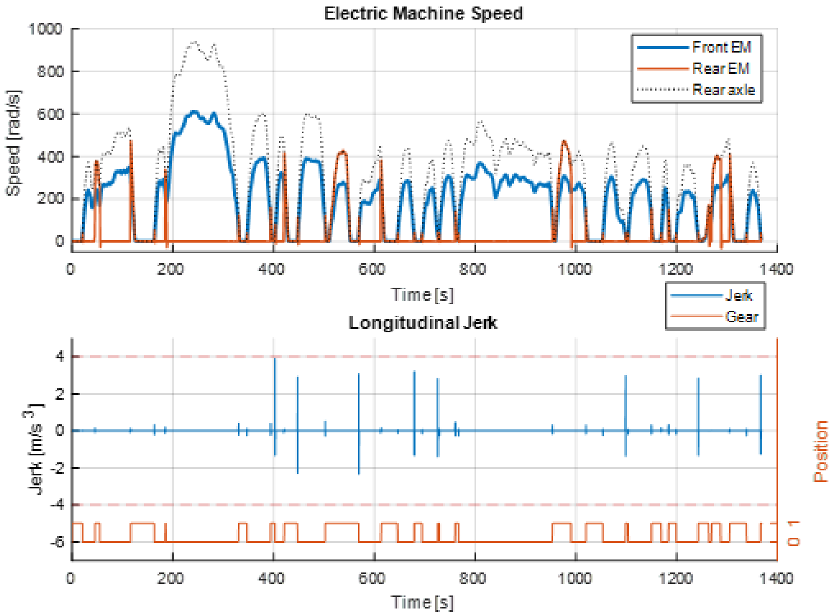


Figure 27: NVH Longitudinal Jerk Results

Further, the difference in simulation results can be seen in Table 7 below for the top three architectures. On average, the disconnect logic that takes into account drivability reduces the vehicle's range by two percent.

Table 7: Disconnect Control's Effect on Vehicle Range

Architecture	**Range [mi] (Base Disconnect Controls)**	**Range [mi] (NVH/Drive Quality Disconnect Controls)**	**Change in Range [%]**
1 (Chosen)	219	215	1.83
2	217	212	2.30
3	210	206	1.90

Sensitivity Analysis

Finally, a sensitivity analysis was completed for the team's top three architecture options. The weight and motor efficiencies were varied in the study. The weight was adjusted between a range of ± 50 kg in increments of 10; further, the motor efficiencies were adjusted within a range of ± 2% in 0.4% increments. This totaled to 121 combinations. The results from the sensitivity analysis can be seen below for the architectures. This study helped the team conclude that the proposed or chosen architecture will hit team VTS even if received components operate at the lower end of their nominal peak performance.

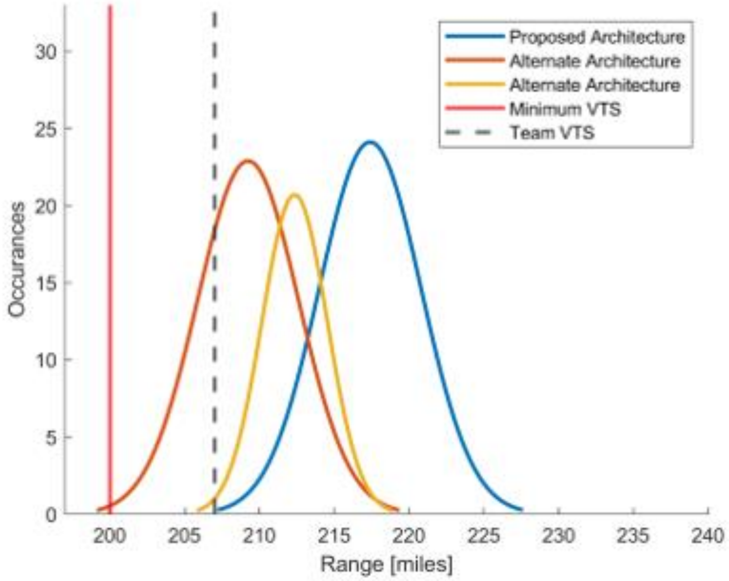


Figure 28: Sensitivity Analysis Results

3.2.2 Model Overview

The team minimized the model assumptions with industry level standards and tolerances along with sponsor data with high accuracy. The modeling was performed on the MathWorks tool, Virtual Vehicle Composer that was adapted to simulate a Cadillac LYRIQ. With the partnership of General Motors, the MathWorks provided tool was validated internally to a high accuracy in the energy consumption parameters. Further, the team validated other elements of the model by comparing it to published data and past testing experiences.

AWD Model Validation

One of the crucial steps while modeling the AWD architectures is to validate the energy split strategy of an existing AWD vehicle. This will allow the team to confidently measure the ability of the Hamiltonian Minimization strategy in the multi-motor configuration on the simulation tool. A Ford Mustang Mach-E GT is considered for this validation process. The data that is required to model this were collected from published sources and are tabulated in Table 8.

Table 8: Validation Data

Parameter	Source	Data	Notes
EM1 (Front)	Ford Published – Eluminator Motor [10]	Peak Torque: 430 Nm Max Speed:12000 RPM	Scaled Permanent Magnet EM
EM2 (Rear)	Ford Published – Eluminator Motor	Peak Torque: 430 Nm Max Speed:12000 RPM	Scaled Permanent Magnet EM
Gear Ratio	Ford Published	9.05	
Wheel Radius	EV-Database [11]	365	Millimeters
Mass	EPA MY2023 Test Car Data [12]	2328	Kilograms
Road Load Coefficient A	EPA MY2023 Test Car Data	198.024	N
Road Load Coefficient B	EPA MY2023 Test Car Data	3.498	N/ms
Road Load Coefficient C	EPA MY2023 Test Car Data	0.476	N/ms^2
Battery Cell Capacity	Private Test Data	70	Ah
Number of Cells in Series	EV-Database	94	
Number of Cells in Parallel	EV-Database	4	

Since the benchmarking vehicle has the same electric machine on both axles, the motor data can be scaled and reused. The new road load values, along with the mass and other crucial data, were replaced in the design data files. This vehicle was tested on the EPA's UDDS and HWFET cycles with the same test conditions for simulation validity. The important takeaway here is the minimization strategy is 11.6% under the EPA published numbers for the urban cycle and just under 0.01% under the highway cycle, as summarized in Table 9. This can be justified as the regenerative braking strategy can be optimized for the benchmarking vehicle. The team has a high level of confidence in the energy split algorithm and shows room for optimization in the future.

Table 9: Drive Cycle Simulations and Published Results

EPA Cycle	Published MPGe	Simulated AC Unadjusted MPGe	Error (%)
Urban UDDS	125.8	111.2	-11.6
Highway HWFET	107.6	107.5	-0.01%

Component Data Validation

All the data from the electric machine manufacturers were validated for consistency. The resolution of efficiency tables and other data was important to achieve a higher level of accuracy. In Table 10, the data for the top five team architectures is compared. In general, the efficiency/loss maps were high in resolution and validated by the manufacturers. On average, the tolerance for the efficiency maps is +/- 2%, and it was tested for effect on range in a sensitivity analysis. Some of the data that was not available from the manufacturers were approximated from dimensions in CAD, such as the rotor inertia for drive units.

Table 10: Drive Unit Data Summary

Parameter	Dana	AAM	Parker-Hannifin
Torque Curves	Peak and continuous	Peak	Peak and continuous
Performance Map Configuration	Loss table	Efficiency map and loss table	Efficiency map and loss table
Maps Resolution	41 x 31	50 x 48	61 x 31
Spin Loss	Interpolated from loss table	Explicit table	Interpolated from loss table
Rotating Component Inertia	Yes	No	Yes
Detailed CAD	Yes	Yes	Yes
Inverter Losses	Included in the loss table	Included in the loss table	Parameterized effective inverter loss equation (current and voltage based)

Level of Confidence and Assumptions

Some of the modeling and data parameters that may affect the range estimation and performance estimation are listed in Table 11. The team uses a combination of GM and supplier data to make assumptions while modeling. Some of the parameters that do not affect the energy consumption and performance were treated as standard parameters and approximated to market average ESUV estimates.

Table 11: Model Data Confidence Levels

Component / Parameter	Assumption	Source of Assumption	Level of Confidence of Assumption	Energy Consumption Impact	Performance Impact
Electric Drive Unit	Motor Torque Capabilities	Supplier	High	High	High
	EDU Efficiency		High	High	High
Energy Storage System	Battery Limits	General Motors	High	High	High
	Regenerative Braking		High	High	High
	SOC Region		Medium	High	Medium
Drive Line	Gearbox Efficiency	Supplier	Medium	High	Low
	Disconnect Control		High	High	High
	Friction Brakes/ABS	Model	Low	Low	High
Mechanical Integration	Drive Quality	Model	Medium	Medium	Low
	NVH		Low	Low	Low
	Vehicle Mass	Team	High	High	High
	Aerodynamic Drag		Medium	Medium	Low
DC-DC	Accessory Load	Model	Medium	High	Low
Thermal System	Thermal Modeling	Not Modeled	Low	Medium	Medium
	Derating Strategy		Low	Medium	High

The approximated braking model poses a low impact on energy modeling while adversely affecting the braking distance simulation. Similarly, the gearbox efficiency has a medium level of confidence and highly impacts the energy efficiency estimation. One of the major low confidence level components of the model is the thermal simulation. The team has requested additional data from GM to improve this area of modeling, however, this was not deemed a factor for selecting an architecture.

Probability

The team relied on data provided by the supplier. The data provided is either laboratory tested at a constant voltage or simulated through manufacturer's models. The suppliers also provided the team with an error tolerance and level of variance in the measurements. The team modeled this variance in a sensitivity analysis to understand the impact on the range and performance. Along with the team's EcoCAR Mobility Challenge data and experience from past models, the team identified a few critical parameters and blocks to understand the effect on the model, which are summarized in Table 12. The effect of tolerance is visualized in Figure 29. The loss map tolerance increases the range by 3 miles on average when the losses are reduced by 2%. Similarly, the range decreases by 4 miles on average when the losses are increased by 2%. Additionally, increasing the HV accessory load to a constant 2A will reduce the range by 6 miles on average. The backlash compensation algorithm will add delays in torque shuffle and reduce the range by 4.1 miles on average. The mass estimation only affects the range by +/- 3.2 miles when we add or remove 50 kgs from the estimated mass on the three architectures.

Table 12: Parameter and Block Variance Modeling

Parameter	Tolerance Based on EcoCAR MC	Tolerance Based on Supplier
Loss Maps	N/A	+/- 2%
Accessory Load	0 to 2A of HV load	N/A
Backlash Compensation	With and without backlash, the compensation algorithm	N/A
Vehicle Mass	+/- 50 kgs	N/A

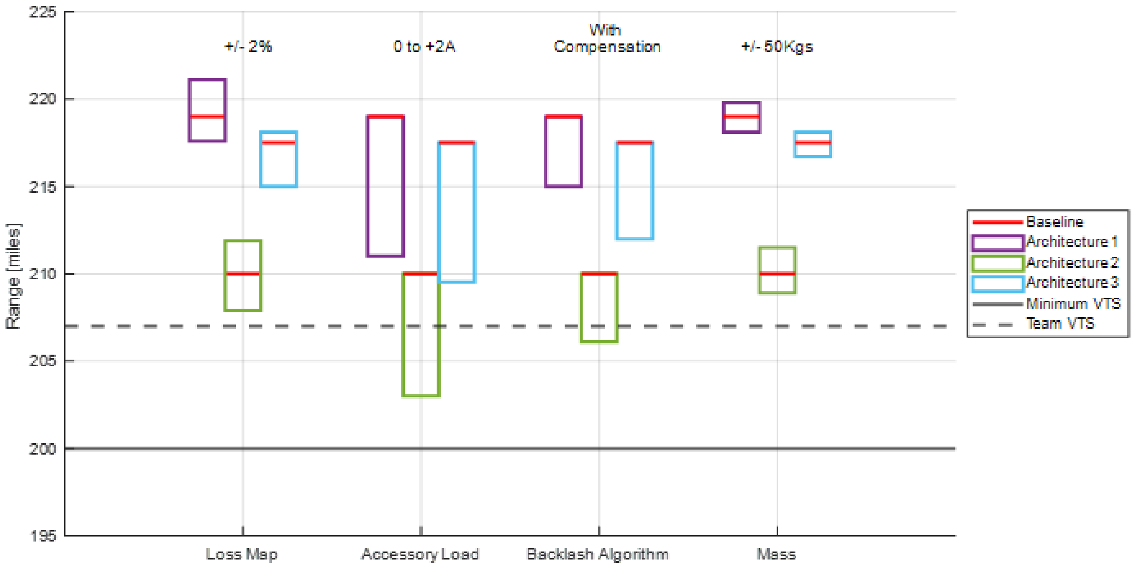


Figure 29: Parameter and Block Variance Modeling Results

Chapter 4: Vehicle Model Development

Chapter four describes the vehicle model developed. The plant model is built off of a plant model from MathWorks Virtual Vehicle Composer. The Virtual Vehicle Composer model served as a base that the team improved upon and modified to meet team needs. The plant model was used to develop and validate driveline control strategies, estimate vehicle efficiency, estimate vehicle acceleration, and validate mechanical, thermal, and electrical designs. The plant model was developed to model the team vehicle, which has been nicknamed OhioForce; because of this, the model is named OF-EVSim. OF-EVSim works in MIL, SIL, PIL, and HIL testing environments. The inputs and outputs in the HIL environment are all fed through the SpeedGoat IO blockset. More information on their block set can be found here [13].

4.1 Model Overview

OF-EVSim runs using Matlab Simulink. At the top layer the OF-EVSim has three main components the plant, driver, and controller. An image of the top level of the model can be seen in Figure 30 below.

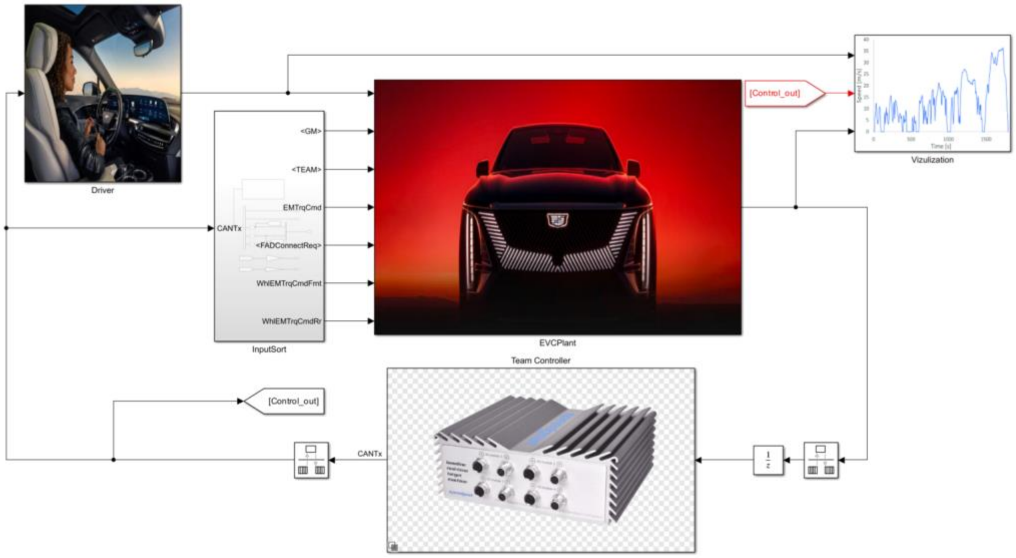


Figure 30: Plant Model Top Level View

The model is a forward based model that uses the difference in velocity between the plant and the drive trace to control the driver. The driver uses a predictive model. The model uses a basic physics-based equation to predict the current vehicle state. The equations for this can be seen below and the parameter values are contained in Table 13.

$$x_1 = v \qquad \text{Equation 2}$$

$$\dot{x}_1 = x_2 = \frac{K_{pt}}{m} - g\sin(\gamma) + F_r x_1 \qquad \text{Equation 3}$$

$$F_r = -\left[\tanh(x_1)\left(\frac{a_r}{x_1} + c_r x_1\right) + b_r\right] \qquad \text{Equation 4}$$

In the driver block the preview distance is chose, as well as, the driver response time. The preview distance corresponds with how far in the future the driver looks and the lag time delays the driver output to more realistically model a person. For a more complete description of the driver block refer [14] [15] [16]. Further gear commands are scheduled through the driver.

Table 13: Driver Block Parameters

Parameter	Value	Units	Symbol
Vehicle Mass	2920	Kg	M
Vehicle Total Tractive Force	2700	N	Kpt
Driver Response Time	0.3	s	Tau
Preview Distance	2	M	L
Rolling Resistance Coefficient	166.22	N	A_r
Rolling and Driveline Resistance	2.4571	N*s/m	B_r
Aerodynamic Drag Coefficient	0.52114	N*s^2/m	C_r

Inside the controller block there are two sub-variants the MIL controller and HIL controller. These can be swapped to allow for the plant model to be used in different testing environments. In the MIL model the controller applications are directly place in the plant model. In the HIL model the outputs and inputs flow through the SpeedGoat IO block set. This can be seen in Figure 31 and 32 below.

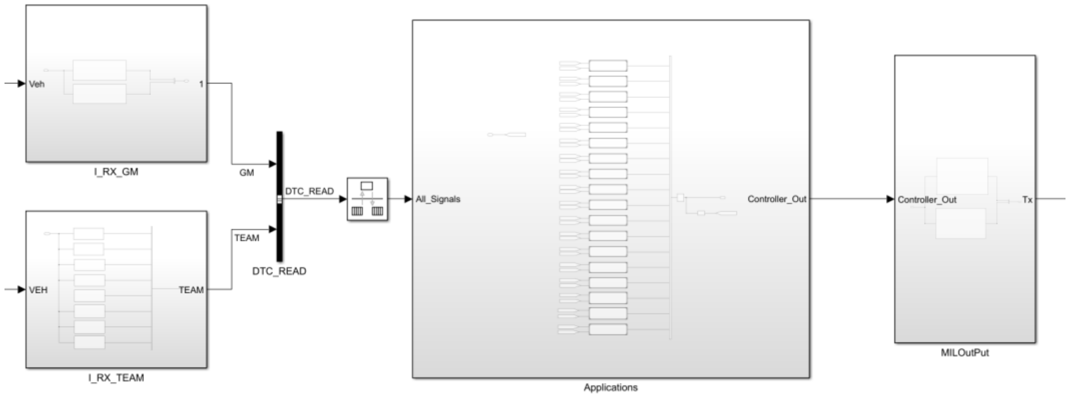


Figure 31: MIL I/O Layer Configuration

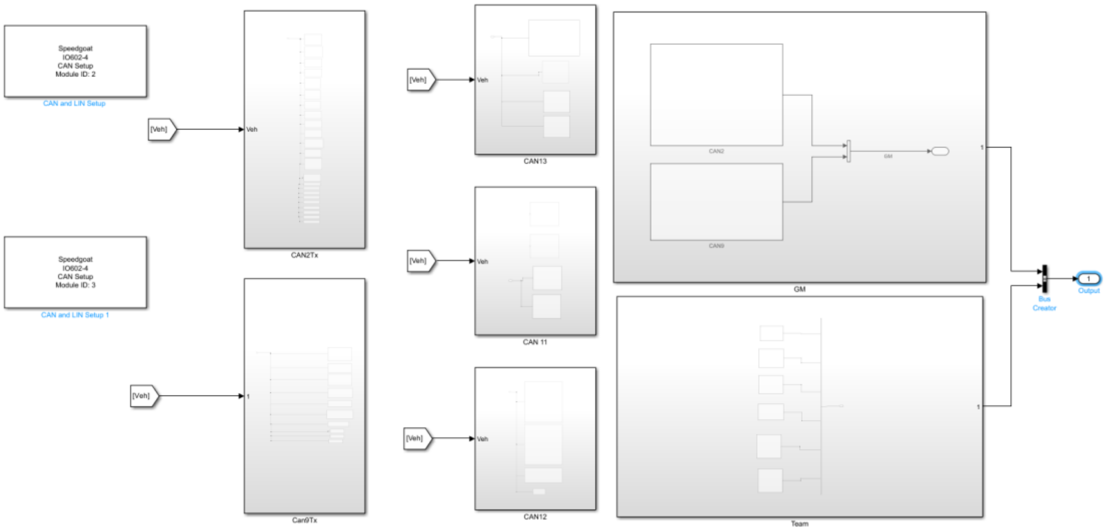


Figure 32: HIL I/O Layer Configuration

Finally the plant model has five main subcomponents drivetrain, electrical system, soft ECUs, regen braking, vehicle dynamics, and the pedal cluster. These subcomponents will be explained more in detail in the following sections.

4.2 Model Development Process and Structure

Before going into detail on the model components the development process for the entirety of the model will be detailed in this section. Further, the overall structure of the model will be explained.

4.2.1 Development Process

As stated in the literature review, the OF-EVSim was developed using a v-diagram approach and the model-based design process. The first step in the model design process is to define requirements. Vehicle level, system, and subsystem level requirements are generated at this point. Based on these requirements, model specifications were developed, and four main model specifications were included in the model development process.

1. Model Components: Based on the model specifications developed from team vehicle requirements the necessary components of the model can be chosen. Here what needs to be modeled versus what can be left out and dealt with in vehicle or component level testing is determined.
2. Interface: In this stage the overall layout of the model is determined. How the model will interact with each in environment is decided. For example, it can be how the virtual busses are setup, how model components and subsystems share information, and how signals will be processed and communicated on physical hardware.
3. Fidelity: The next stage is to determine the necessary amount of detail needed in the modeling. There are two main sides to fidelity in modeling. The first is what level of soft ECUs are needed for components. Many components, especially the GM stock system, have many different states and signals. Replicating everyone in a custom made stock sECU can be quite time consuming to execute correctly. Because of this, in this stage, it is decided what signals and states are necessary to replicate or not. The next decision in deciding model fidelity is to what level does the physics of the vehicle need to be modeled. For example, this model is used mostly for acceleration and fuel efficiency estimations. Using a 17-degree-of-freedom vehicle dynamics model would not make sense because most of the dynamics needed are just along the vehicle's longitudinal direction.

4. Assumptions: The final stage is to understand model assumptions. By understanding model assumptions, the level of accuracy of the model can be understood, and the significance of model results can be determined. Model assumptions and fidelity go hand in hand, as the level of assumptions made in the model will affect the overall fidelity.

Model components were decided based on requirement needs. The main driving requirements for model component decisions are functional, interface, and safety requirements. Here, model components are decided based on what is needed to test controller and vehicle functionality and safety to an appropriate level. All components in the model are safety and functionally critical. The goal of the model was to be as simple as possible but still meet team needs. In order to execute this, the plant model only consists of the five main sub-systems mentioned above. Some components are also interface critical; for example, many signals received and transmitted by the GM stock system are not used in the actual team controller. However, they are included in the plant model because the signals are needed for the stock system to perform correctly, as the team controller needs to spoof old stock GM signals from the original drive unit. Further, event based and time sensitive signals were included in the plant model to ensure that the controller would correctly interface with the stock GM system and team added components, especially during start-up, shutdown, and emergency processes. Both the model components and interface were based heavily on vehicle and system level requirements.

The model fidelity and assumptions are based more on component level requirements. At a high level they are based on vehicle level requirements but the overall model fidelity is built off of the fidelity of individual components and the level of validation needed for that sub-system.

4.2.2 Model Structure

The original model structure is based on the MathWorks Virtual vehicle composer structure; however, internal model components were changed in order for multiple people to be able to make edits to the model at one time. The main change to the model is referenced subsystems, were placed into the model. Reference subsystems allowed team members to create branches in git and work on specific sections of the model without overriding others' works. This was the main structural changes to the model. Further, most team-added systems and subsystems were directly integrated with current subsystems. The only team-added model components that are separate from the existing subsystems are the soft ECUs.

4.3 Model Components

The model is made up of five main subsystems: the electrical system, drivetrain, pedal cluster and regenerative braking, the vehicle dynamics model, and sECUs. The components in each of the subsystems. Are detailed below.

4.3.1 Electrical System

The electrical system is made up of four main components the battery, the front EDU, and the rear EDU. A top layer of the subsystem can be seen in Figure 33 below.

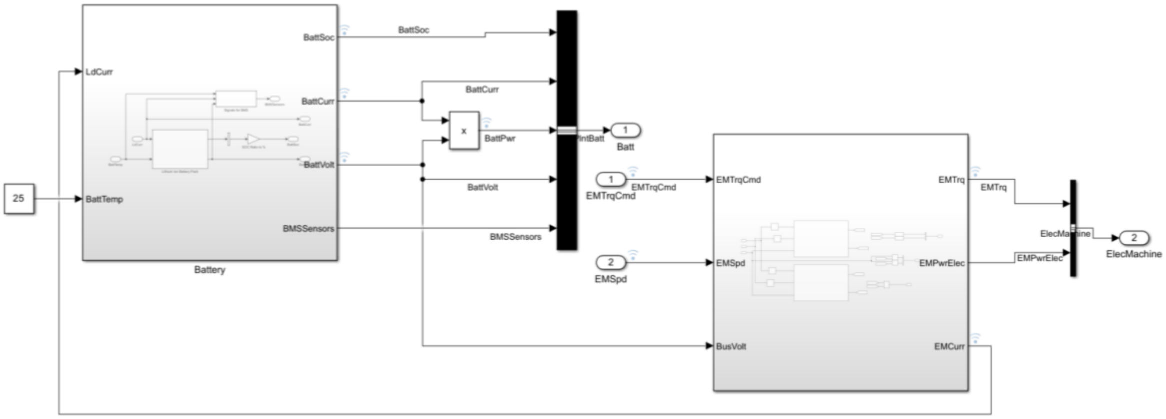


Figure 33: Top Layer of Electrical System Model

Battery

The battery used in the OF-EVSim is a mapped battery. All data for the battery and the map were given by general motors to support modeling of the stock LYRIQ battery. The mathematical relationships used in the battery model are shown below. Where SOC is the battery state of charge, I is the battery current, V is the battery voltage, R is the batteries internal resistance, and T isthe battery temperature.

$$SOC = \left(Batt_{Capacity\ Init} - \int I\ dt\right) * \frac{1}{Batt_{max\ capacity}} \qquad \text{Equation 5 [17]}$$

$$V = f(SOC) \qquad \text{Equation 6 [17]}$$

$$PowerLoss = I * R \qquad \text{Equation 7 [17]}$$

$$R = f(SOC, T) \qquad \text{Equation 8 [17]}$$

The battery's internal resistance and voltage are both lookup tables in the model. All other outputs are determined based on physics-based equations. The exact details of GM's battery are confidential and have been left out of this thesis.

Front and Rear Electric Drive Unit

The front and rear drive unit are identical. The electrical current delivered to the electric drive unit is based on a 2-D loss map supplied by Dana, the motor supplier. The input to this loss table is the EM speed and torque. From here electrical losses are found and overall power demand from the motor is then used to calculate the current demand from the battery for each motor. The relationship can be seen below.

$$I_{Batt} = I_{FEM} + I_{REM} \qquad \text{Equation 9 [18]}$$

$$I_{EM} = \frac{P_{mech} + P_{Loss}}{V_{Batt}} \qquad \text{Equation 10 [18]}$$

$$P_{mech} = T\ \omega \qquad \text{Equation 11 [18]}$$

$$P_{loss} = f(T, \omega) \qquad \text{Equation 12 [18]}$$

$$T = T_{cmd} - \tau \int T_{cmd} \qquad \text{Equation 13 [18]}$$

Where I is the system current, P is the power, T is torque, omega is rotational speed, and tau is EM torque control time constant. Further, the cmd refers to the commanded EM torque. The details on the mechanical dynamics of the drive units will be discussed in the driveline section.

4.3.2 Drivetrain

The drive train is made up of two subsystems the electric motor dynamics system, and the differential and axle subsystem. The inputs and outputs to theses subsystems can be seen below.

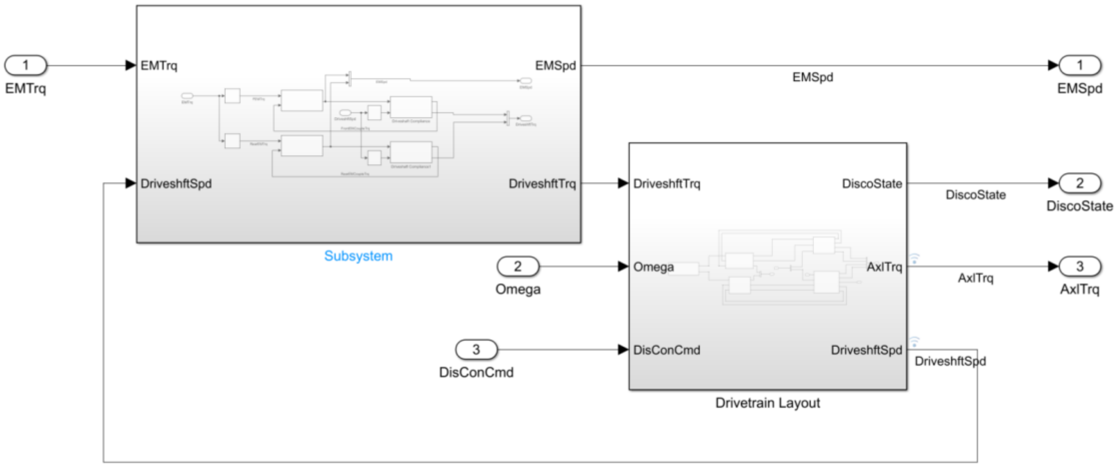


Figure 34: Top Level of Drivetrain Subsystem

The majority of this system was inherited from the original MathWorks virtual vehicle composer model. The main changes were in the parameters of the components. These, again, will be left out of the reports due to confidentiality. The other major change in this system is the addition of an axle disconnect on the rear driver-side axle. This was done by modeling an ideal fixed gear transmission with a single gear and a neutral. Further, there were no frictional losses and very low inertia in the model. To integrate the transmission model on the axle, a secondary torsional compliance block was added to model the dynamics of the shaft between the differential and axle disconnect.

4.3.3 Pedal Cluster and Regenerative Braking

This system has two main functions. The first is to smooth out and bound the driver accelerator and brake pedal outputs. The second is to communicate to the team controller the amount of regenerative braking that is being requested by the motor. This is done because the team added driveline controller does not make decisions on braking, but it responds to motor torque requests from the vehicle's electronic brake control module

(EBCM). Because of this, it is necessary to model how GM's EBCM would request braking torque from the motors and how the ECBM splits braking torque between the mechanical brakes and the motors.

In the model, the braking system used is series regenerative braking, meaning braking is done regeneratively through the electric machines until their braking capacity is saturated. Then braking from the mechanical brakes is requested. Other constraints have been put on regenerative braking as well, such as cutting off regenerative braking at low vehicle speeds and at high battery states of charge. The basics of the regenerative controller are shown in the equation below.

$$T_{EM} = \min\left(T_{EM\,max}, T_{req}\right) * C_1 * C_2 \qquad \text{Equation 14 [18]}$$

$$T_{Brake} = T_{EM} - T_{req} \qquad \text{Equation 15 [18]}$$

$$C_1 = f(v); C_2 = f(SOC) \qquad \text{Equation 16 [18]}$$

Where T is torque, C1 and C2 refer to the regenerative braking de-rate factors, the subscript req refers to the overall brake torque request, brake refers to the mechanical brakes, and EM refers to the electric motor. The above relationships are used to determine the amount of braking that will be done by the motors and electric machines.

4.3.4 Vehicle Dynamics

The vehicle dynamics model is quite simple for the OF-EVSim. A one degree of freedom vehicle body model is used from the vehicle dynamics toolbox, there is no modeled suspensions, a simple disk brake model is used, and a magic tire model is used.

1 DOF Vehicle Body Model

The vehicle body model uses basic vehicle dynamic relationships; pairing together a simple road load equation and basic dynamics relationships about the vehicle center of gravity (CG) and axle. The inputs to the model are the front and rear wheel tractive force, road grade, and wind speed. The block outputs the vehicle displacements, velocity, and acceleration from the CG, as well as, the force in the z – direction applied on the front and rear tires. More details can be found [19].

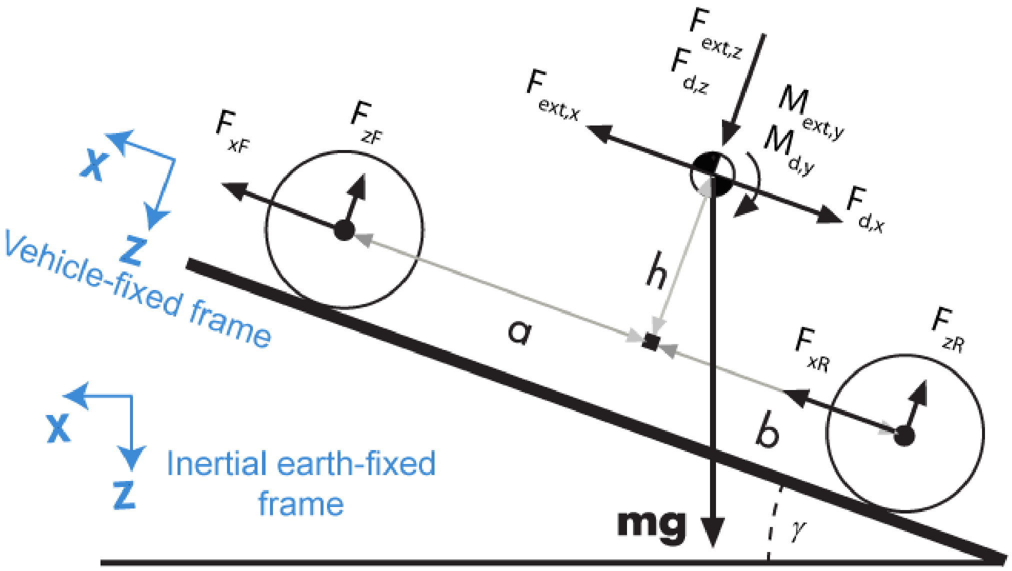


Figure 35: Vehicle Body Model [19]

Brakes and Magic Tire Model

The brakes and magic tire model were developed by the team using data and parameters given from general motors. The brake model can be seen below in Figure 36.

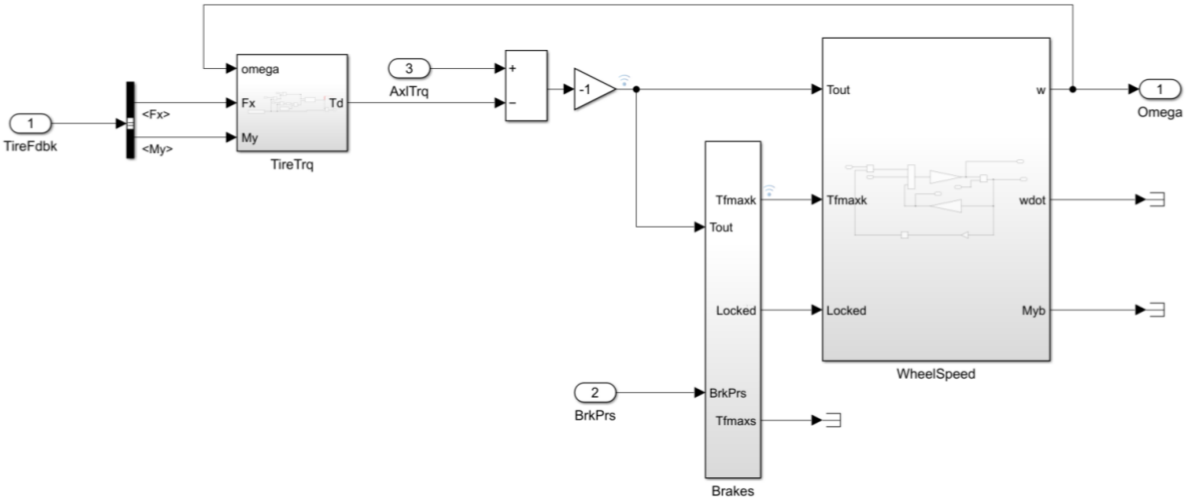


Figure 36: Top Level of Brake and Tire Model

There are three subsystems that make up the brake module. The TireTrq subsystem converts the forces on the tire to an overall wheel torque and applies a dynamic low pass filter to the output torque based on wheel speed. The dynamic low pass filter relationship is below.

$$y = F_{out} \qquad \text{Equation 17 [20]}$$

$$\dot{y} = w(u - y) \qquad \text{Equation 18 [20]}$$

$$u = F_x + \frac{M_y}{R} \qquad \text{Equation 19 [20]}$$

$$w = \frac{\omega R}{\tau} \qquad \text{Equation 20 [20]}$$

$$T_w = F_{out} * R \qquad \text{Equation 21 [20]}$$

Here T_w is the torque from the wheel, F_{out} is the force on the wheel from the road, R is the wheel radius, ω is the wheel speed, τ is the filter relaxation factor, M_y is the tire's moment about its y-axis, and F_x is the force from the road in the x direction.

The next subsystem of the brake model is the actual brake block. In this subsystem, the brake pressure command is converted to static and kinetic braking torque. And ABS is modeled using a clutching system on the brakes to prevent lock-up. The equations below are for the braking torques. Where P is brake pressure, B is the disc brake actuator bore, n is the number of brake pads, R_b is the mean brake pad radius, μ is the coefficient of friction, the subscript k refers to kinetic friction, and the subscript s refers to static friction.

$$T_{bk} = P_b * \frac{B^2 n\pi}{4} R_b * \mu_k \qquad \text{Equation 19 [20]}$$

$$T_{bs} = P_b * \frac{B^2 n\pi}{4} R_b * \mu_s \qquad \text{Equation 20 [20]}$$

The relationship used to determine if the wheel is locked up is below.

$$T_d = T_{axl} - T_b$$

$$Td \leq T_{bs}; Brake\ is\ locked\ up \qquad \text{Equation 21 [20]}$$

$$Td > Tbs; The\ brake\ is\ slipping$$

These relationships are fed into a combinational logic block with the previous output of the combinational logic block. To determine finally if the wheel is locked. The logical relationships and their outputs are contained in Table 14 below.

Table 14: Wheel Lockup Logic Table

Lock Detected	Slip Detected	Locked $_{t-1}$	Locked Output
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

The final subsystem of the brake module is the wheel speed calculation. The equation for this is below.

$$\dot{\omega} = \frac{1}{I}(T_{bk}\tanh(-4\omega) - T_d - \omega b); if\ wheel\ is\ slipping$$

$$\omega = 0; if\ wheel\ is\ locked$$

Equation 21 [20]

Where ω is the wheel speed, I is wheel inertia, T_{bk} is the max kinetic braking torque, b is the wheel damping factor, and T_d is the sum of torque between the tire and axle. These three subsystems work in conjunction to make the brake model.

As stated above the tire model used is a simple magic tire model the relationship for the tire model is below and Table 15 containing the equations letters meanings.

$$s = \frac{(\omega R - V_x)}{\max(V_x, \omega R)}$$

Equation 22 [8]

$$Fx = F_z Dsin\{Ctan^{-1}[\ (Bs - E)\ (Bs - \tan^{-1} Bs)\ ]\ \}$$

Equation 23 [8]

$$M_y = R(a + b * |Vx| + cV_x^2) * \left(F_z^{\beta} p^{\alpha} * \tanh(4V_x)\right)$$

Equation 24 [8]

Table 15: Symbols and Definitions for Wheel Model

Symbol	Definition	Units
s	Tire slip	Ratio
$\boldsymbol{\omega}$	Wheel speed	rad/s
F_z	Force in the z	N
Vx	Linear velocity	m/s
R	Wheel radius	m
B	Stiffness	Unit less
C	Shape	Unit less
D	Peak	Unit less
E	Curvature	Unit less
a	Constant force component	Unit less
b	Linear velocity force component	N
c	Quadratic velocity force component	$\frac{N\ s}{m}$
$\boldsymbol{\alpha}$	Tire pressure exponent	$\frac{N\ s^2}{m}$
$\boldsymbol{\beta}$	Normal force exponent	Unit less
P	Tire Pressure	kPa

The exact values of each parameter are left out of this report because it is confidential information from GM. The magic tire model works with the brakes model to find the overall wheel velocity and combined torque to the axle from the brakes and the wheels, which feeds back to the entire drivetrain model.

4.3.5 sECU's

The final components of the plant model are the sECUs. These are implemented to model states, state transitions, and the corresponding signals for vehicle components and system. There are five main sECUs the EDUs, the axle disconnect controller, the GM stock system, the power systems, and finally the analog/digital signals and pumps. The following sections will discuss these sECU's and the amount of fidelity that was used in each one and the reasoning behind it.

Electric Drive Units sECU

The front and rear EDUs each have their own separate but identical soft ECUs. The goal of these soft ECUs is to model the different operating states of the electric drive units and to ensure that the team controller can successfully transition the EDUs from state to state and respond to any requests or emergency states from the EDUs. The internal states of the EDUs and necessary transition criteria were collected from documentation supplied by Dana. Figure 37 below is the state chart used to model the electric drive unit’s internal states.

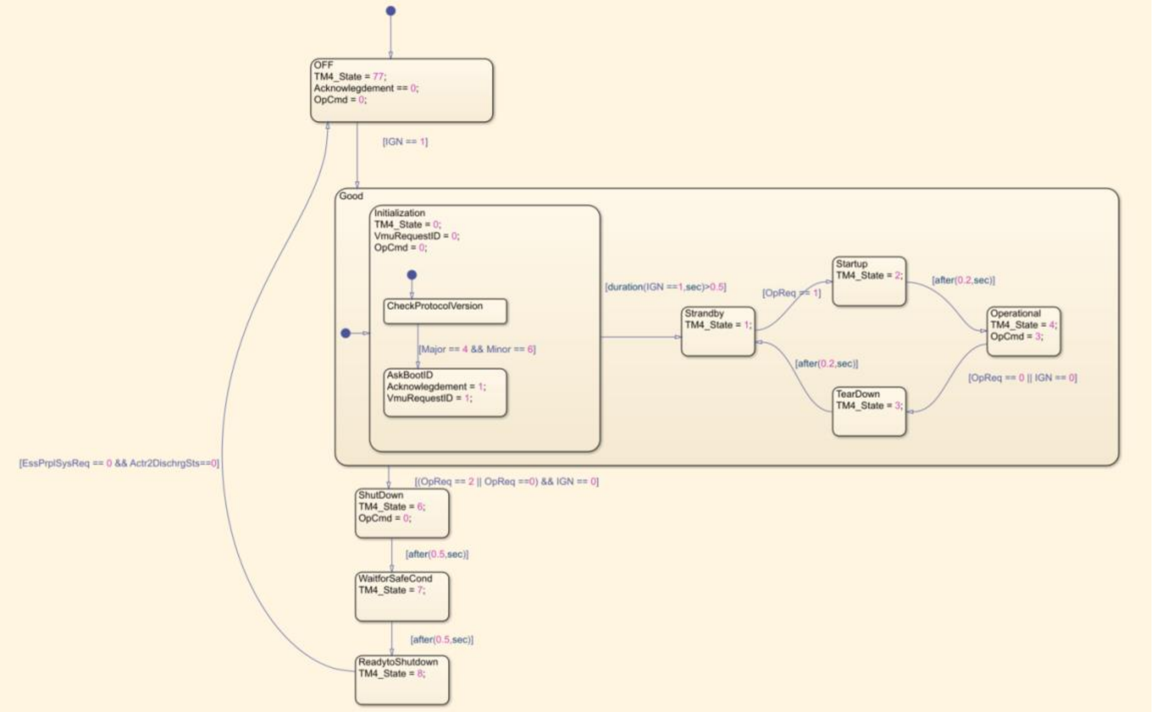


Figure 37: EDUs’ Stateflow Chart

In conjunction with the above state flow charts, emergency signals, and invalid flags were also added to the plant model. These were executed using step functions and were only initialized in specific test cases.

Axle Disconnect Controller

The axle disconnect is paired with its own independent controller which the team's controller communicates with. The team created a sECU of this controller. Like the drive unit sECUs, a stateflow chart was made to model the axle disconnect controller. Again this controller was made to ensure that the team could take the axle disconnect through its proper states and monitor the overall status of the system. The stateflow chart can be seen below in Figure 38.

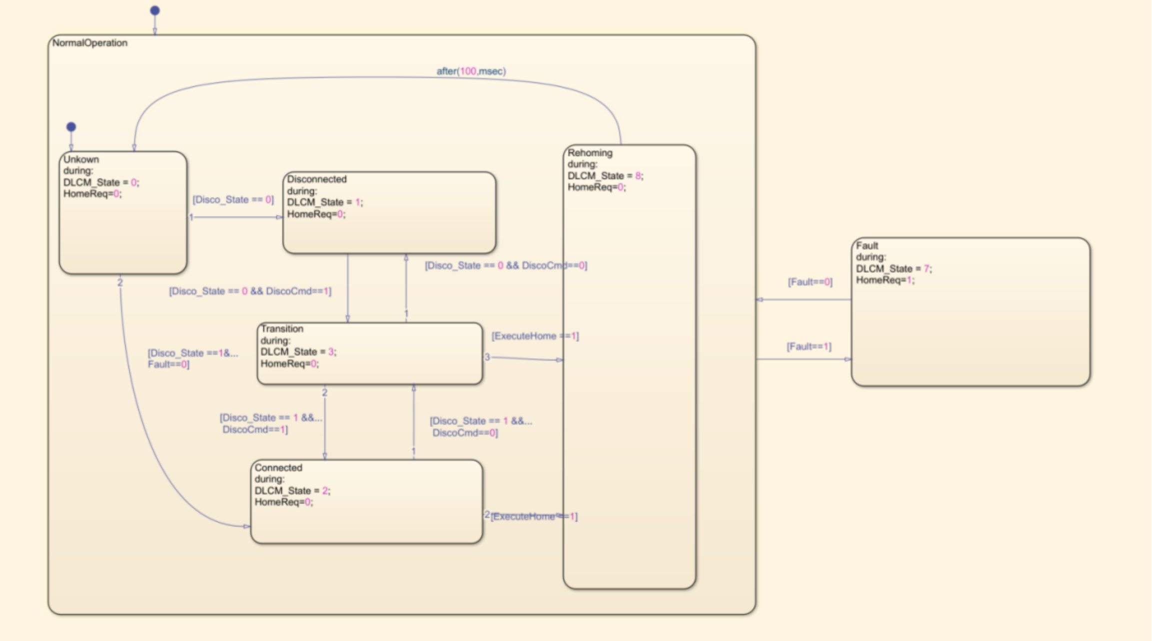


Figure 38: Axle Disconnect Controller's Stateflow Chart

GM Stock System sECUs

There are three main components to the GM stock soft ECU. These are the GM power limits, the GM internal PRNDL logic, and power moding. The GM power limits are voltage, current, and power limits from the GM stock battery. These are included in the model to ensure that the team controller never requests operations outside of these limits. All of these limits are implemented as 2-D lookup tables where the two inputs are the battery SOC and the battery's last operating power. These lookup tables were generated directly from data from the stock LYRIQ. Currently, these look-up tables only function between zero and thirty percent SOC because the vehicle is limited to that charge range.

The GM internal PRNDL logic is a simple state space as well. The team controller is responsible for controlling shifting, confirming the state of the vehicle to the rest of the vehicle, and controlling the electric park brake. The PRNDL soft ECU models the park brake states and the response messages on the vehicle state from the stock vehicle system.

The final soft ECU is the GM power modeing soft ECU; this soft ECU is a simple state space model that is used to test vehicle startup, normal shutdown, and emergency shutdown procedures. The state space chart can be seen in Figure 39 below. The state space chart is very simple and ignores a lot of the sub-states of the stock GM system. This decision was made because complete documentation on all sub-states was never received from GM, so it was only possible to model overall states that were apparent through team data collection.

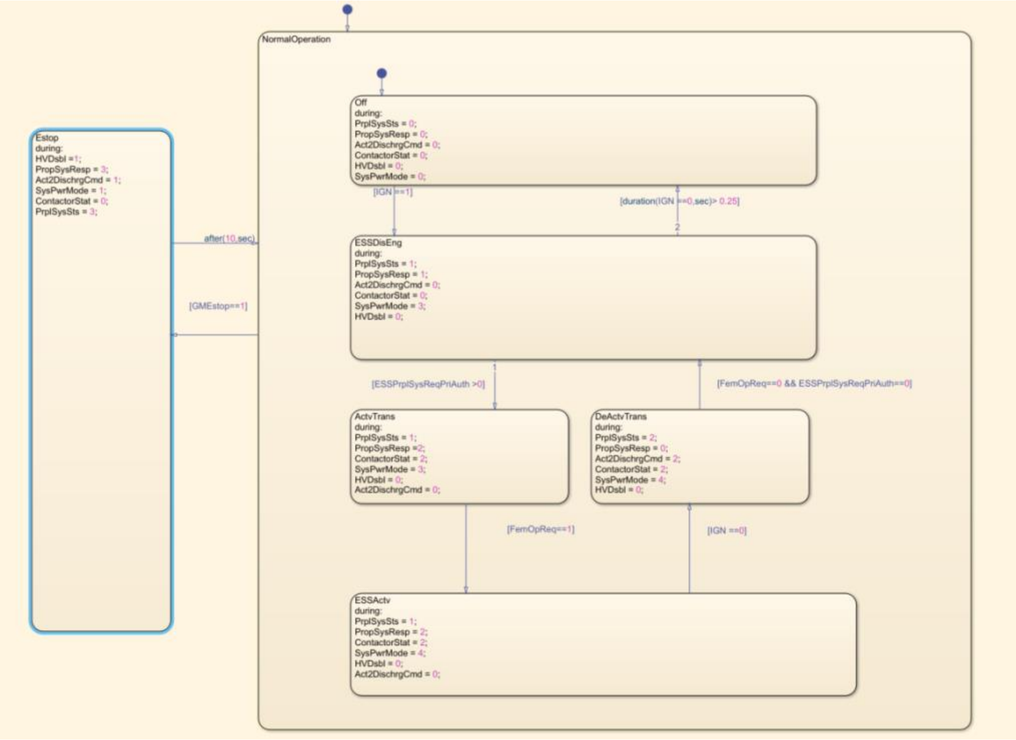


Figure 39: GM System Power Moding sECU

Power Systems

The power systems consist of two main categories team controlled low voltage distribution boxes and high voltage monitoring devices. The vehicle has three added team low-voltage power distribution boxes. There are two multiplexed vehicle electrical centers (MVEC) and one intelligent power distribution system (IPDS). The MVEC is a fuse and relay box which communicates relay and fuse states and status. The IPDS is a relay box that communicates the relays’ states and status as well. The sECU for these are a simple state space, shown in Figure 40 below.

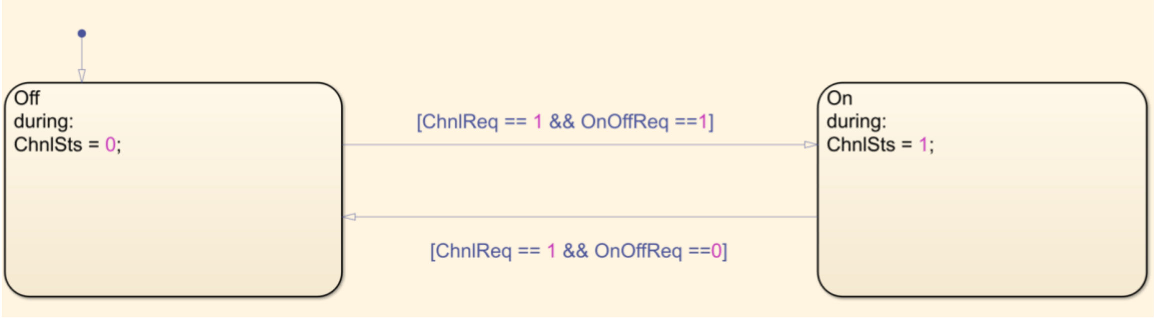


Figure 40: Relay Systems' Soft ECUs

The high voltage monitoring system has two devices, the first being a rail voltage reader from CSM called HV PT2. In the model, this device simply reports the rail voltages from the battery. A step function is used to model a ground fault, which would be a low rail voltage measurement. The next device is a Bender isolation monitoring device. This device is used to measure the HV bus isolation. This sECU also uses step functions to change the isolation reading values, signal validity, and device states.

Analog/Digital Signals and Pumps

The final soft ECUs of the plant model are the analog and digital signals and pump signals. The analog signals are all voltage readings from different measuring devices; because the plant model contains no real thermal model, they are all fed a value by a constant block; however, they are paired with a step function that can change these values to be outside of the acceptable bounds of the controller to see how the team controller would respond in a thermal emergency. The digital signals are fed by a constant block whose signals are initialized at startup. Finally, the pump soft ECU has two parts, the first being pump speed. The pump speed is found by passing the pump's commanded speed through a transfer function to model the pump's response time. The second part of the pump signals are the pump error and emergency signals. These are all modeled using step functions so the team controller can be tested to see how it responds to errors in the thermal and lubrication systems.

4.7 Model Validation and Results

This next section goes over the results obtained from the model and how they were validated by comparing them to real world data.

4.7.1 Coastdown Test

The coast down test is the first test performed to validate the model. The coast down test validates the internal efficiency, inertias, and damping factors are correct on the drivetrain, body, and wheels model. The real world coast down test was performed at the Transportation Research Center (TRC) large oval track. Two tests were completed, one going east to west and the other going west to east. This, again, was to account for road grade and wind. In Figure 41, the speed versus resistance force curves from the model coast down test and real world coast down tests can be compared.

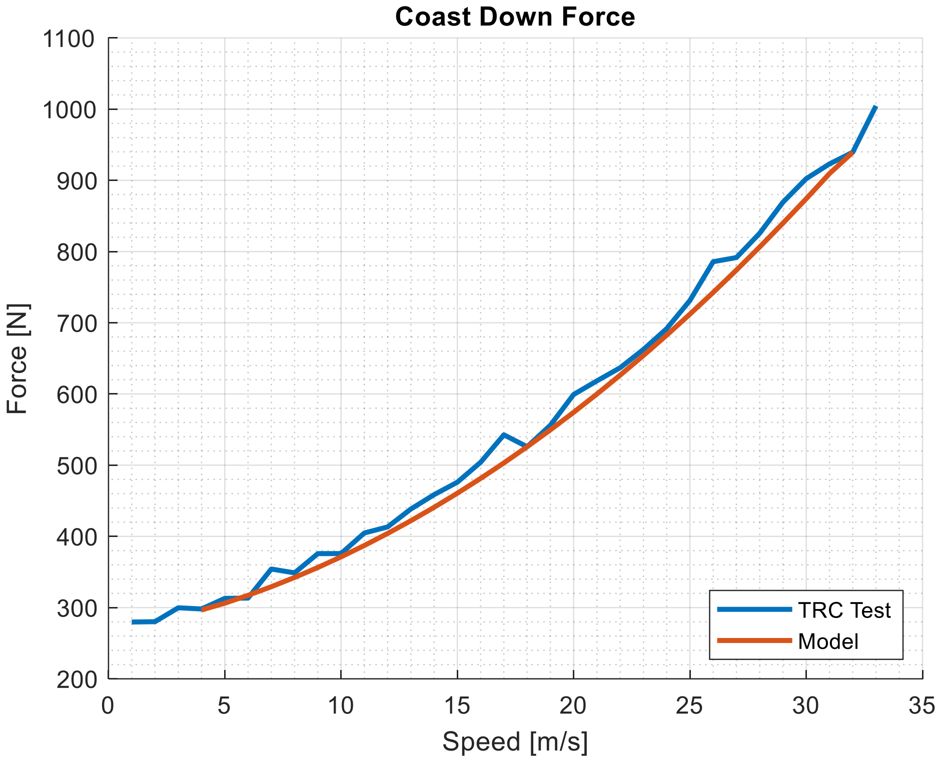


Figure 41: Coast Down Test Force Plot

4.7.2 Acceleration Tests

Using the model, an average zero to sixty time of 5.58 seconds was found. The image below shows the drive trace of the model for this zero to sixty time. This acceleration test was done by using a wide-open throttle (WOT) drive cycle. For this cycle, the driver applies full throttle until the target vehicle's speed of 60 mph is hit. The EM torque and vehicle speeds obtained from the model can be seen in Figure 42 below. The model results are in blue and the results from real world testing are in black and grey.

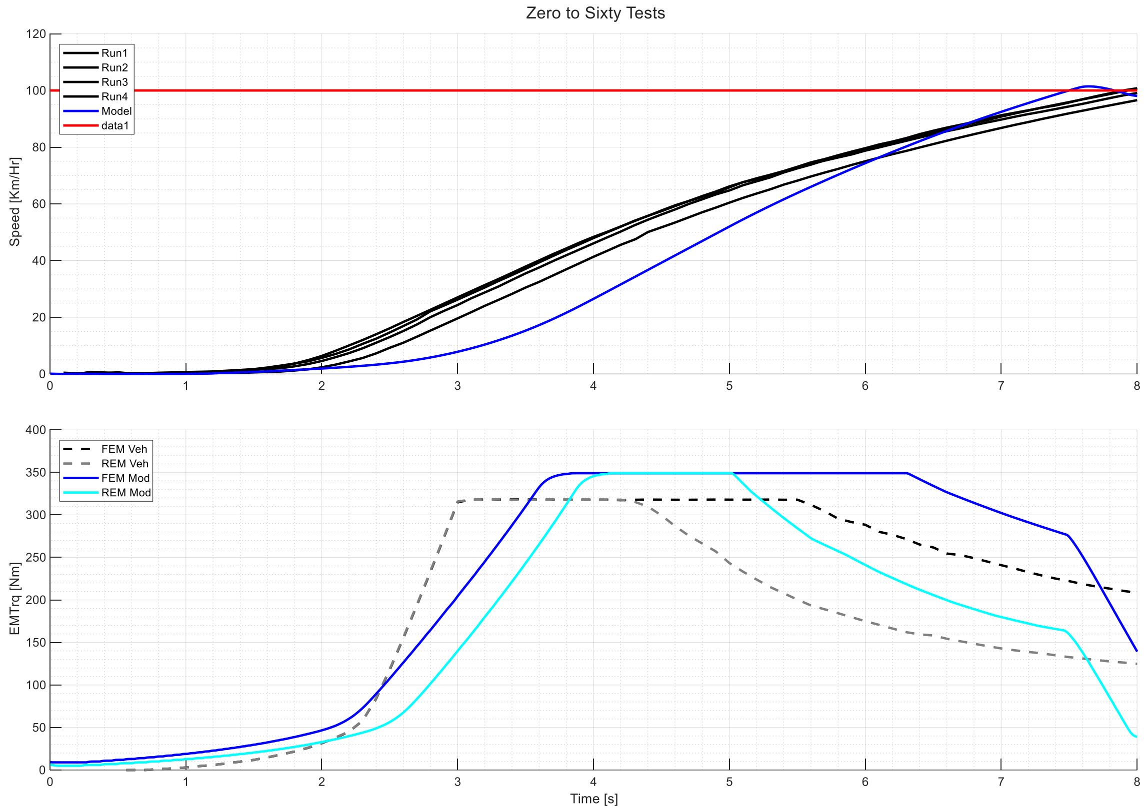


Figure 42: Acceleration Test Results for Model and Vehicle

The results are accurate when compared to real world driving of the vehicle. At general motors proving ground in Yuma, Arizona a professional driver completed a zero to sixty test for the vehicle. In this test four zero to sixty times were taken and averaged. Two runs went west to east and the other two went east to west to decrease errors caused by road grade or wind the results are tabulated below in Table 16.

Table 16: Zero to Sixty Test Times

Run	Avg Accel (g)	Time (s)
Yuma 1	0.42	5.66
Yuma 2	0.40	5.76
Yuma 3	0.42	5.72
Yuma 4	0.40	5.85
Yuma Average	0.41	5.75
Model Results	0.42	5.58

4.7.3 Energy Efficiency Test

The accuracy of the model for computing energy efficiency is also compared to a drive cycle complete on vehicle and the General Motors Desert Proving Grounds. The drive trace of the cycle can be seen in Figure 43 below. Only one run of this drive cycle was completed due to time constraints. The energy efficiency results from the actual drive cycle can be seen below in Table 17.

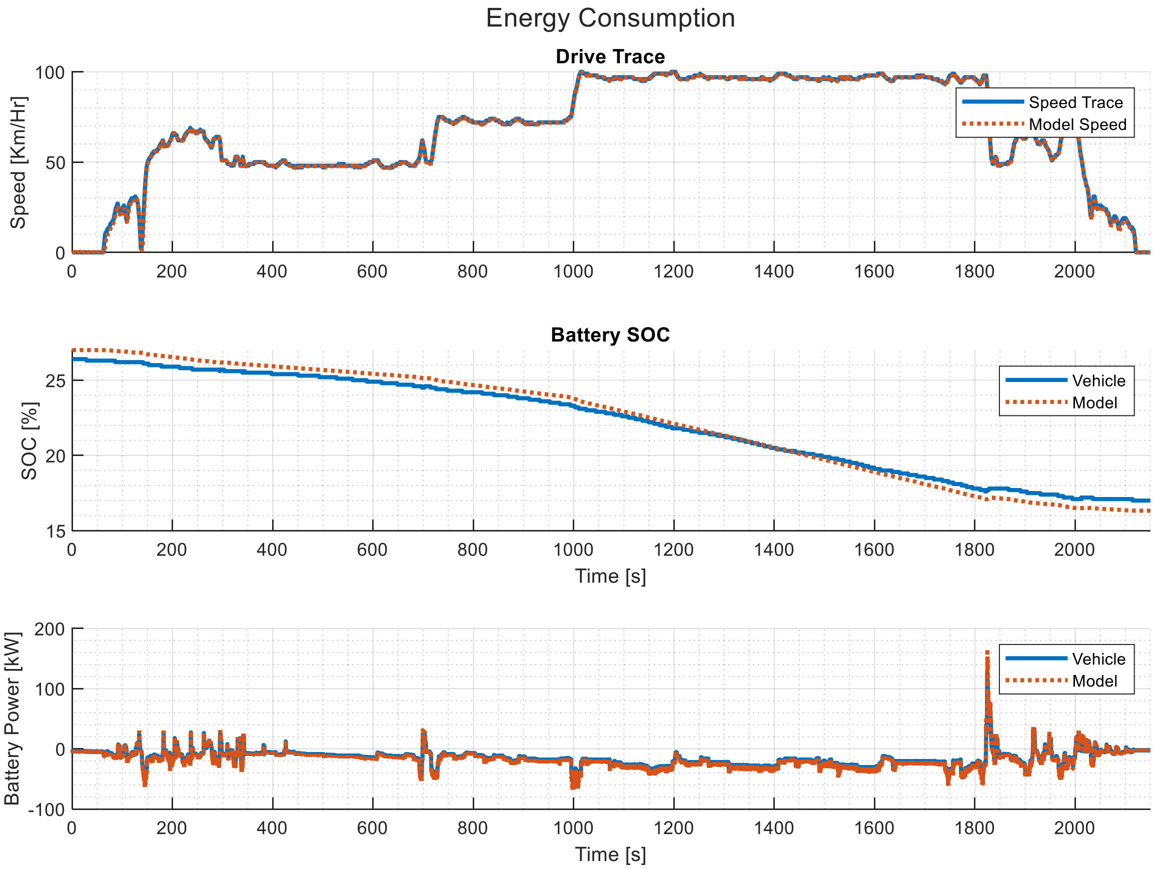


Figure 43: Energy Efficiency Results

Table 17: Validated Energy Efficiency Results

	Energy Efficiency (Wh/mi)	% Difference
Model	406	9.1
Vehicle	372	NA

Chapter 5: Testing Framework and Workflow

This chapter will expound upon the workflow and testing framework the team used to validate software functionality and safety before implementing the software on the team vehicle. On vehicle testing can be quite time consuming and vehicle availability in the second year of the competition was very limited due to the mechanical and electrical integration timeline being pushed back because of supplier delays. A virtual and benchtop validation process allowed the team to develop controls software in a safe manner and when the vehicle was down. The procedure is based off of research from the literature review and past experience on the team that dates back over twenty years. The team contributes its success at competition to this testing practice. In the following text, the testing procedure will be described, as well as the process the team used to decide testing environments and verification techniques. Further, the team requirements generation process will be described. All MIL, SIL, and HIL testing was done in a MathWorks Simulink environment, using SpeedGoat software packages for hardware integration on the team controller and HIL computer. Using these tools, testing was done at an industry-level standard, testing functionality, safety, fault responses, and signal input-output processing. This all allowed for a safe and robust control strategy to be applied to the vehicle.

5.1 Control’s Logic & Feature Verification & Validation

The team implemented a standardized process to validate controller logic and functionality. The controller validation process was generated alongside the plant model and plant model validation process. These were used together during year two to build the controller and plant model in parallel. The process overall follows a V-Diagram workflow, where vehicle level requirements were made, and controller features were decided from those. Year two dealt with rapid changes and implementation of controller features, so a more cyclical feature driven development approach was taken for the controller in the short term. The process started with the development of feature-level requirements. After that, test cases were designed in their respective environments. The environment in which the test case occurs depends on the controller's requirements and environmental testing abilities. After this step, the development process becomes more agile as multiple iterations of controls are developed and tested. Figure 44 below shows the overall flow of the team verification and development process.

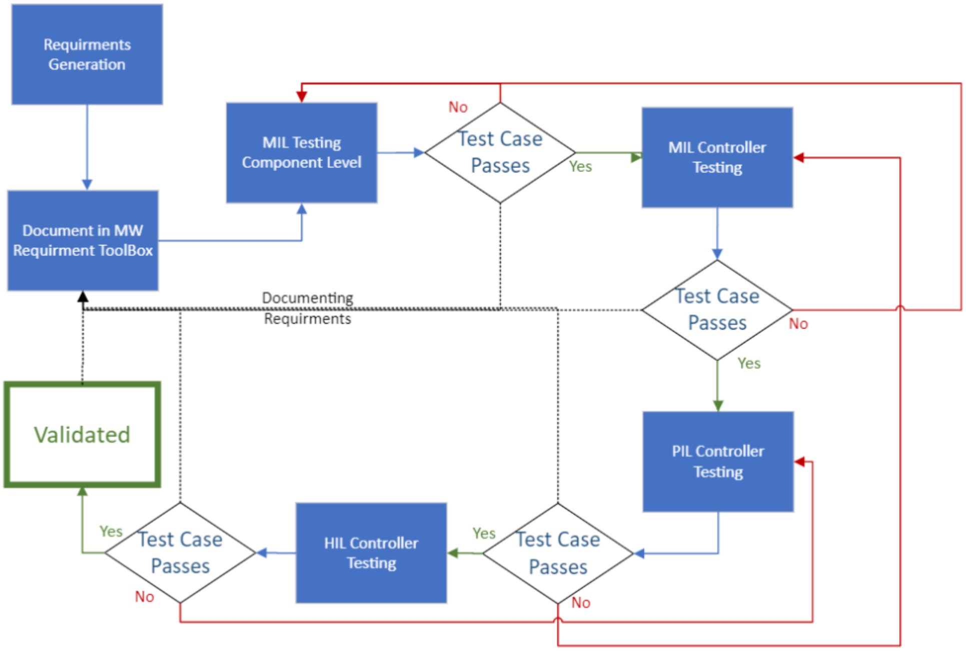


Figure 44: Software Testing and Validation Process

The next subsections will describe in detail each step of the process in Figure 44 above. This will be done by using an example controller feature and the steps the team took to validate the controller feature. The next section will describe this feature

5.1.1 Power Moding – Example Controller Feature Description

Power Moding is the controller feature responsible for the startup, shutdown, and emergency stop of the vehicle. This feature is safety critical as it is responsible for controlling the electric drive units' states and communicating them with the stock GM system. Further, this feature is responsible for requesting the high voltage contactors to close and open on the general motors battery pack. This feature is also responsible for monitoring all emergency signals and relays that are connected to the HV system, such as EDU e-stop signals, team e-stop signals, contactor relays, and ground isolation faults. In short, this feature is responsible for synchronizing the handshake for power on and power down of the stock GM system, as well as the team-added motors and components in normal and emergency scenarios. The power moding process is as follows.

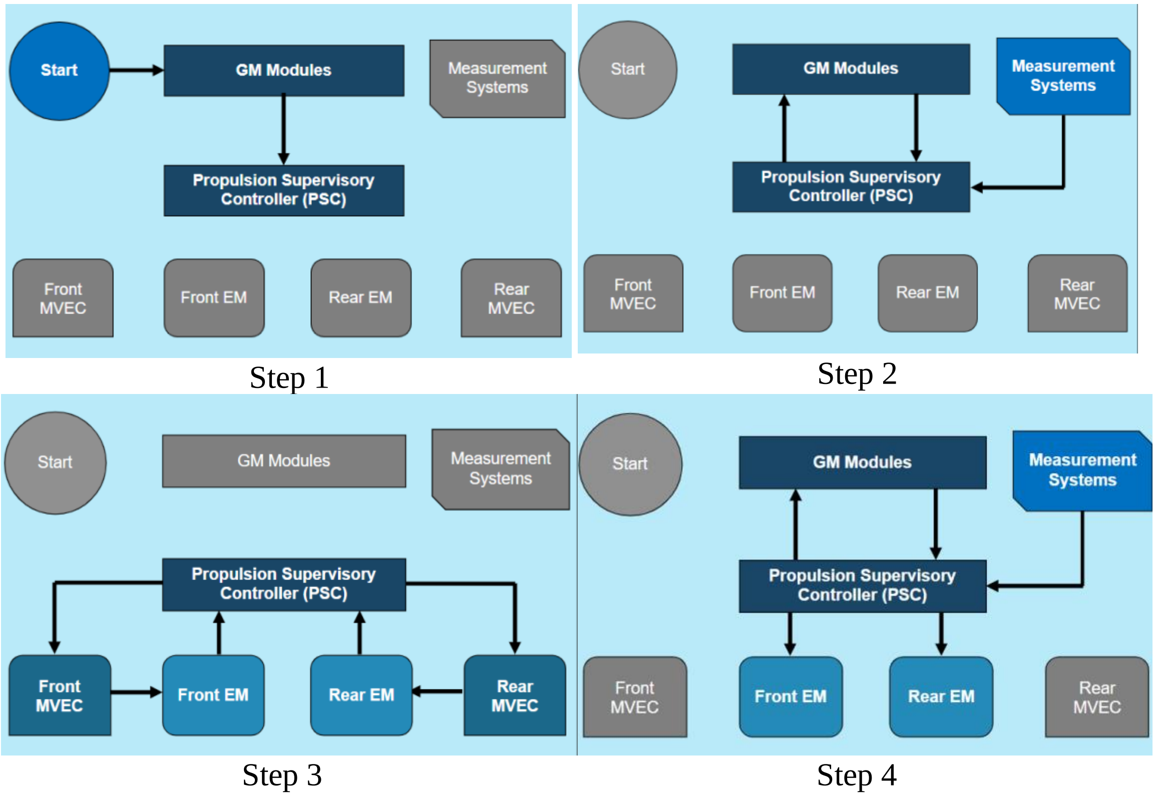
Step 1: Start request is received from the GM system after the driver has pressed brake and hit the ignitions.

Step 2: Team added controller checks, team contactor status, isolation values, HVIL status, and GM system states. Then the controller communicates this back to the GM system.

Step 3: The team added controller engages team controlled relays to turn on the electric drive units ignitions. It then check to ensure the electric motors are in standby state.

Step 4: The team controller then requests for the propulsion system to be active to the GM controller. The GM controller will then close battery contactors. After this the team added controller will request the EDU’s to be operational. From here all team added components and GM systems will be active and ready for driving.

Figure 45 below details the Power Moding start-up process, which will be used as an example for the remainder of the chapter.



Step 1 Step 2

Step 3 Step 4

Figure 45: Power Moding Process

The above functionality of the power moding algorithm were all generated from requirements which will be discussed further in the next section. The test cases and fault scenarios tested for power moding are quite extensive because the reliability of this feature was very important only a few will be detailed below.

5.2 Requirement Generation and Tracking

5.2.2 Requirement Generation

The majority of requirements stemmed from two major sources: supplier documentation and system safety tools. The team used a hazard and operability study (HAZOP) to generate possible safety risks. From this HAZOP the team generated controller requirements to prevent these unsafe scenarios. An example of a HAZOP case and analysis is shown in the table below.

Table 18: HAZOP Example

Example HAZOP Case	
Potential Vehicle Level Hazard	PROP_SYS_23.0
Potential System Hazard State	Unintended travel in the wrong direction, unintended propulsion flow
Team Designation	Locked activation of high voltage isolation
Hazard Description	PCM_HV SYSTEM
Worst-Case Mishap Potential	If high-voltage components of a vehicle remain disconnected from the rest of the electrical system, it can cause a loss of power to the vehicle's electric components, potentially resulting in unintended vehicle motion and decreased controllability.
Controllability or Other Mitigating Factors	The vehicle will not be able to start, and the 12 V battery may be drained, causing auxiliary systems to not be operable in the future.
Potential Causes for System Failure	Redundant controls and multiple levels of authentication to isolate the battery
Automotive Safety Integrity Level (ASIL) Assessment	Malfunction of the high voltage isolation system or damage to the electrical components of the vehicle.
System Safety Goal/Requirement	The system shall prevent locked activation of high voltage isolation.

In conjunction with the HAZOP analysis, a system theoretic process analysis (STPA) was completed. The STPA was used to help identify edge cases of use for the vehicle and controller. These are identified and unsafe control actions (UCAs). UCAs corresponding with electric drive units and power moding are contained in the table below with their corresponding requirement. These helped the team to capture all possible accidental or non-accidental use cases that could cause unsafe scenarios. These also went into the requirements generation. Finally, the team reviewed all documentation for components in detail and ensured that the requirements ensured the team controller would communicate with the added components and GM stock system as intended by the manufacturers.

Table 19: STPA Example

Unsafe Control Actions	**Potential Causal Scenarios**	**Requirements**
UCA-35: PSC does not command EDUs inverter enable in vehicle run mode - EDUs cannot provide propulsion torque or capture regen power. EDUs generate back EMF that damages the ESS.	Loss of communication between the PSC and the EDU	The PSC shall monitor the status of the EDU and ensure that systems communicate
	EDU loses power and cannot respond to PSC messages	The PSC shall monitor the feedback of the EDU to ensure that the EDU is functioning
	PSC does not correctly identify the state of the vehicle	The PSC shall monitor the state of the vehicle with redundancy to ensure the proper functions of components in each drive mode
UCA-37: PSC commands EDUs inverter enable in a vehicle off mode - Risk of unintended EDU propulsion torque.	PSC does not correctly identify the state of the vehicle	The PSC shall monitor the state of the vehicle with redundancy to ensure the proper functions of components in each drive mode
UCA-39: PSC enables inverter before ESS contactors close.	PSC does not correctly identify the state of the contactors	The PSC shall monitor the state of the vehicle with redundancy to ensure the

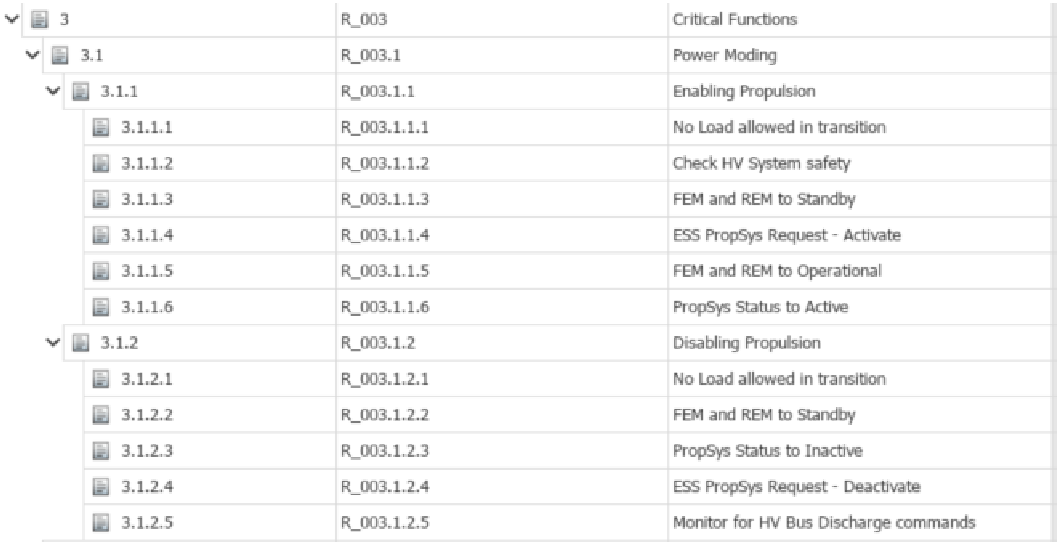
		proper functions of components in each drive mode.
UCA-40: PSC disables inverter during EDU operation	PSC does not correctly identify the state of the vehicle	The PSC shall monitor the state of the vehicle with redundancy to ensure the proper functions of components in each drive mode

5.2.2 Requirements Tracking

After requirement generation, requirements were tracked using MathWorks' Requirement Toolbox. This allowed for easy integration and automatic tracking of the requirements from the MIL, PIL, and HIL testing environments. Tests cases were generated for each requirement and linked to the requirement. When all the test cases for a given requirement was passed the requirement was satisfied. All of this was automatically tracked in MathWorks' Requirements Editor. There are five main requirements or requirement categories kept by the team.

1. Main Propulsion Supervisory Controller
2. Bus Loading
3. Critical Functions
4. Team Added Functionality
5. System Safety

Each of these overall requirements or categories is made up of smaller requirements, which are then broken down into smaller feature level requirements that are applied straight into the controller. Figure 46 below shows how power moding is just a subrequirement of the critical functions category and how power moding is broken into smaller feature level requirements.



3	R_003	Critical Functions
3.1	R_003.1	Power Moding
3.1.1	R_003.1.1	Enabling Propulsion
3.1.1.1	R_003.1.1.1	No Load allowed in transition
3.1.1.2	R_003.1.1.2	Check HV System safety
3.1.1.3	R_003.1.1.3	FEM and REM to Standby
3.1.1.4	R_003.1.1.4	ESS PropSys Request - Activate
3.1.1.5	R_003.1.1.5	FEM and REM to Operational
3.1.1.6	R_003.1.1.6	PropSys Status to Active
3.1.2	R_003.1.2	Disabling Propulsion
3.1.2.1	R_003.1.2.1	No Load allowed in transition
3.1.2.2	R_003.1.2.2	FEM and REM to Standby
3.1.2.3	R_003.1.2.3	PropSys Status to Inactive
3.1.2.4	R_003.1.2.4	ESS PropSys Request - Deactivate
3.1.2.5	R_003.1.2.5	Monitor for HV Bus Discharge commands

Figure 46: Power Moding Requirements

Finally, each requirement is saved with a description and a link to its test case. This allows for easy knowledge transfer in the future years of the EcoCAR EV competition.

5.3 MIL Environment Development and Testing

The first step to testing is making the plant model able to test possible test cases. Many safety critical digital signals were monitored using a PCAN digital module, including the HVIL circuit status. Figure 47 below shows how, in the plant model, the HVIL status and other signals were changed using step blocks. An initialization script exists for each test case, which will automatically change the initial value, step time, and final value of these step blocks. This was repeated for both MIL, PIL and HIL simulations.

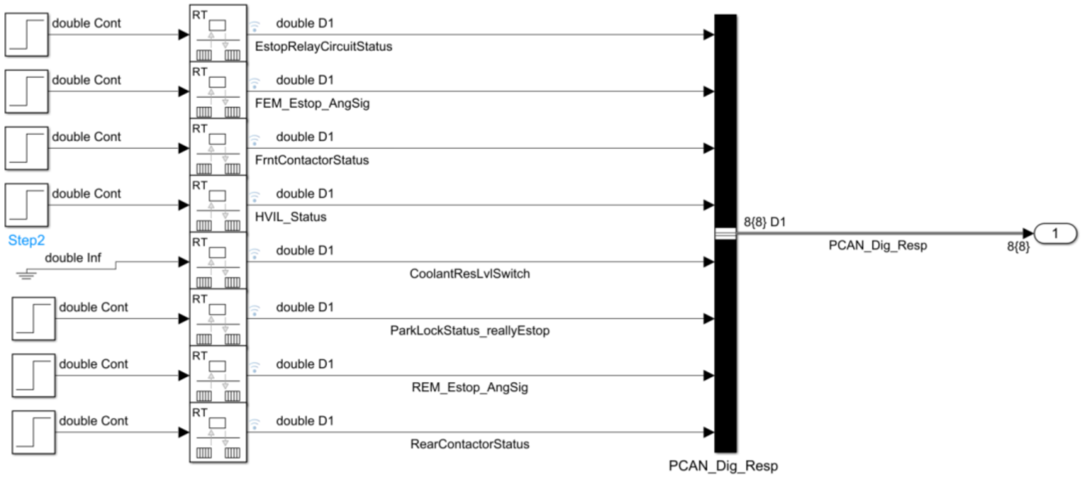
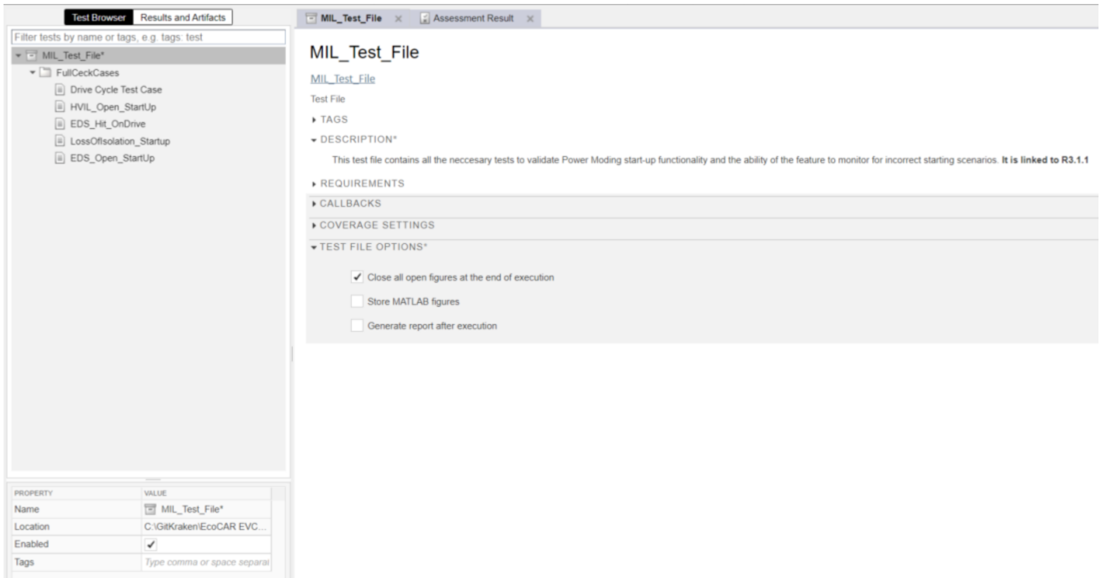


Figure 47: Example of Step Inputs for Signal Triggers

After the model changes are made, the next step is to implement the test cases. This is all done inside MathWorks Test Manager. The overall MIL test suite for the Power Moding start-up can be seen in Figure 48 below.



MIL_Test_File

MIL_Test_File

Test File

DESCRIPTION*

This test file contains all the neccesary tests to validate Power Moding start-up functionality and the ability of the feature to monitor for incorrect starting scenarios. **It is linked to R3.1.1**

TEST FILE OPTIONS*

☑ Close all open figures at the end of execution

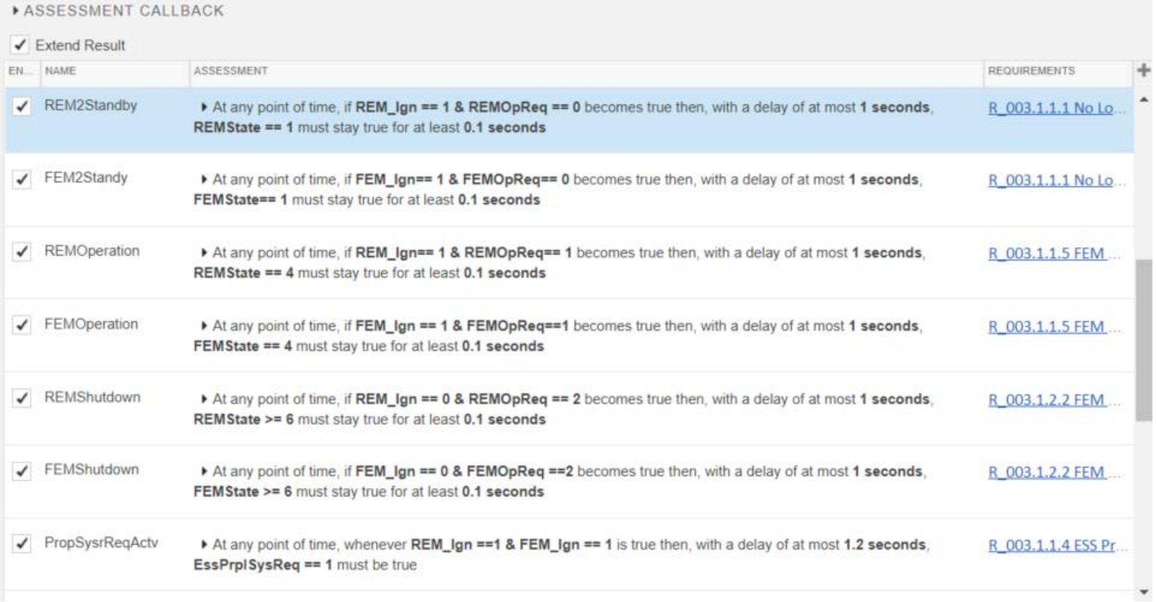
☐ Store MATLAB figures

☐ Generate report after execution

PROPERTY	VALUE
Name	MIL_Test_File*
Location	C:\GitKraken\EcoCAR EVC...
Enabled	✓
Tags	Type comma or space separa

Figure 48: Test Window

Inside this test suite there are individual test cases, then inside these individual test cases logical and temporal assessments are used to evaluate individual requirements that make up the power moding star-up requirements. An example of linked logical and temporal assessments can be seen in Figure 49 below.



▸ ASSESSMENT CALLBACK

☑ Extend Result

EN...	NAME	ASSESSMENT	REQUIREMENTS
☑	REM2Standby	▸ At any point of time, if **REM_Ign == 1 & REMOpReq == 0** becomes true then, with a delay of at most **1 seconds**, **REMState == 1** must stay true for at least **0.1 seconds**	R_003.1.1.1 No Lo...
☑	FEM2Standy	▸ At any point of time, if **FEM_Ign== 1 & FEMOpReq== 0** becomes true then, with a delay of at most **1 seconds**, **FEMState== 1** must stay true for at least **0.1 seconds**	R_003.1.1.1 No Lo...
☑	REMOperation	▸ At any point of time, if **REM_Ign== 1 & REMOpReq== 1** becomes true then, with a delay of at most **1 seconds**, **REMState == 4** must stay true for at least **0.1 seconds**	R_003.1.1.5 FEM ...
☑	FEMOperation	▸ At any point of time, if **FEM_Ign == 1 & FEMOpReq==1** becomes true then, with a delay of at most **1 seconds**, **FEMState == 4** must stay true for at least **0.1 seconds**	R_003.1.1.5 FEM ...
☑	REMShutdown	▸ At any point of time, if **REM_Ign == 0 & REMOpReq == 2** becomes true then, with a delay of at most **1 seconds**, **REMState >= 6** must stay true for at least **0.1 seconds**	R_003.1.2.2 FEM ...
☑	FEMShutdown	▸ At any point of time, if **FEM_Ign == 0 & FEMOpReq ==2** becomes true then, with a delay of at most **1 seconds**, **FEMState >= 6** must stay true for at least **0.1 seconds**	R_003.1.2.2 FEM ...
☑	PropSysrReqActv	▸ At any point of time, whenever **REM_Ign ==1 & FEM_Ign == 1** is true then, with a delay of at most **1.2 seconds**, **EssPrplSysReq == 1** must be true	R_003.1.1.4 ESS Pr...

Figure 49: Linking Logical Assessments and Requirements

After the test cases are created, the test can be run anytime there is an update in the controller software. The entire test suite can be run at once, or individual test cases can be run separately depending on the changes made to the controller and what the user wants to test. Below are the results from the HVIL open on start-up test case. This is checking to ensure that the controller never calls for the propulsion system to be active or for the GM contactors to close. Figure 50 shows what this assessment evaluation looks like inside of MathWorks Test Manager, and Figure 51 and Figure 52 compare a normal start-up sequence to the erred sequence.

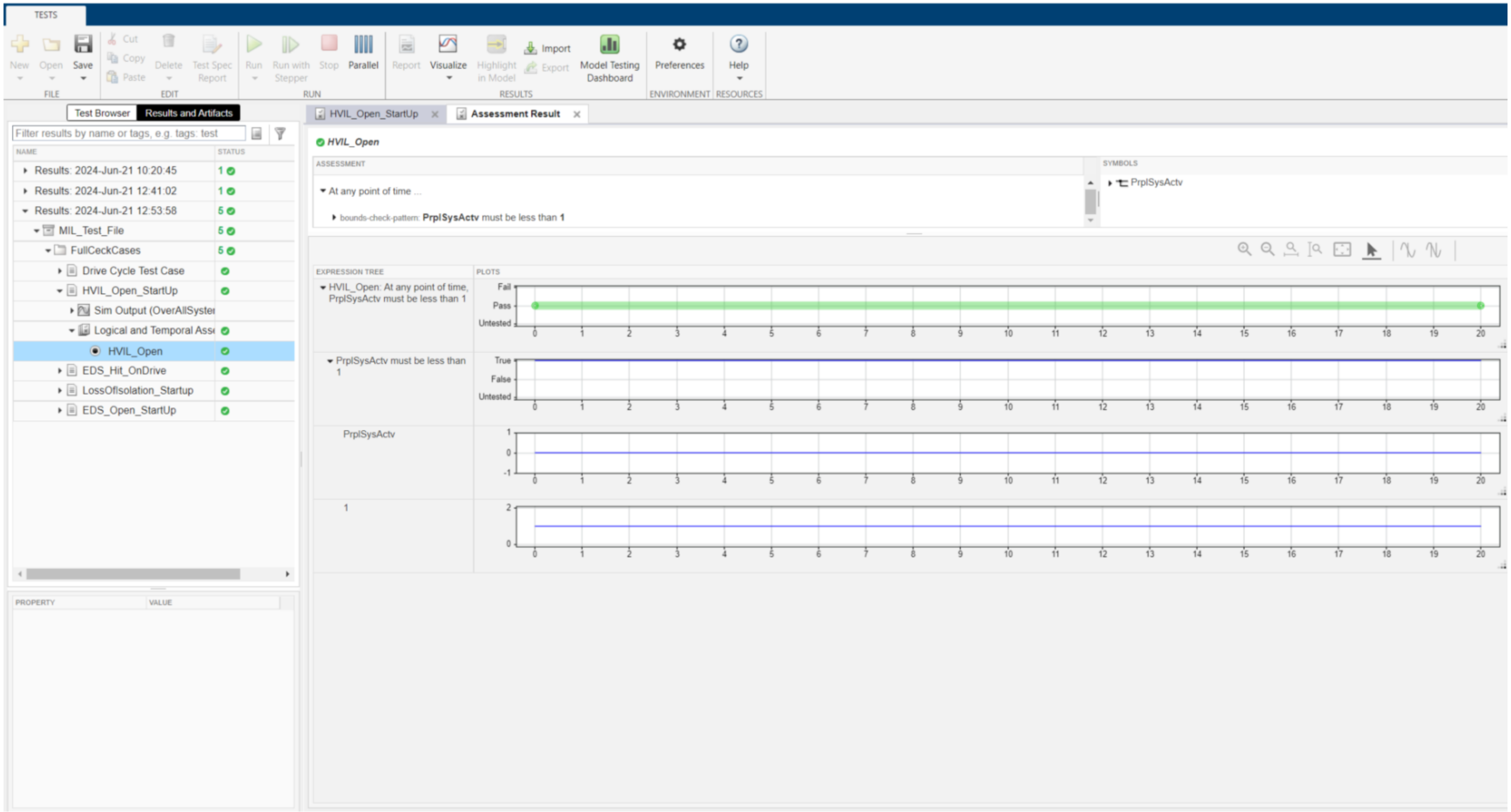


Figure 50: Evaluated Logical Assessment

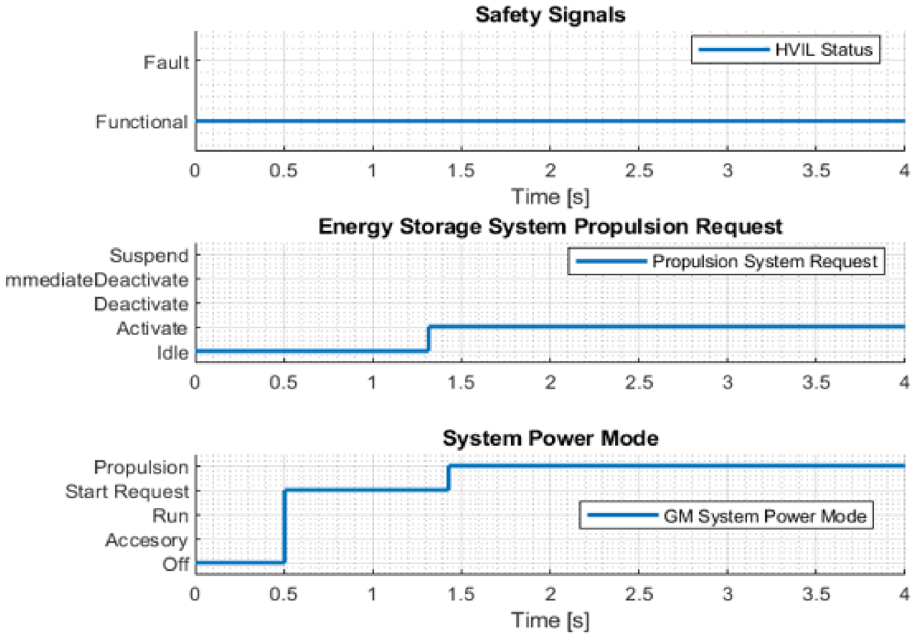


Figure 51: Correct Start-up Response

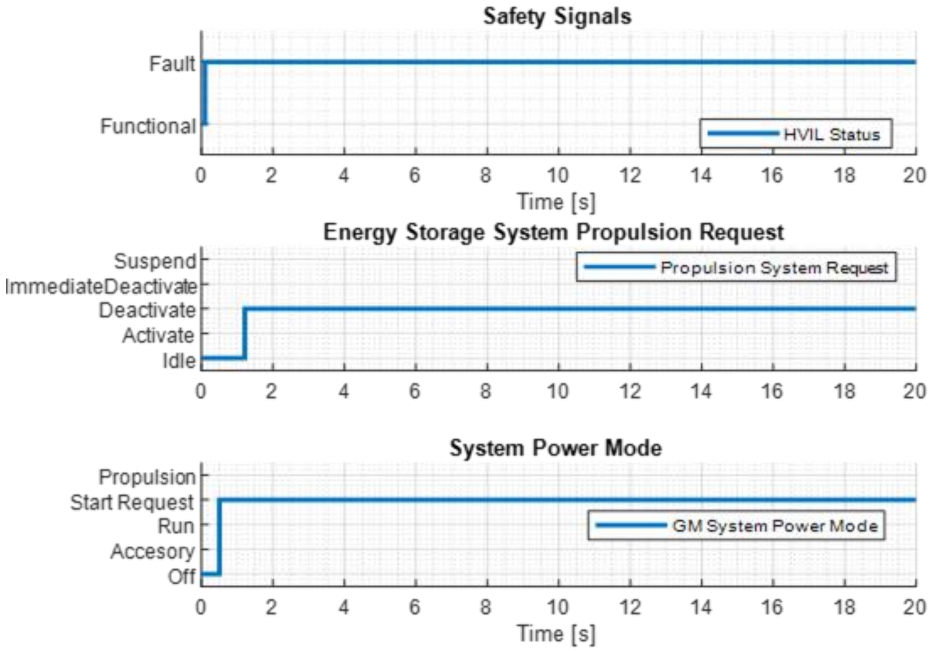


Figure 52: Fault on Start-up Response

In the normal start-up operation, after the vehicle start request, the propulsions system requests to activate, and then the GM system goes to propulsion mode. In the case where the HVIL is open, the propulsion system never requests activation but requests deactivation, ensuring the vehicle will not start. Validating the controller functions as intended in this scenario. This testing strategy was applied to all MIL test cases and allowed the team to evaluate a large amount of controller functionality in the MIL testing environment.

5.4 HIL Environment Development and Testing

The HIL development and testing work flow   is nearly identical to the MIL testing workflow. Because of this this section will document two major portions of HIL testing. The first being the HIL software and hardware workflow and then the decision making process for which features need to be evaluated in HIL specifically.

5.4.1 HIL Workflow

All HIL testing is done using SpeedGoat computing systems. SpeedGoat is directly integrated into Simulink. This allows for code generation, testing, and evaluation all to be done inside the MathWorks software environment. Model changes are necessary to implement HIL testing. As discussed in the earlier chapter, the I/O layer of the model is switched to the SpeedGoat CAN pack/unpack and read/write blocks and is disconnected from the team controller applications and virtual bus creator. The next step in model changes is to make the model run at a fixed interval of 0.0025 seconds. This time was chosen because it is also the step time of the controller. All of this is done automatically in Simulink Test Manager through the initialization script.

Moving on from model changes, the hardware setup will be discusse. The controller runs on SpeedGoat's Baseline M rapid prototyping controller and the plant model runs on SpeedGoat's Performance Real Time Target Machine or Performance Machine for short. The physical channel connections and the components on each channel can be seen below in Figure 53 and Table 20 below.

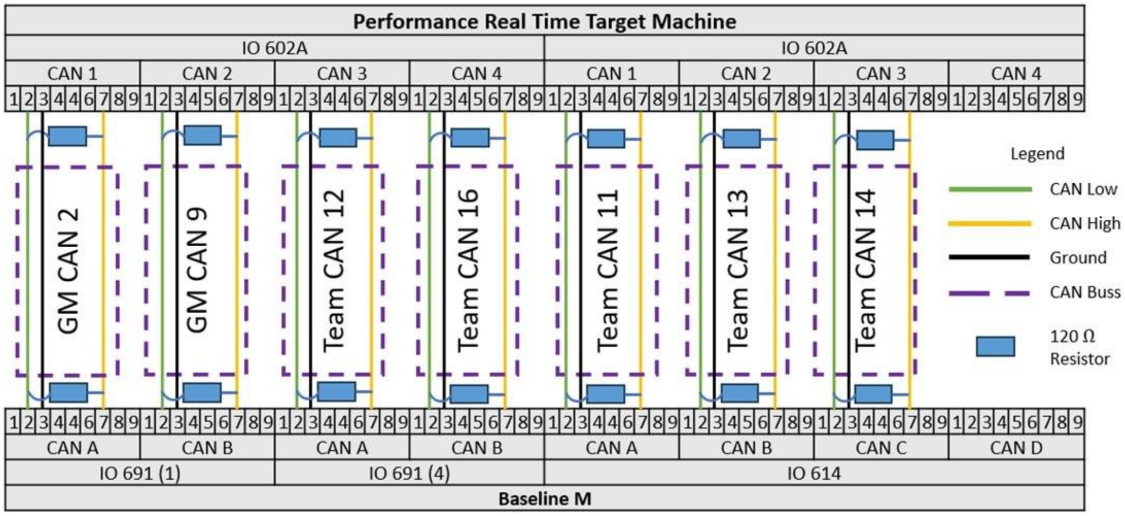


Figure 53: Team CAN Channel Mapping

Table 20: Team CAN Channel Description

CAN Channel Name	Baseline M Channel	Performance Target Channel	Signal Type	Baud Rate (kbps)	Components on the Channel
GM CAN 2	IO691(1) CAN A	IO602A CAN1	CAN-FD	500	Baseline M, GM CAN 2
GM CAN 9	IO691(1) CAN B	IO602A CAN2	CAN-FD	500	Baseline M, GM CAN 9
CAN 12	IO691(2) CAN A	IO602A CAN3	CAN-FD	500	Baseline M, DLCM, REM, PCAN-LIN, Isolation Monitor
CAN 11	IO614 CAN A	IO602B CAN1	CAN	250	Baseline M, MVEC Front, MVEC Rear, Rail Voltage Monitoring System
CAN 13	IO614 CAN B	IO602B CAN2	CAN	500	Baseline M, FEM, PCAN – Analog, PCAN - Digital
CAN 14	IO614 CAN C	IO602B CAN3	CAN	500	Baseline M, Autera, IPDS, Cohda, Data Logger, HMI

The workflow of testing is as follows. First, using Simulink's Real-Time app, the software application is connected to the Baseline M. Simlink's Real-Time app will then begin code generation and upload the controller application onto the machine; further, the controller will automatically re-initialize any time the Baseline M's power is cycled. After uploading the application to the Baseline M, the computer is then connected to the Performance Machine. Once connected to the performance machine, the HIL test suite can be run. This test suite is identical to the MIL test suite, but it is integrated with the Real-Time app and automatically code gens and uploads the plant model application for each test case and evaluates the results. Having all testing able to be done in a single software environment makes switching from MIL and HIL environments very straightforward, as only a few model changes and hardware connections need to be made.

5.4.2 Deciding Testing Environments

Deciding on what testing environment to use for specific test cases and features depended solely on requirements, safety level, and feature functionality. Figure 54 below lays out the decision making process for the feature testing environments. The first deciding factor on testing environments for software features is whether or not hardware is need for this test case. Next the safety level is considered. The higher the safety level the more testing environments the software feature will need for complete validation. Based on these two factors the correct testing environments are chosen.

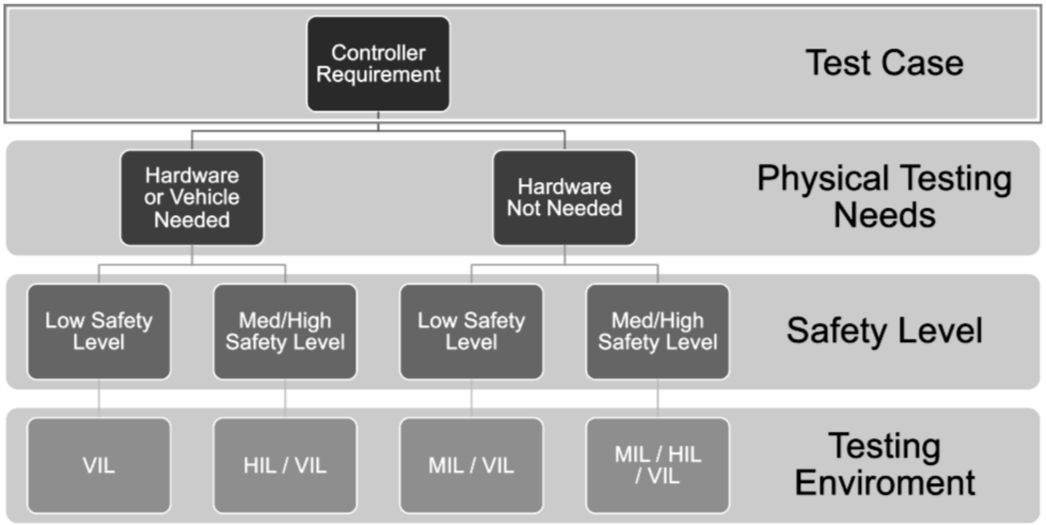


Figure 54: Testing Environment Decision Process

The table below holds every major feature/function of the team's controller its safety level and testing environment. For example, the controller function/feature GM_CAN_RX is responsible for reading and processing all incoming CAN signals from the GM system. This feature needs physical signals to actually be validated and is safety critical. Because of these two factors, the function will be tested both in HIL and VIL. This is the standard process used to make each decision in the table below.

Table 21: Controller Function and Testing Environment

Controller Function	Description	Safety Level	Testing Environment
GM_CAN_RX	GM CAN Signals Incoming	High	HIL / VIL
TEAM_CAN_RX	Team CAN Signals Incoming	High	HIL / VIL
I_DTC_READ	Incoming DTC Read	Medium	MIL / HIL/ VIL
GM_CAN_TX	GM CAN Signals Write	High	HIL/ VIL
TEAM_CAN_TX	Team CAN Signals Write	High	HIL/ VIL
O_DTC_WRITE	Declaring DTCs with team system faults	Medium	MIL / HIL/ VIL

A_HVREQ	High Voltage Request Control	High	MIL / HIL/ VIL
A_PRNDL	Transmission Position Control	High	MIL / HIL/ VIL
A_2PDP	Two Pedal Protection	High	MIL / HIL/ VIL
A_VEHSPD	Vehicle Speed Calculation	Medium	MIL / HIL/ VIL
A_CAVS	PCM <-> CAVs interface and CC/ACC Torque Req	High	HIL / VIL
A_EPD	LV Power Distribution	Medium	MIL / VIL
A_APM	Acceleration Pedal Map	Medium	MIL / VIL
A_TMS	Thermal Management Strategy (Aux pumps)	Low	VIL
A_TRQ	Torque Management	High	MIL / HIL / VIL
A_SOX	State of Power/Charge Management	High	MIL / HIL / VIL
A_FAULT	Fault Diagnostics	Medium	MIL / HIL / VIL
A_BKLSH	Backlash Control	Medium	MIL / VIL
A_REGEN	Regeneration Optimization	Medium	MIL / HIL / VIL
A_FLARE	Flare Management	Low	VIL

Chapter 6: Closing

6.1 Conclusion

In conclusion, this project carried out the architecture selection, model development, and controller testing process for a novel electric vehicle. This was all done as a part of the EcoCAR EV Challenge, where students from The Ohio State and Wilberforce University modified a RWD Cadillac LYRIQ to be more efficient and implemented autonomous features. An extensive architecture selection process was completed. Over 150,000 architectures were tested at low fidelity before the top architectures were strictly tested using a modified MathWorks model. The project presented the first iteration of the model development and controller testing process for the EcoCAR EV challenge. This process will be used through the remaining two years of the competition to implement more controller features and further improve on current functionality.

A plant model was developed to be used for controller testing and functionality in MIL, PIL, and HIL environments. The compartmentalized plant model was developed in a way that allows for easy modifications and improvements as higher fidelity testing may be necessary in some areas. The model uses physics and data based components to model the teams designed vehicle. Further, soft ECUs were implemented to validate controller functionality with vehicle components.

Finally, requirements and test cases were generated for controller features. This allowed for robust and repeatable testing to be implemented on the controller. All testing is done within a MathWorks software environment, allowing for easy linking of controller features, requirements, and test cases. All requirement generation, model development, controller development, and controller testing follow industry-level standards. This work helped propel The OSUWU EcoCAR team to a first place finish overall at the year one and year two competitions, and it has laid the groundwork for a sold controller development process in the future.

6.2 Future Work

There are two main tasks to be completed in the future. The first is model validation. Even though the model was developed using industry-level standards and data sent directly from suppliers, the model is only as good as the assumptions made. Now that the vehicle is up and running and has had over 250 miles of testing done on it with the team's new architecture, more accurate data can be collected and applied to the model. The second task is plant model changes. Well, this project has helped produce a fully functional controller and plant model. There are still more features planned to be implemented in the controller. Because of the new controller features, the plant model will have to be modified to test these new features.

As stated above with new data from the actual team vehicle current plant model components can be improved upon. For example, motor loss maps and battery limit maps can be improved upon. Further, physical parameters in many of the equations applied in the model can be improved upon. A simple example is the coast down coefficients or inertial values of many of the rotating components. All of these inputs into the model are a lot easier to validate with a real vehicle.

There are three major problems to tackle for the new model features and improvement. The first is a physics-based model of the axle disconnect dynamics. The axle disconnect was not used while the vehicle was moving in year 2. An accurate physics-based model of the axle disconnect dynamics needs to be made to test the team controller's ability to use the disconnect before testing on the vehicle. This will help prevent any major mechanical failures. The next model component that needs to be added is a detailed thermal model. Now that the vehicle is running with the team added components, the team can accurately model their thermal system and create a blackbox model of GMs thermal system based on data received from the vehicle. This will allow for a more nuanced thermal control strategy in the following years. Finally, autonomous vehicle systems will be a bigger part of the controller design process in the following years. A soft ECU of the team’s CAVs systems will need to be developed once the CAVs system is finalized. Completing these tasks would help bring about a more complete plant model and testing process, allowing for a high-function controller and continued team success.

Bibliography

1. Environmental Protection Agency. "Sources of Greenhouse Gas Emissions." https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.
2. Colorado Engineering, Inc. "Capabilities and Solutions: Sub-Systems Solutions." https://coloradoengineering.com/capabilities-and-solutions/sub-systems-solutions/.
3. Rumbaugh, J., Jacobson, I., & Booch, G. *The Unified Modeling Language Reference Manual*. 2nd ed. Boston, MA: Addison Wesley, 2004.
4. Holt, J., Perry, S. A., & Brownsword, M. *Model-Based Requirements Engineering*. Institution of Engineering and Technology (The IET), 2012.
5. Petkov, P. H., Slavov, T. N., & Kralev, J. K. *Design of Embedded Robust Control Systems Using MATLAB®/Simulink®*. Institution of Engineering and Technology (The IET), 2018.
6. Nibert, J., Herniter, M. E., & Chambers, Z. "Model-Based System Design for MIL, SIL, and HIL."
7. Spiegel, A. *A Soft ECU Approach to Develop a Powertrain Control Strategy*. B.S.
8. Harvey, D. R. "Willans Line Modeling for Powertrain Analysis and Energy Consumption of Electric Vehicles."
9. Accessed document. "https://vtechworks.lib.vt.edu/server/api/core/bitstreams/13442d6a-a0ec-4e0f-b5d3-457d49a48d64/content."
10. Ford Performance. "Eluminator Motor." https://performanceparts.ford.com/part/M-9000-MACHE.
11. EV Database. "Electric Vehicle Database." https://ev-database.org/
12. Environmental Protection Agency. "Data on Cars Used for Testing Fuel Economy." https://www.epa.gov/compliance-and-fuel-economy-data/data-cars-used-testing-fuel-economy.
13. Speedgoat. "Software Products and Services." https://www.speedgoat.com/products-services/software.
14. MacAdam, C. C. "An Optimal Preview Control for Linear Systems." *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 3, Sept. 1980.
15. MacAdam, C. C. "Application of an Optimal Preview Control for Simulation of Closed-Loop Automobile Driving." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, June 1981.
16. MacAdam, C. C. *Development of Driver/Vehicle Steering Interaction Models for Dynamic Analysis*. Final Technical Report UMTRI-88-53. Ann Arbor, MI: The University of Michigan Transportation Research Institute, Dec. 1988.
17. *Lithium-Ion Batteries: Basics and Applications*. 1st ed.
18. *Hybrid Electric Vehicles: Energy Management Strategies*. (SpringerBriefs in Control, Automation, and Robotics).

19. MathWorks. "Vehicle Body 1DOF (Longitudinal)." https://www.mathworks.com/help/vdynblks/ref/vehiclebody1doflongitudinal.html.
20. MathWorks. "Longitudinal Wheel." https://www.mathworks.com/help/vdynblks/ref/longitudinalwheel.html.
21. Bringmann, E., & Krämer, A. "Model-based Testing of Automotive Systems."
22. Shahbakhti, M., Li, J., & Hedrick, J. K. "Early Model-Based Verification of Automotive Control System Implementation."
23. Ohata, A. "Improving Model-based Design for Automotive Control Systems Development." Toyota Motor Corporation.
24. Korssen, T., & Dolk, V. "Systematic Model-Based Design and Implementation of Supervisors for Advanced Driver Assistance Systems."
25. Sovey, S. "Utilization of a Programmable Node in a 'Black-Box' Controller Area Network in Conjunction with a Serial Gateway to Prototype Control of a P0+P4 Gage."
26. Satra, M. K. "Hybrid Electric Vehicle Model Development and Design of Controls Testing Framework."
27. Cal. Code Regs. Tit. 13, § 2703 - Emission Testing Requirements, State Regulations, US Law, LII, Legal Information Institute. https://www.law.cornell.edu/regulations/california/13-CCR-2703

Appendix A. Vehicles Included in Market Research

Model	Year	Manufacturer	Drivetrain
Mustang Mach-E GT AWD	2021	Ford	AWD
Mustang Mach-E GT AWD	2021	Ford	AWD
Mustang Mach-E AWD Extended	2021	Ford	AWD
Q4 e-tron quattro	2022	Audi	AWD
Q4 e-tron Sportback quattro	2022	Audi	AWD
e-tron quattro	2022	Audi	AWD
e-tron	2021	Audi	AWD
e-tron Sportback quattro	2022	Audi	AWD
e-tron Sportback	2021	Audi	AWD
e-tron S Sportback (20" wheels)	2022	Audi	AWD
e-tron S (20" wheels)	2022	Audi	AWD
e-tron S Sportback (21" or 22" wheels)	2022	Audi	AWD
e-tron S (21" or 22" wheels)	2022	Audi	AWD
iX xDrive50 (20" Wheels)	2022	BMW	AWD
iX xDrive50 (22" Wheels)	2022	BMW	AWD
iX xDrive50 (21" Wheels)	2022	BMW	AWD
Lyriq	2024	Cadillac	AWD
Mustang Mach-E AWD California Route 1	2022	Ford	AWD
Mustang Mach-E AWD	2022	Ford	AWD
Mustang Mach-E AWD Extended	2022	Ford	AWD
Mustang Mach-E GT AWD	2022	Ford	AWD
Mustang Mach-E GT Performance	2022	Ford	AWD
Ioniq 5 AWD (Long Range)	2022	Hyundai	AWD
I-PACE EV400	2021	Jaguar	AWD
Model Y Long Range AWD	2021	Tesla	AWD
Model Y AWD	2022	Tesla	AWD
Model Y (Long Range) AWD	2022	Tesla	AWD
Model Y Performance AWD	2022	Tesla	AWD
Model Y Performance AWD	2021	Tesla	AWD
Model X Long Range Plus	2021	Tesla	AWD
Model X AWD	2022	Tesla	AWD

Model X Plaid (20" Wheels)	2022	Tesla	AWD
Model X Performance (20" Wheels)	2021	Tesla	AWD
Model X Plaid (22" Wheels)	2022	Tesla	AWD
ID.4 AWD Pro	2022	Volkswagen	AWD
ID.4 AWD Pro	2021	Volkswagen	AWD
ID.4 AWD Pro S	2022	Volkswagen	AWD
ID.4 Pro S AWD	2021	Volkswagen	AWD
C40 Recharge Twin	2022	Volvo	AWD
Mustang Mach-E AWD	2021	Ford	AWD
R1S	2022	Rivian	Part-Time 4WD
XC40 Recharge Twin	2022	Volvo	FWD
Bolt EUV	2022	Chevrolet	FWD
Kona Electric	2022	Hyundai	FWD
Mustang Mach-E RWD California Route 1	2021	Ford	RWD
Mustang Mach-E RWD	2021	Ford	RWD
Mustang Mach-E RWD Extended	2021	Ford	RWD
EQS450+	2022	Mercedes-Benz	RWD
Lyriq	2022	Cadillac	RWD
Mustang Mach-E RWD	2022	Ford	RWD
Mustang Mach-E RWD California Route 1	2022	Ford	RWD
Mustang Mach-E RWD Extended	2022	Ford	RWD
Ioniq 5 RWD (Long Range)	2022	Hyundai	RWD
Ioniq 5 RWD	2022	Hyundai	RWD
Model Y RWD	2022	Tesla	RWD
ID.4 Pro	2022	Volkswagen	RWD
ID.4 Pro S	2022	Volkswagen	RWD
Kona Electric	2021	Hyundai	RWD
Model Y Standard Range RWD	2021	Tesla	RWD
ID.4 Pro	2021	Volkswagen	RWD
ID.4 1st	2021	Volkswagen	RWD
ID.4 Pro S	2021	Volkswagen	RWD
XC40 Recharge	2021	Volvo	RWD
Niro Electric	2021	Kia	RWD

Appendix B: List of Abbreviations

BEV	Battery Electric Vehicle
AVTC	Advanced Vehicles Technology Competition
EcoCAR EV	EcoCAR Electric Vehicle Challenge
GM	General Motors
CAV	Connected and Automated Vehicle
PCM	Propulsion Controls and Modelling
VTS	Vehicle Technical Specifications
REM	Rear Electric Motor
V2X	Vehicle to X
SOC	State of Charge
HIL	Hardware-in-the-Loop
ESS	Energy Storage System
TMS	Thermal Management System
PSC	Propulsion Supervisory Controller
EM	Electric Motor
NVH	Noise, Vibration, and Harshness
IVM	Initial Vehicle Movement
HVIL	High Voltage Interlock Loop
CAN	Controller Area Network
HV	High Voltage
ECU	Electronic Control Unit
sECU	Soft Electronic Control Unit
EDU	Electric Drive Unit
PIL	Processor in the Loop
MIL	Machine in the Loop
VIL	Vehicle in the Loop
RWD	Rear Wheel Drive
AWD	All Wheel Drive
FWD	Front Wheel Drive
CAD	Computer Aided Design
DOE	Department of Energy
ETRS	Electronic Transmission Range Select
PRNDL	Park Reverse Neutral Drive Low
GHG	Greenhouse Gas
UDDS	Urban Dynamometer Driving Schedule
HWFET	Highway Fuel Economy Test
CAN	Controller Area Network
LIN	Local Interconnect Network
I/O	Input and Output