THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

EXPLORING THE EFFECTS OF A VOLTAGE BASED SHIFTING ALGORITHM ON A HYBRID-ELECTRIC VEHICLE LIMITED BY BACK EMF

KYLE BROWN
SPRING 2018

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Electrical Engineering and Computer Science
with honors in Electrical Engineering

Reviewed and approved* by the following:

Gary Neal
Lecturer of Mechanical and Nuclear Engineering
Thesis Supervisor

Julio Urbina
Associate Professor of Electrical Engineering and Computer Science
Honors Adviser

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

In hybrid vehicle architectures where the engine is directly coupled to the motor, the two components must share the same speed. In a vehicle where the motor speed is limited by back EMF issues, this can greatly limit vehicle performance. This thesis explores a method of improving vehicle performance in such hybrid architectures.

The maximum speed of the motor varies with voltage. A real-world bench test is conducted to map the top speed limits of this motor at different voltages. With this real-world data, an algorithm is designed that controls vehicle shifts to maintain higher engine and motor speeds as much as possible.

The algorithm is tested using a Simulink vehicle model of the hybrid vehicle architecture described above. Testing shows that the algorithm successfully maintains the motor speed below its specified limit at all times. Simulations also show an improvement in vehicle acceleration using the new shifting algorithm. The vehicles IVM-60 mph time is decreased by 0.04 to 0.08 seconds, and its 50 – 70 mph passing times is decreased by 0.26 to 0.30 seconds.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Appreciation and thanks goes to

Mr. Gary Neal

for his dedication to the Penn State Advanced Vehicle Team and his role in the creation of this thesis

The members of The Pennsylvania State Advanced Vehicle Team

for inspiring the work contained in this thesis

Chapter 1

Introduction

Hybrid-electric vehicles are becoming an increasingly popular option for consumers due to increased fuel economy and government incentives. Many consumers have been hesitant to switch to hybrids due to a lack of performance in such vehicles. As a result, automotive manufacturers have begun to make hybrids with a larger focus on performance. However, new hybrid vehicle innovations bring with them new challenges to overcome. One of these challenges is utilizing an internal combustion engine (ICE) and an electric motor together to produce torque to the wheels of the vehicle. To accomplish this, a variety of hybrid vehicle architectures have emerged. The Pennsylvania State University Advanced Vehicle Team (PSU AVT) chose one of these architectures for its EcoCAR3 Camaro, and discovered a unique problem caused by back electromotive force (EMF) in its electric motor which limits vehicle performance. To remedy this problem, a software solution was created.

1.1 Vehicle Architecture

PSU AVT chose to design a pre-transmission parallel hybrid vehicle [1]. The location of major components in this architecture can be seen in Figure 1. The first component in the powertrain is a GM 4-cylinder turbocharged Internal Combustion Engine (ICE), which is directly coupled through a splined shaft to the YASA P400 motor. The shaft then connects to the torque converter, which is also the input to the vehicles 8-speed GM transmission. Finally, the output of the transmission connects to differential via drive shaft, and the differential powers the rear wheels of the vehicle. In addition to the powertrain, there is an A123 Energy Storage System (ESS) that powers the electric motor.

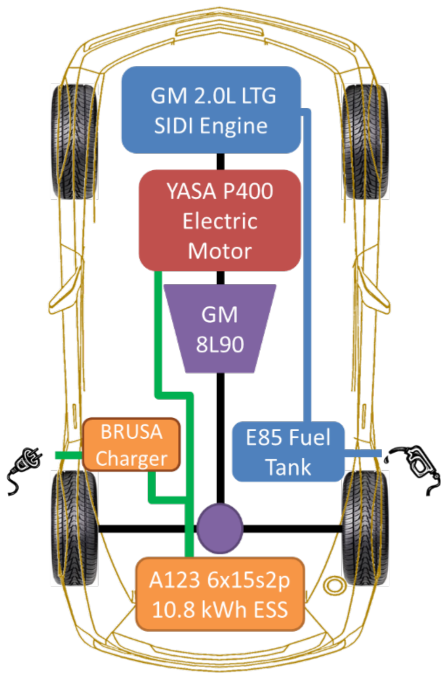


Figure 1: Diagram of PSU AVT's Parallel Pre-Transmission Hybrid Electric Vehicle [1].

As stated, the engine and motor of this vehicle are directly coupled to one another via a splined shaft. Therefore, during operation the engine must always be on, and both components will maintain the same angular velocity. Thus, the engine-motor system is speed-limited by the component with the lowest angular velocity capabilities.

1.2 Back EMF Limitations

In the PSU AVT Camaro, the motor has back EMF limitations that cause the maximum speed of it to be much lower than that of the engine. The vehicle uses an axial flux AC motor. Like all motors, this motor produces back EMF at all speeds. This back EMF increases with speed, and eventually the voltage created will be as high as the ESS voltage. At this point the motor can no longer be controlled by the motor controller to produce more torque or go to higher speeds. For this reason, bus voltage directly affects the speed that a motor can achieve. Furthermore, this motor was designed by YASA to operate at 800V. The maximum voltage of the vehicles ESS is 325V. Therefore, the motor was designed to operate in a system with a much higher bus voltage. It was also designed to achieve 8000 RPM [2]. However, motor speed is proportional to the bus voltage available to it, so by decreasing the voltage of the motor from its expected 800 V to the 325 V the vehicle can support, the maximum speed of the motor is greatly limited.

Chapter 2

Determining System Limitations

To design an improved algorithm, a test was conducted to determine the real-world limitations of the motor system. This test involved the YASA motor, Rhinehart motor controller, and an ABC 150 high voltage power supply, used to simulate the ESS at different voltages. The goal of this test was to determine the maximum angular velocity of the motor within the possible voltage range of the ESS.

2.1 Test Setup and Procedure

A diagram of the test setup can be seen in Figure 2. The ABC 150's high voltage DC lines are connected to the motor controller. The motor controller converts DC to three phase AC, and those AC lines are run directly to the motor. The motor controller is controlled via Controller Area Network (CAN) messages. For this test, the CAN messages are sent from a computer using the CANoe software tool. This tool allows the user to create a graphical interface for sending commands to the motor, as well as seeing motor states and speed. The tool is also used to log all CAN messages during the test for analysis after the test. The motor controller is powered using a standard 12 V car battery. The low voltage pin assignments of the motor controller used in this test can be found in Appendix A [3].

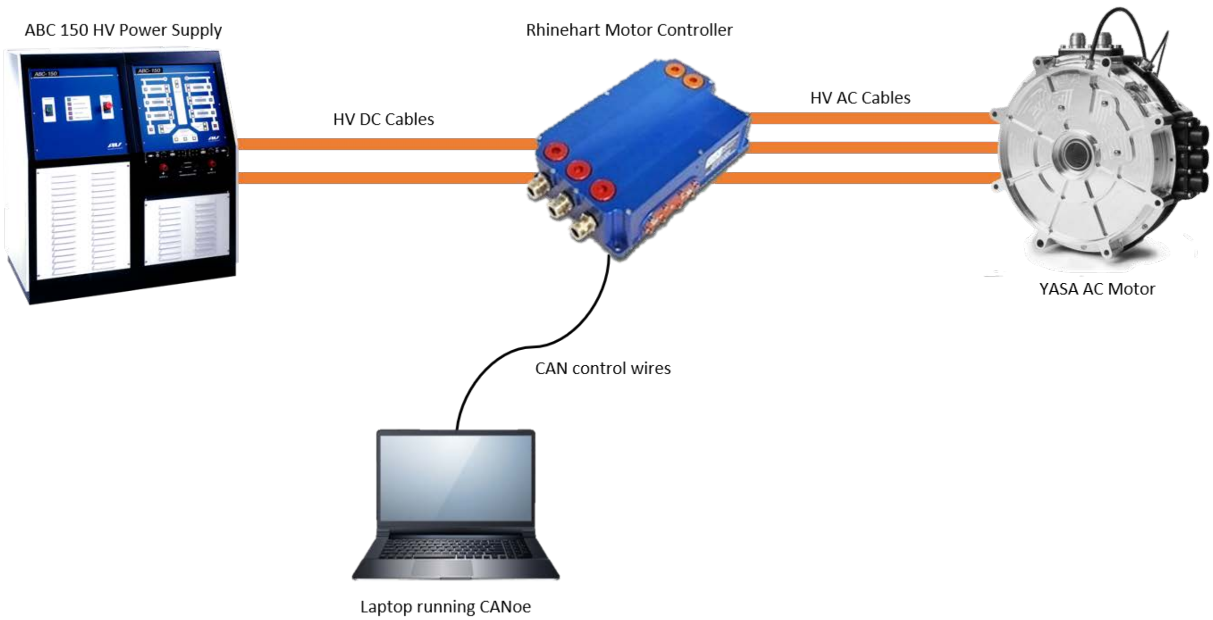


Figure 2: Electrical setup for motor bench test performed. Orange lines represent high voltage. Black lines represent low voltage signals.

In addition to the electrical setup, a coolant loop was created to keep the motor and motor controller within operating temperatures during operation. A diagram of the coolant loop can be seen in Figure 3. The pump used is from a Chevrolet Volt, and it is powered by the same 12V car battery used to power the motor controller. The coolant used is OptiCool H, as required by the motor.

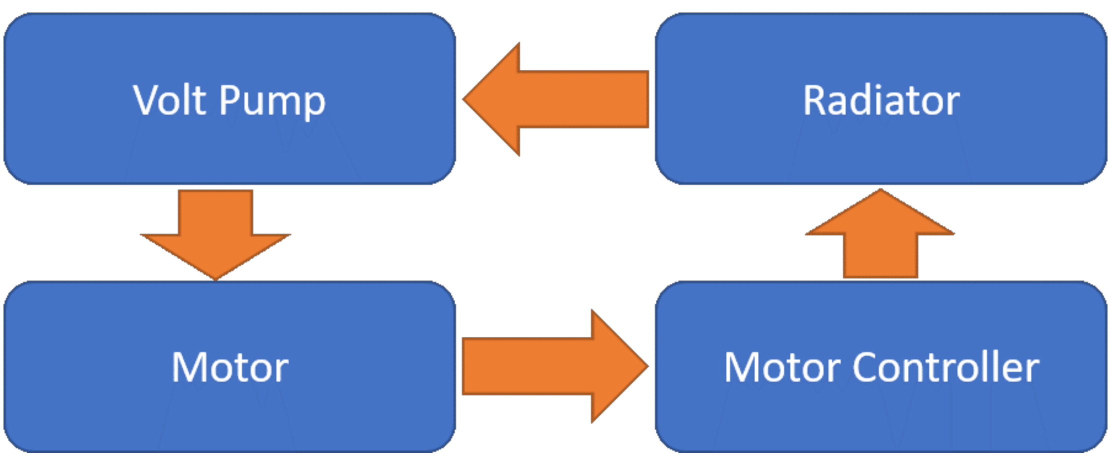


Figure 3: Coolant flow for motor bench test.

Once setup was complete, the ABC 150 machine was powered on and set to a specific voltage. At this voltage the motor was given increasing torque commands until the increase in torque no longer caused an increase in motor speed. Figure 4 shows the motor speed curve during one of these tests. The speed of the motor at this point represents the maximum speed that it can achieve at the given voltage. The motor was then brought back to rest and the voltage was changed to the next set point. The full procedure can be found in Appendix A.

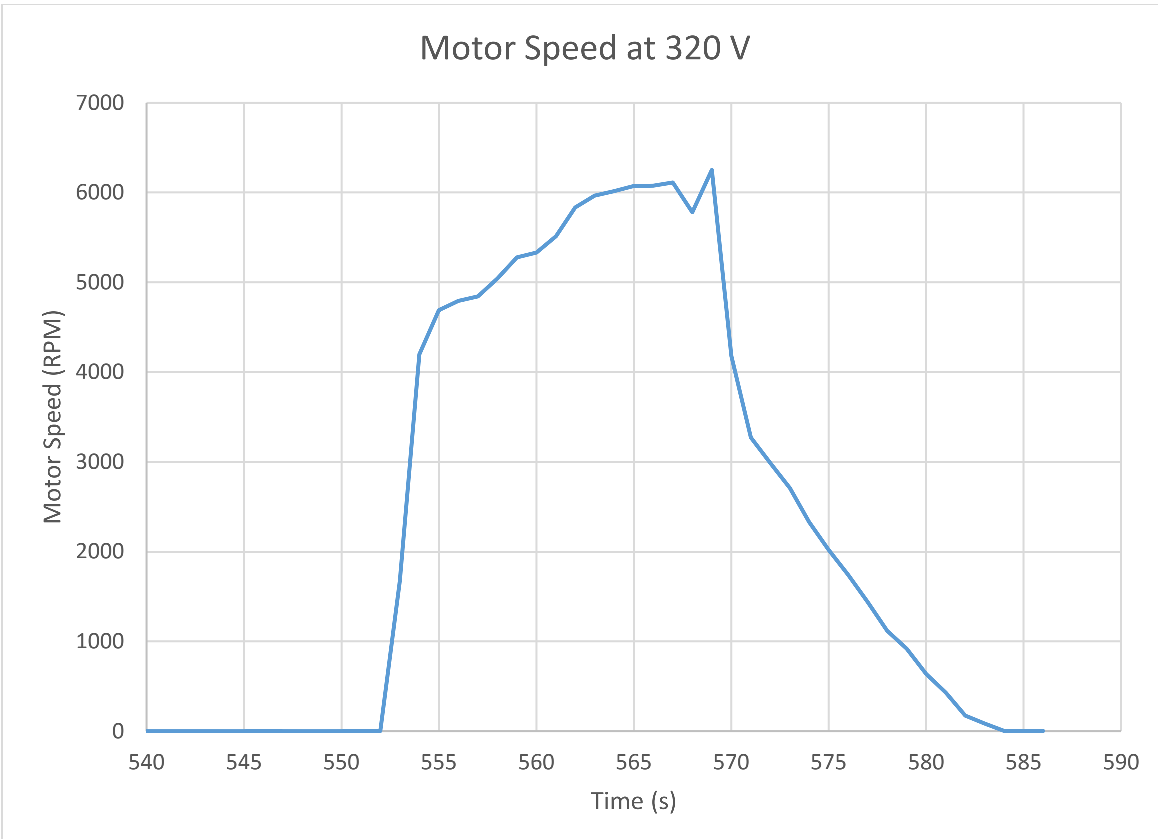


Figure 4: Motor speed curve for bench test at 320 V. The motor speed increases until it gets to 6050 RPM.

2.2 Results

Once testing was completed, the CAN logs were examined to determine the maximum speed achieved at each voltage. Table 1 shows the results. As expected, the maximum speed achieved increases with the system voltage.

Table 1: Maximum speed of motor system within ESS voltage range

Voltage (Volts)	Maximum Motor Speed (RPM)
325	6150
320	6050
300	5500
280	5100
260	5000
240	4500
225	4300

Chapter 3

Algorithm Development

Once the limitations of the motor system were known, the next step was to design a control algorithm to mitigate this back EMF system issue. The goal of this algorithm is to increase vehicle performance, including acceleration, as much as possible while respecting the limitations of the vehicle. This was accomplished through modifying the shifting algorithm of the vehicle. Under normal operation, the vehicle will shift based on a shift map with inputs current gear, speed, and accelerator pedal position. This map is extensively calibrated by the manufacturer to provide the best driver feel possible. Therefore, another goal of the algorithm was to do minimal modifications to the map to maintain the best driver feel under normal driving conditions.

To achieve these goals, the minimum gear functionality of the transmission was used. A CAN message is sent to the transmission controller giving it a minimum gear request. In response, the transmission controller will not allow the transmission to be in a gear lower than the one requested. If it is in a lower gear at the time of the request, an upshift will be triggered. This is effective for achieving our goals since it maintains the stock map as much as possible, but triggers an upshift, and lowers the motor speed, when needed.

Figure 5 shows a block diagram of the process used to calculate the minimum gear request. This calculation is done inside the hybrid supervisory controller (HSC) and sent to the transmission controller. The algorithm will take the vehicle speed at the wheels and convert it from kilometers per hour (km/h) to revolutions per minute (RPM). Next, the algorithm will multiply the revolutions per minute at the wheel by the differential ratio to get the drive shaft speed at the output of the transmission. Finally, that output transmission speed is multiplied by the gear ratio for all eight gears of the transmission to get the input shaft speed. This gives the potential motor speed in every possible gear. Since the algorithm only introduces an upshift at high motor speeds, we can ignore the effects of the torque converter, as it will always be locked (acting as a 1:1 ratio) at high speeds. Lastly the algorithm compares this to the maximum speed limit and chooses the lowest gear that produces a motor speed below that limit. Appendix B shows the Simulink code for this algorithm.

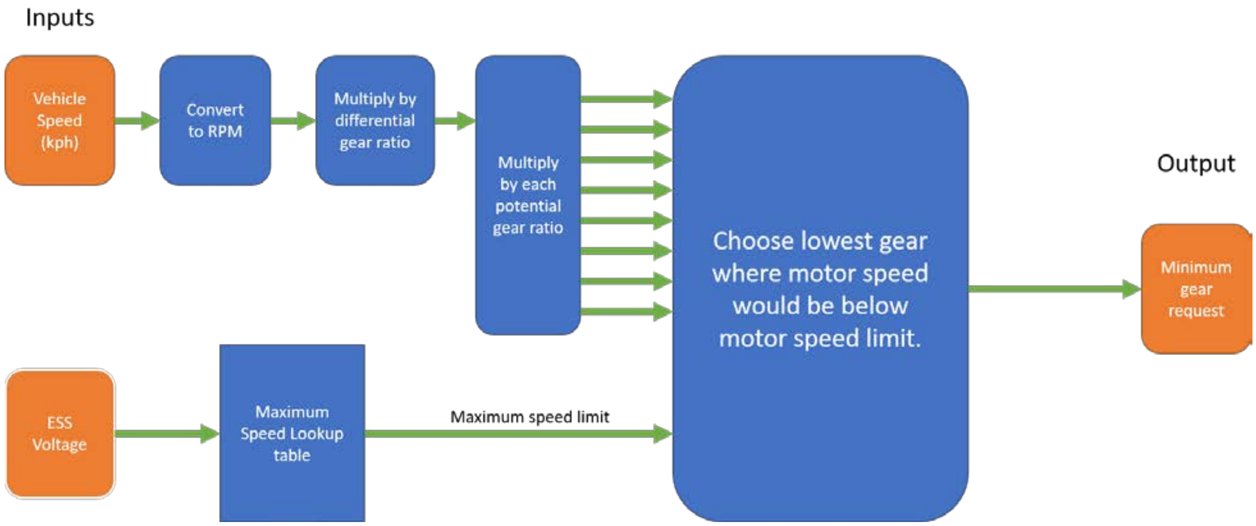


Figure 5: Algorithm to calculate minimum gear request for transmission.

The maximum speed limit mentioned above is from the bench test conducted earlier. A lookup table was created with the data in Table 1, with a built-in factor of safety of 300 RPM. The input to this table is the current pack voltage, and using linear interpolation, the table outputs the maximum limit for the speed of the motor at that voltage. Figure 6 shows the maximum speed limit for all potential ESS voltages.

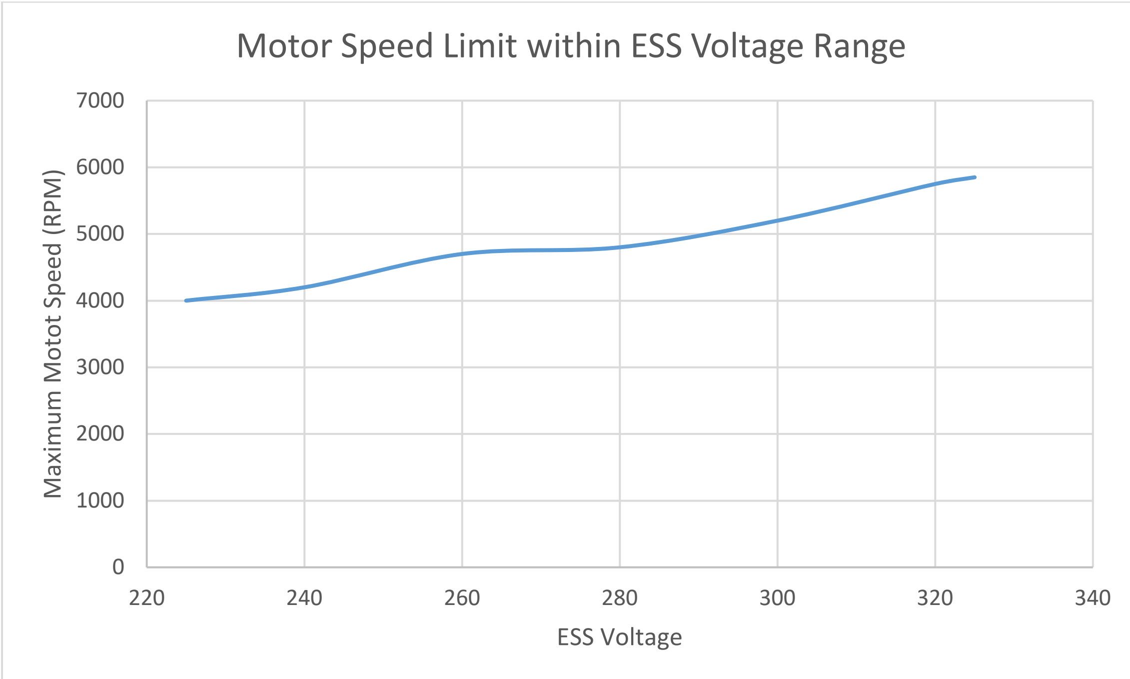


Figure 6: Maximum motor speed limit over ESS voltage range. These limits include the factor of safety added into the algorithm.

Chapter 4

Validation and Performance Evaluation

Upon completion of development, the algorithm was put through various tests. The goal of these tests was twofold: to validate the algorithm works as intended and to determine if any performance improvements were made. To run these tests, MATLAB Simulink was used. Since the algorithm is developed using Simulink, it can be inserted into a Simulink model and run to obtain various estimated parameters of the vehicle.

4.1 Simulation Environment

The simulation model is created in Simulink and composed of three main parts, shown in Figure 7. The driver model simulates the role of a human in a vehicle. This model takes speed and grade information from the given drive trace and turns it into an acceleration and brake pedal command. The control algorithm represents all the vehicle control code that runs on the HSC in the vehicle. It takes inputs from the driver and the vehicle and uses them to control various parts of the vehicle. The vehicle model represents a virtual vehicle. It takes inputs from the control algorithm and determines vehicle parameters such as speed.

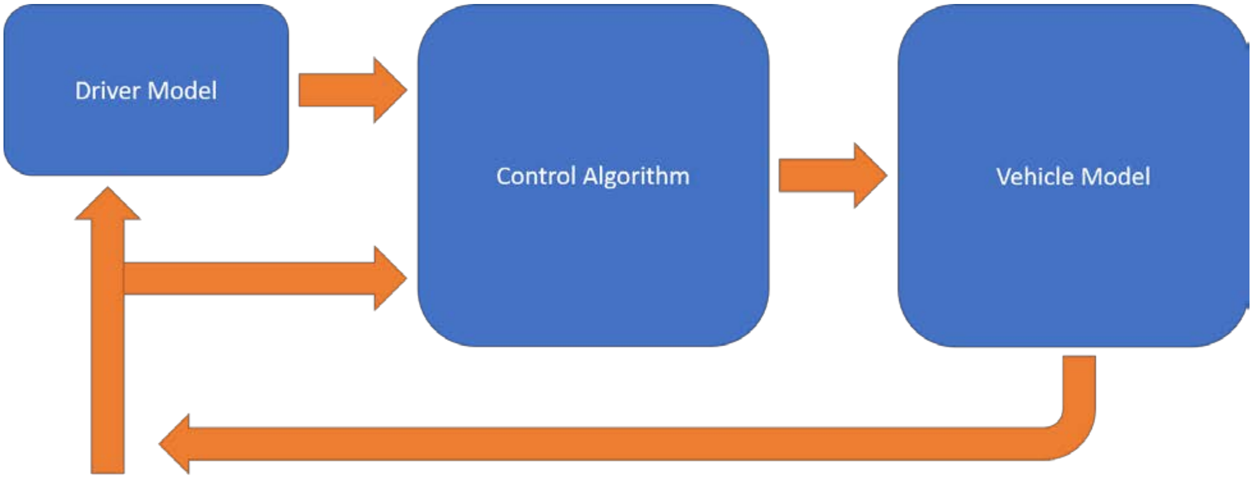


Figure 7: Diagram of high-level portions of simulation model. The arrows represent data flow between them.

4.1.1 Control Algorithm

The purpose of the control algorithm is to control the hybrid operation of the vehicle. The code found in the portion of the vehicle model is the same code that is compiled into C code and flashed onto the vehicle's HSC. This code handles interaction between different stock components integrated into the vehicle and controls any non-stock components added by PSU AVT. This is also the portion of the code that contains the voltage based shifting algorithm that is the focus of this thesis. Table 2 shows a list of all the high-level portions of the control algorithm, along with a brief description of their purpose.

Table 2: Subsystems contained in control algorithm with brief description.

Control Algorithm Subsystem	**Description**
Diagnostics	Validates input signals to ensure their values are within an expected range and sets faults if they are not.
Correctional Action	Modifies powertrain component torque limits based on faults detected by the diagnostics subsystem.
Torque Split	Responsible for determining the portion of total requested torque that will be produced by the engine, and the portion produced by the motor. It essentially controls the hybrid operation of the vehicle.
Transmission Gear Selection	This is the system described in the Algorithm Development chapter.
Startup and Shutdown	Controls the startup and shutdown procedure of all vehicle components.

Vehicle Indicators	Controls vehicle indicators to the driver.
Thermal Loop	Controls the thermal systems of the motor and motor controller. It keeps those components within their specified operating temperatures.
Regenerative Braking	Determines the amount of regenerative braking requested during a braking event.
Torque Reduction	Modifies torque requests during a shift event to provide smoother shifts to the driver.

4.1.2 Vehicle Model

The vehicle model simulates the vehicle. It includes physics-based models for many of the major powertrain components in the vehicle, as well as a vehicle dynamics model. Based on the inputs from the control algorithm, this model can determine the speed of the vehicle, as well as the transmission gear, and fuel consumed by the engine. This tool allows algorithms to be tested for functionality and estimated performance prior to vehicle testing. This portion of the code also contains the transmission model, which is the largest portion of the vehicle effected by the voltage based shifting algorithm. Table 3 shows the subsystems of the vehicle code, along with a brief description of their functionality.

Table 3: Subsystems of vehicle model along with brief description.

Vehicle Model Subsystem	**Description**
Motor	Represents the YASA motor. Draws current from the battery model and converts it into torque.
Engine	Represents the GM ICE. Simulates the usage of E85 to create torque.
Battery Pack	Represents the ESS. Provides current to the motor and keeps track of the simulated state of charge and energy consumed.
Impeller	Represents the torque converter in the vehicle. Combines torque from engine and motor for the input of the transmission.
Transmission	Represents the GM 8-speed transmission. Changes gears and therefore torque/speed ratio’s.
Vehicle dynamics	Takes transmission output torque and road conditions (grade, headwind, braking torque) to determine vehicle speed.

Auxiliary systems	Represents the current draw caused by the component controllers and vehicle accessories (Radio, AC, etc.).

4.2 Modeling Results

To accomplish the first goal of testing, the algorithm must successfully keep the motor speed below the limits that have been defined. The vehicle model, with the voltage based shifting algorithm, was run on both the HWFET and US06 City drive cycle. The HWFET cycle is a highway cycle with the vehicle mainly at high speeds, while the US06 City cycle is a city drive cycle where the vehicle is at lower speeds. The results of these cycles can be seen in Figure 8 and Figure 9. In both cycles, the motor speed stayed below the maximum motor speed defined by the algorithm. Therefore, the algorithm accomplishes its first goal by maintaining the motor at a safe operating speed.

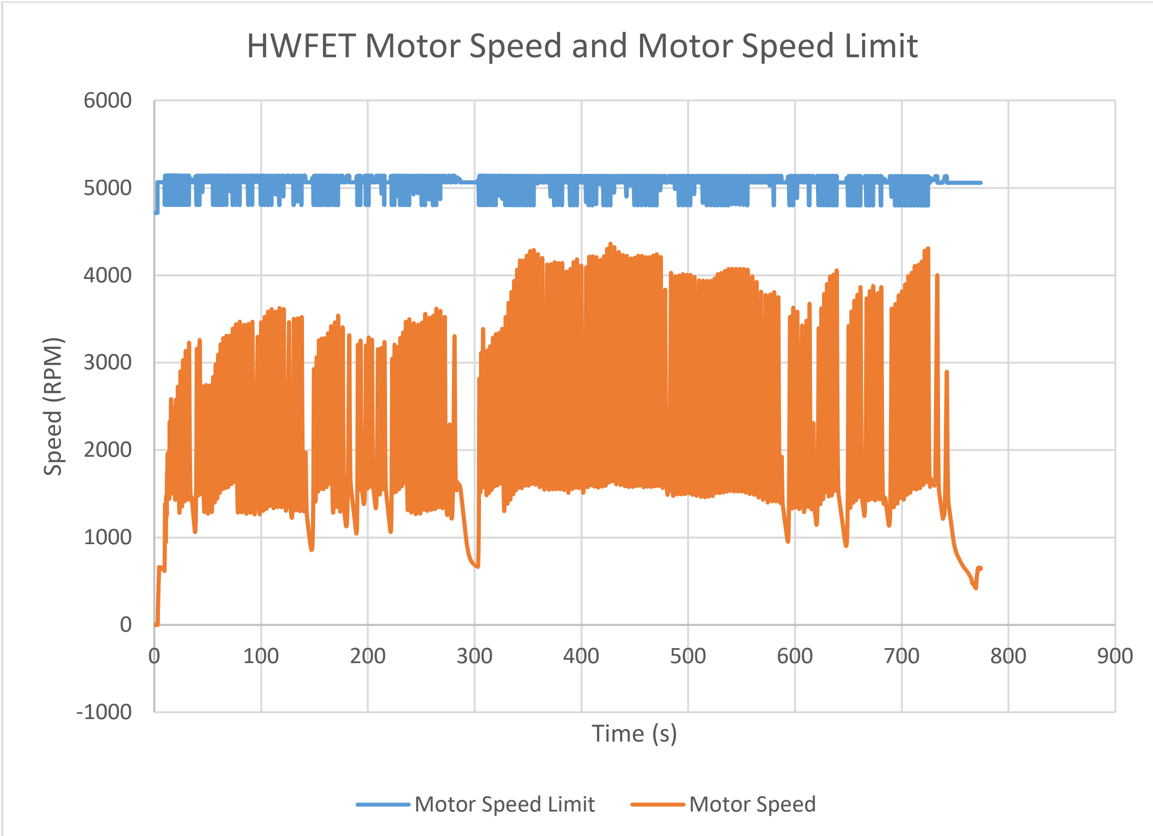


Figure 8: Graph of motor speed limit and motor speed during HWFET cycle. At all times the motor speed stays below the limit.

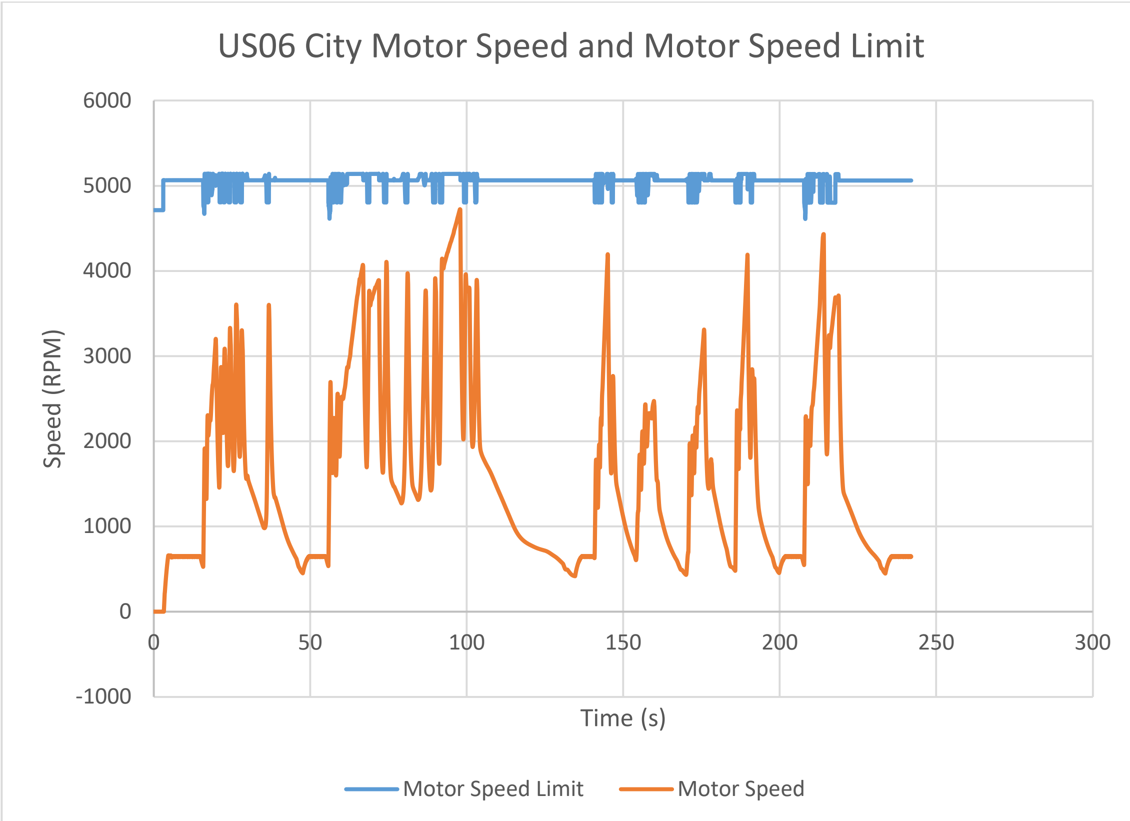


Figure 9: Graph of motor speed limit and motor speed during US06 City cycle. At all times the motor speed stays below the limit.

After validating the functionality of the algorithm, the performance of this new algorithm was tested. To do so, the voltage based shifting algorithm was compared to a safety only algorithm that ignores the ESS voltage and always maintains the motor speed below 4000 RPM. To test for potential improvements, both algorithms were simulated on a performance drive cycle designed to test IVM-60 mph times, as well as 50-70 mph passing times. These parameters allow the acceleration improvements caused by the voltage based shifting algorithm to be quantified. Table 4 shows the results of this comparison. In addition to full state of charge, the same tests were run across the entire operational ESS SOC range. Figure 10 and Figure 11 show the results of these tests.

Table 4: Performance parameter comparison for two shifting algorithms at full SOC.

Parameter	Voltage based shifting algorithm	Safety only shifting algorithm
IVM – 60 mph	5.29 sec	5.33 sec
50 – 70 mph	2.50 sec	2.78 sec

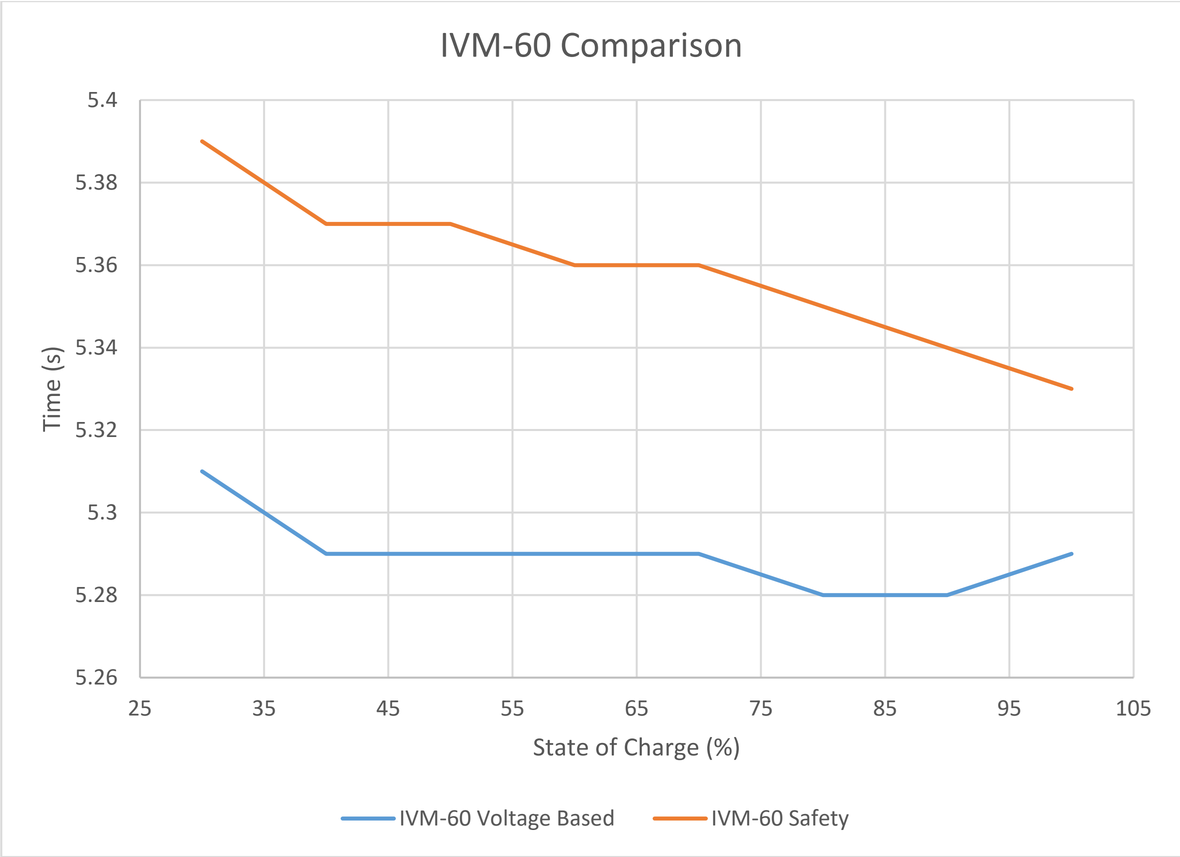


Figure 10: Comparison of IVM to 60 mph times over ESS state of charge. Voltage based shifting times were always lower than the safety algorithm.

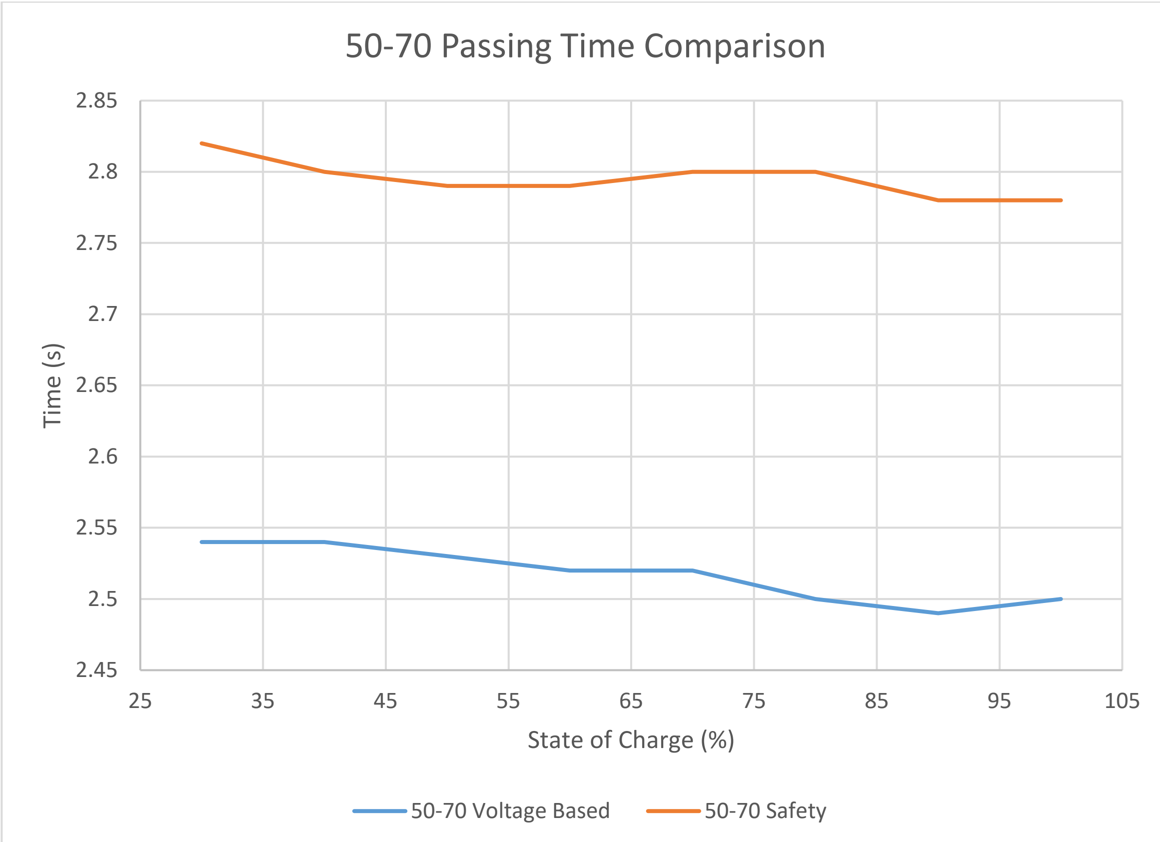


Figure 11: Comparison of 50-70 mph passing times over ESS state of charge. Voltage based shifting times were always lower than the safety algorithm.

In all performance aspects, the voltage based shifting algorithm performed better than the safety algorithm. At full SOC, the voltage-based algorithm had a 0.04 second faster IVM-60 time and a 0.28 second faster 50-70 time. This trend holds true when looking at the IVM-60 and 50-70 times over the entire ESS voltage range as well. An interesting observation is that there is a larger increase in performance in the 50-70 times than in the IVM-60 times. This is likely because accelerating from 50-70 quickly has to do with downshifting to a lower gear to provide large amounts of torque. Therefore, using the voltage-based algorithm the vehicle can shift to a lower gear than using the safety algorithm.

Chapter 5

Conclusions

In a hybrid architecture like PSU AVT's Camaro, you will always have performance limitations when using a motor limited by back EMF and an ESS that does not fully support that EMF. However, with better understanding of those limitations, improvements can still be made. A motor bench test was performed to map the maximum speed limitation of the motor over the possible voltage range of the vehicle's ESS. This test showed that as voltage increases, so does the maximum speed of the motor. It also showed that the maximum speed of the motor varies greatly over the ESS voltage range, from 4300 RPM to 6150 RPM.

The limitations determined from the bench test were used to create a new shifting algorithm in the vehicle code. This algorithm allows the vehicle to get closer to the maximum speed limit of the motor according to the current ESS voltage. This provides more time in a high-speed state, where the engine can provide more torque.

To test the voltage based shifting algorithm developed, simulation was used. A simulation model was constructed to represent PSU AVT's Camaro. This model can then be run on different drive cycles to obtain estimated performance and fuel economy. This model was first used to verify the algorithm was functioning correctly. This was done by showing that the motor speed stays below its limit at all times. Then the model was used to compare vehicle performance when using the new voltage based shifting algorithm, as well as a safety algorithm that did not take voltage into account and always utilized the minimum motor speed limit.

The goal of the voltage based shifting algorithm discussed was to provide the best performance possible with the given limitations. This goal has been achieved. Using the voltage based shifting algorithm improvements were seen in both initial vehicle acceleration (IVM-60 mph) and high-speed acceleration (50-70 mph). The IVM – 60 mph improvements ranged from 0.04 seconds to 0.08 seconds. The 50 – 70 mph improvements ranged from 0.26 to 0.3 seconds. While these improvements seem small, hundredths of seconds are very important in motor sports where vehicles are constantly decelerating and accelerating.

Chapter 6

Future Work

This thesis provides a method of improving vehicle performance in a hybrid with a clutch-less pre-transmission hybrid architecture, and motor speed limitations. To validate the algorithm developed, simulation was used. Unfortunately, simulation lacks certain parameters that can only be found in the real world. Therefore, it should be tested in vehicle to determine the real-world benefit of the algorithm. However due to time constraints as well as limited vehicle availability, in vehicle testing was not possible for this thesis. Therefore, prior to use in a vehicle, this testing should be completed.

Appendix A

Motor Bench Test Pinouts and Procedure

MCU/Motor Bench Test Plan

Setup

The following components will be necessary:

- Yasa Motor
- Rinehart Motor Controller
- Volt coolant pump (1 of two from vehicle)
- Low voltage harness
- Computer with CANoe installed
- CANcase
- Coolant lines (including reservoir)
- Opticool-H
- ABC 150
- HV wire (DC and 3-phase)

Low Voltage System

A low voltage test harness will need to be made to go to the MCU. It will have 12V power and ground (from a 12V battery). It will also have CAN which will run from a CANcase (connected to PC) to the MCU. A few low voltage connections will also need to be made from the motor to the MCU. The pump will be powered directly from a 12V battery and will be always on (at full power) while testing.

High Voltage System

There will be DC cables from the ABC 150 machine to the DC terminals of the MCU. Three phase cables will run from the three phase terminals of the MCU to the motor.

Coolant System

There will be a single coolant loop (with a reservoir) that runs through both the MCU and the motor. The coolant used will be Opticool-H. The loop will also have one pump, which will be always on.

Testing Procedures

MCU Repair Verification

1. Connect MCU to 12V power
2. Use Rinehart GUI to ensure proper settings are enabled on motor
    a. Extended CAN messages

   b. Motor parameters correct
3. Disconnect and reconnect MCU to 12V power
4. Turn on pump by connecting to battery
5. Turn on ABC 150, slowly increase voltage to 292V
6. Enable MCU via CAN (CAPL script)
7. Ensure MCU is enabled (via can)
8. Command Torque request from motor
9. Ensure motor spins
10. Zero torque request
11. Disable inverter
12. Turn off ABC 150
13. Disconnect 12V from MCU and coolant pump

Voltage based shifting data collection

1. Enable system (step 3-7 of MCU repair verification)
2. Set to desired voltage
3. Start at 0 torque request
4. Increment slowly until you see ripples in current (and maybe motor side voltage)
5. Set torque request back to 0
6. Change to next desired voltage, move to step 3
7. Shutdown system (steps 10-13 of MCU repair verification)

Table 1: Voltages to test

Voltage (Volts)	Max motor speed (rpm)
325	6150
320	6050
300	5500
280	5100
260	5000
240	4500
225	4300

LV Pinouts

23 pin connector MCU

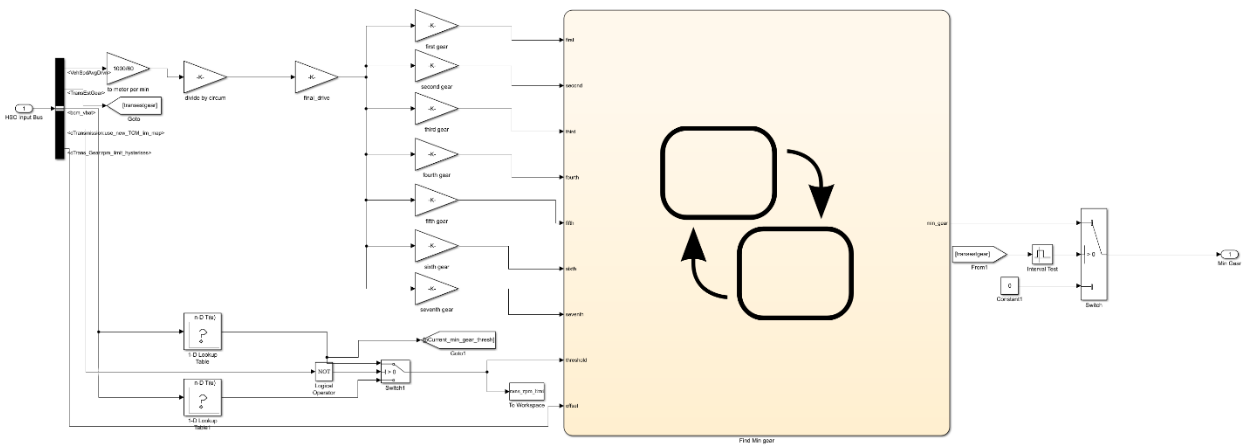
MCU pin	Description	Destination
1,4,11 (connect together)	5V power	YASA rotor pos sensor D
3	Resolver ground	YASA rotor pos sensor C
8	BATT+ (12V)	Battery positive
12	Resolver cos	YASA rotor pos sensor B
14	Ground	Battery neg
18	Resolver sin	YASA rotor pos sensor A
19	Resolver shield ground?	Resolver shield??

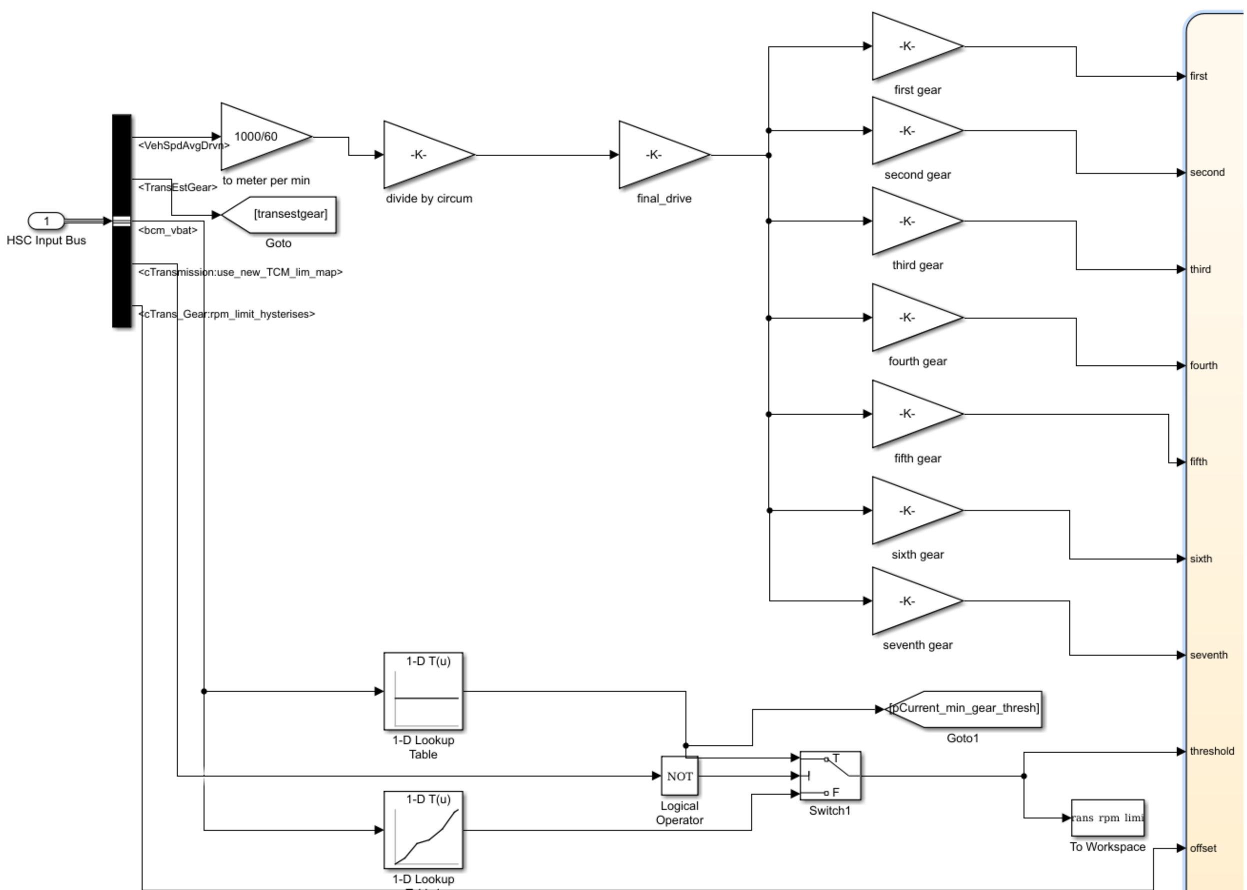
35 pin connector MCU

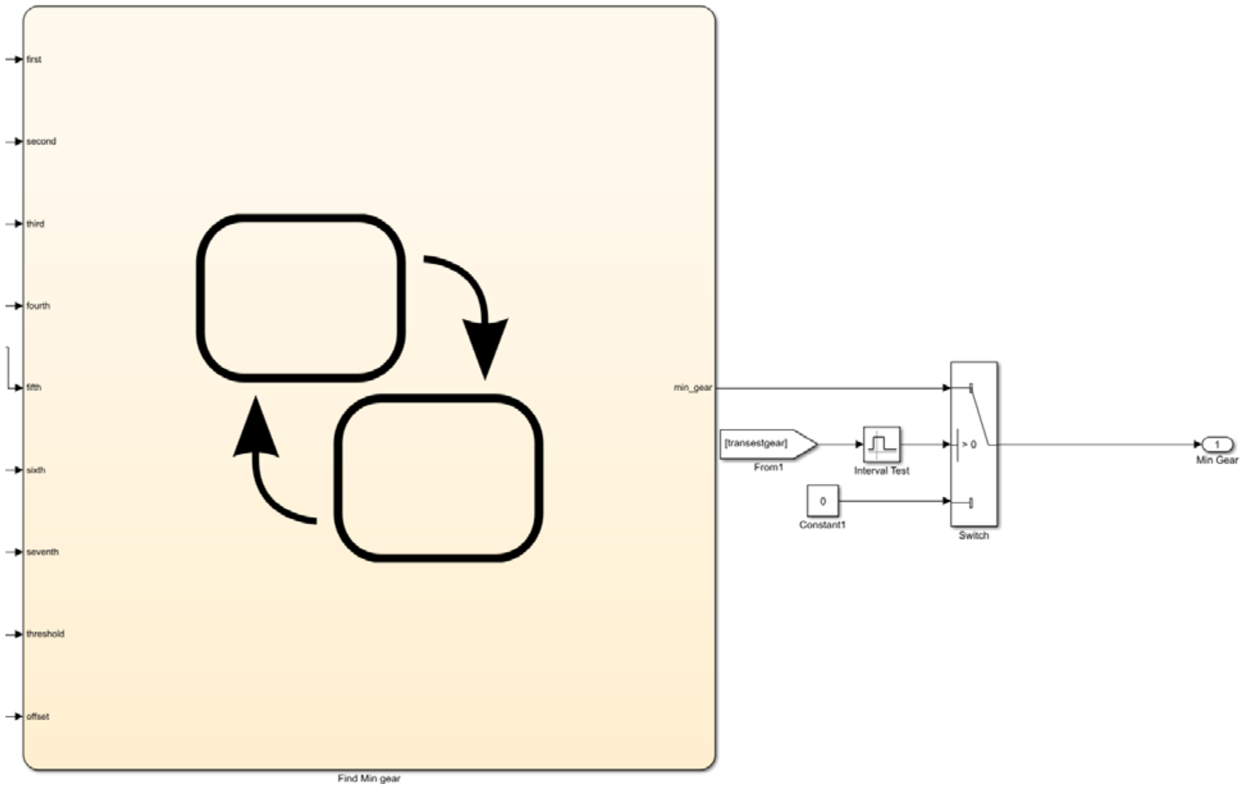
MCU pin	Description	Destination
3	Rotor temperature from motor	8F pin A
4	Stator core temp “motor temp param” on can	YASA stator data pin A (LV-007)
7	Program enable	RS 232 program enable (connect to ground when needed)
11	CAN low	CANcase
12	RS232 transmit	RS 232 connector
14	Rotor temp sensor power	8F pin C
15	Stator core temp	Stator data pin B (LV-007)
16	Stator winding temp	Stator data pin c
17	Stator winding temp	Stator data pin d
19	Rotor temp ground	8F pin b
22	RS232 ground	RS232 connector
33	CAN high	CANcase
35	RS232 receive	RS232 connector

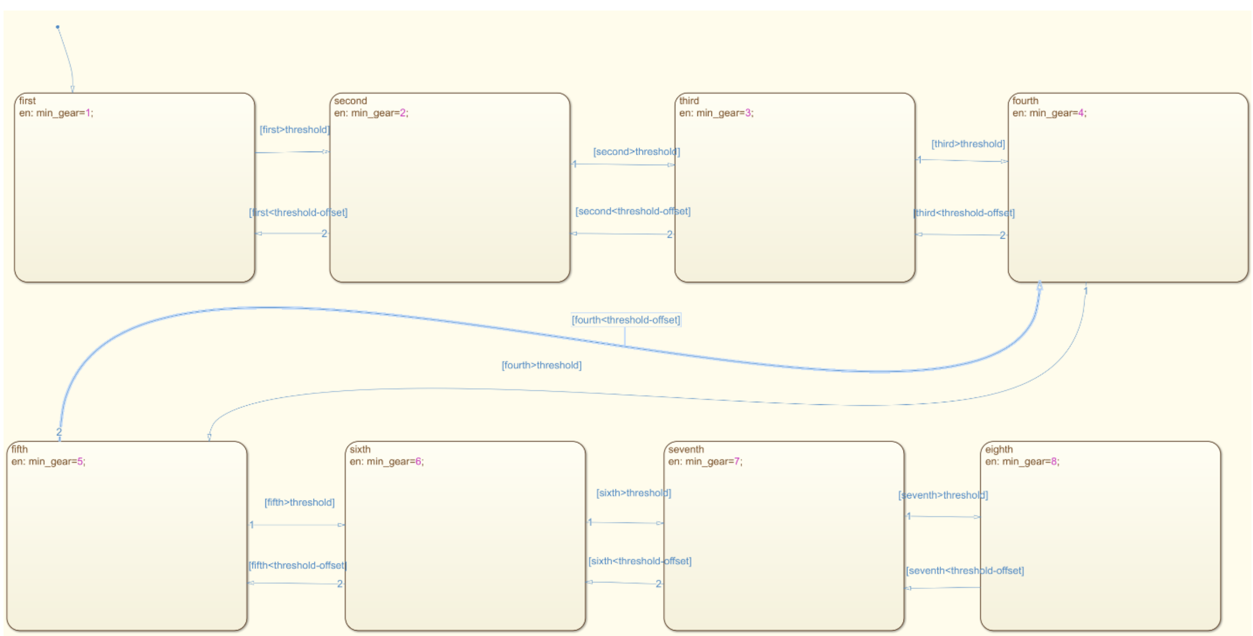
Appendix B

Voltage Based Shifting Algorithm Simulink Code






first
second
third
fourth
fifth
sixth
seventh
threshold
offset
min_gear
Find Min gear
[transestgear]
From1
Interval Test
> 0
0
Constant1
Switch
1
Min Gear


first
en: min_gear=1;
second
en: min_gear=2;
third
en: min_gear=3;
fourth
en: min_gear=4;
[first>threshold]
[first<threshold-offset]
[second>threshold]
[second<threshold-offset]
[third>threshold]
[third<threshold-offset]
[fourth<threshold-offset]
[fourth>threshold]
fifth
en: min_gear=5;
sixth
en: min_gear=6;
seventh
en: min_gear=7;
eighth
en: min_gear=8;
[fifth>threshold]
[fifth<threshold-offset]
[sixth>threshold]
[sixth<threshold-offset]
[seventh>threshold]
[seventh<threshold-offset]

BIBLIOGRAPHY

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

[2] The Pennsylvania State University's Advanced Vehicle Team, "EcoCAR3 Year Two Final Technical Report," The Pennsylvania State University, University Park, 2016.

[3] YASA Limited, "YASA P400 Product Sheet," August 2017. [Online]. Available: http://www.yasamotors.com/wp-content/uploads/2015/09/YASA_P400_Product_Sheet.pdf. [Accessed March 2018].

[4] Rinehart Motion Systems LLC, "PM100 Series AC Motor Drive User's Manual," 8 March 2011. [Online]. Available: https://www.neweagle.net/support/wiki/docs/Rinehart/PM100_User_Manual_3_2011.pdf. [Accessed March 2018].

[5] A. Emadi, D. Gao and C. Mi, "Modeling and Simulation of Electric and Hybrid Vehicles," *Proceedings of the IEEE,* vol. 95, no. 4, pp. 729-745, 2007.

ACADEMIC VITA

Academic Vita of Kyle H. Brown

Thesis

Exploring the Effects of a Voltage Based Shifting Algorithm on a Hybrid-Electric Vehicle Limited by Back EMF
Thesis Supervisor: Gary Neal

Education

The Pennsylvania State University, University Park, PA
Bachelor of Science, Electrical Engineering, Computer Science
Schreyer Honors College Graduation: May 2018
Dean's List: FA14, SP15, FA15, SP16, FA16, SP17, FA17

Technical Skills

- Knowledge and experience writing code in HTML, JavaScript, C, C++, C#, Java and Python
- Experience using MATLAB and Simulink for controls solutions
- Experience working with vehicle CAN buses and Vector CANoe/CAPL
- Experience designing and using SQL Databases and Microsoft Entity Framework
- Experience creating Windows and web applications
- Basic CAD, machining, wiring and soldering skills
- Experience working with hybrid/electrified vehicles and battery systems

Work Experience

Battery Test Intern, General Motors, Warren, MI **Summer 2016**
- Worked in GM's Global Battery Systems Lab running tests on production and pre-production battery systems

Software Intern, Nordco Rail Services, Beacon Falls, CT **Summer 2015**
- Developed web application using ASP.NET MVC
- Designed and implemented database using code first Entity Framework

Software Intern, Woodpyle Innovations, Southbury, CT **Summer 2014**
- Used Visual C++ and SQL for various projects involving Windows applications

Activities and Leadership

Penn State Advanced Vehicle Team (EcoCAR), Pennsylvania State University **2014- 2018**
- Veteran Member of Controls Team- Developed and tested hybrid vehicle control algorithm using Simulink and dSpace simulation tools
- Taught new members MATLAB/Simulink and our control strategies

Hack PSU, Pennsylvania State University **Spring 2015**
- Developed demo Windows 8 home automation application with a team of 4 in 24 hours
- Made use of voice recognition and Bluetooth communication with a microprocessor

FIRST Robotics, Watertown, CT **2010-2014**

- Team Captain, 2014; Electronics and software team captain, 2011- 2014
    - Wrote C++ code to control robot and programmed autonomous operation