

THESIS

PHYSICAL VALIDATION OF PREDICTIVE ACCELERATION CONTROL ON A PARALLEL HYBRID
ELECTRIC VEHICLE

Submitted by

Samantha M. White

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2022

Master's Committee:

Advisor: Thomas Bradley

Jason Quinn

Jeremy Daily

Bret Windom

Copyright by Samantha M. White 2022

All Rights Reserved

ABSTRACT

PHYSICAL VALIDATION OF PREDICTIVE ACCELERATION CONTROL ON A PARALLEL HYBRID ELECTRIC VEHICLE

Previous research has been conducted towards the development of predictive control strategies for Hybrid Electric Vehicles (HEVs). These methods have been shown to be effective in reducing fuel consumption in simulation, but no physical validation has been conducted. This is likely due to the fundamental "curses" of dynamic programming mostly the "curse of dimensionality" wherein the run-time needed to generate the optimal solution renders the method unfit as a real-time control. Predictive Acceleration Event (PAE) control combats the run-time issues associated with dynamic programming based control methods by pre-computing the optimal solutions for common Acceleration Events (AEs). This method was physically implemented on a 2019 Toyota Tacoma that was converted into a Parallel-3 (P3) HEV with limited information on the vehicle, including a reduced access to the vehicle's Controller Area Network (CAN) bus. Results from on-track testing indicate a Fuel Economy (FE) improvement in the range of 7% is possible to achieve using PAE control in the real world. To the author's knowledge this is the first time that this type of testing has ever been implemented on a vehicle in the real world.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES	v
LIST OF FIGURES	vi
1 Introduction	1
1.1 Project Objective	2
2 Vehicle Development	3
2.1 Hybrid Supervisory Controller Overview	3
2.1.1 I/O Signals	4
2.2 Toyota Gateway	6
3 Predictive Acceleration Event Model Development	7
3.1 Model Validation	8
4 Controller Development	10
4.1 Power Mode	11
4.2 Vehicle State	12
4.3 Pedal Logic	14
4.4 Torque Split	18
4.4.1 Baseline Torque Split Control	18
4.4.2 PAE Torque Split Control	19
4.5 ICE Control	20
4.6 Inverter and EM Control	22
4.6.1 CAN Input Signals	22
4.6.2 CAN Output Signals	23
4.6.3 Inverter States and Control Flow	23
4.7 HV Battery Control	25
4.8 State of Charge Calculations	25
4.8.1 BMS and Calculated SOC Comparison	26
5 Testing	27
5.1 On-Lift Component Testing	27
5.1.1 HV Battery Control	27
5.1.2 EM Control	28
5.1.3 APP Control	29
5.1.4 ICE Control	29
5.2 On-Lift Vehicle Testing	30
5.3 Parking Lot Testing	30
5.4 Airfield Testing	31
5.4.1 Results	34
6 Discussion	38
6.1 Implications of PAE on Hybrid Vehicle Control	38
6.2 Implications of Robust, Map-Based Predictive Powertrain Control	38
6.3 Implications for Vehicle Testing and Demonstration	39
7 Conclusion	40

References	41
Appendix A: PAE Figures	43
Appendix B: Baseline Figures	47

LIST OF TABLES

2.1	Toyota Gateway Signal Output List	6
4.1	Signals Received by the HSC from the Inverter	22
4.2	VSM State Signal Descriptions	22
4.3	Signals Transmitted from the HSC to the Inverter	23
5.1	Battery State List	28
5.2	Shift Schedule	34
5.3	Comparison of FE Change for PAE and Baseline	36

LIST OF FIGURES

2.1	Frequency versus Voltage for APP1 and APP2	5
2.2	Simple APP Signal Flowchart	5
3.1	BSFC Map Developed From Collected Data	8
3.2	PAE Model Comparison to On-Vehicle Testing	9
4.1	Simplified Controller Flowchart	11
4.2	Power Mode Flowchart	12
4.3	Simplified Vehicle State Machine Flowchart	13
4.4	Simplified Pedal Logic Flowchart with Unit Conversions	14
4.5	APP Conversion Charts	15
4.6	State Machine used to Convert APP to a Torque	16
4.7	Comparison of HSC APP Output to the Actual, Vehicle Reported APP Value	17
4.8	Resulting MSE of the Data With a Timeshift	17
4.9	Baseline Torque Split	18
4.10	Optimal Control Matrix Torque Computation Flowchart	19
4.11	Pre-Computed Torque Trace Flowchart	20
4.12	Comparison of the Distribution of Filtered and Unfiltered Engine Torque Data	21
4.13	Plot of Requested Torque and Actual Torque with Variation Lines	21
4.14	Inverter State Machine	24
4.15	HV Battery State Machine	25
4.16	Comparison of Calculated SOC versus BMS Reported SOC	26
5.1	HV Battery Voltage for the EM Torque Output as Reported by the Inverter and BMS	28
5.2	Optimal Matrix Example Plots	32
5.3	Vehicle Speed Trace	33
5.4	Dash Switches	34
5.5	BSFC Plots for PAE AEs	35
5.6	BSFC Plots for Baseline AEs	35
5.7	ICE Fault Causing ICE Torque Output To Equal Zero While APP Commands Were Nonzero	36
A1.1	Vehicle Velocity Comparison for Three PAEs	44
A1.2	ICE Output Comparison for Three PAEs	44
A1.3	EM Output Comparison for Three PAEs	45
A1.4	HV Battery SOC Comparison for Three PAEs Events with the SOC Reset to 50% After Each AE	45
A1.5	Transmission Gear Value Comparison for Three PAEs	46
B1.1	Vehicle Velocity Comparison for Three Baseline AEs	47
B1.2	ICE Output Comparison for Three Baseline AEs	47
B1.3	EM Output Comparison for Three Baseline AEs	48
B1.4	HV Battery SOC Comparison for Three Baseline AEs with the SOC Reset to 50% After Each AE	48
B1.5	Transmission Gear Value Comparison for Three Baseline AEs	49

Chapter 1

Introduction

As people have become more mobile, the amount of emissions from the transportation sector has increased [1]. Currently, over the course of one year, highway vehicles produce 1.6 billion tons of Greenhouse Gases (GHGs), mostly in the form of Carbon Dioxide (CO₂) [2, 3]. According to the Environmental Protection Agency (EPA), the transportation sector generated the largest share of GHGs in 2019, at 27% [4].

As stated in studies by Dong et al. and Peters et al., GHGs should be reduced since they can be harmful to human health [5, 6]. Another researcher, Crowley, has shown the impact of GHGs on the environment and its relationship with climate change. Other researchers have discussed the further impact of climate change on human health through extreme weather events, disease, and reduced access to food and water [7]. Outside of human health, consumer preference and support has changed to support a reduction in GHGs [8].

Vehicle-produced GHGs can be reduced using a variety of methods. These methods tend to fall under two categories: reducing the number of miles driven and increasing the Fuel Economy (FE) of vehicles. Reducing the miles driven on private vehicles can be accomplished through increasing the accessibility of public transportation [9]. However, this can be difficult to test as public transportation requires extensive planning and a large amount of money to implement [10].

Increasing the FE of vehicles can be accomplished through vehicle electrification, and the use of Hybrid Electric Vehicles (HEVs) to increase fuel economy over vehicles that only contain Internal Combustion Engines (ICEs) [11]. HEVs can have a better FE than conventional vehicles since they use both an ICEs and an Electric Motors (EMs) to produce torque that propel the vehicle. This allows for a torque control strategy to be developed to improve the FE by splitting the driver's requested torque between the ICE and EM [12]. Doing this can reduce the fuel required by the ICE while still meeting the driver's requested torque.

Currently, the HEVs on the road use instantaneously optimized torque control strategies to determine the torque output of the ICE and EM on the vehicle [13]. This method attempts to minimize the fuel consumption of the vehicle for the instant the calculation occurs [14]. This paper will focus on a predictive powertrain control method, which pre-computes the optimal control output for the given acceleration event, to produce optimal results.

1.1 Project Objective

The overall objective of this project was to test an optimal HEV torque splitting method against a baseline torque splitting method in a real world vehicle to observe the improvement in FE. This optimal torque split method, called Predictive Acceleration Event (PAE), used a Dynamic Programming (DP) algorithm to pre-compute the torque outputs of the ICE and EM for a specific acceleration event [15, 16]. It then creates a map of the pre-computed control information to use as a lookup table within the vehicle to determine the optimal commands for the specific points of the Acceleration Event (AE) that the vehicle drives through.

The benefits of PAE and other optimal approaches to HEV control have been theorized through simulations in previous literature [17, 18, 19, 20, 21, 22]. However, this project was the first time it had been tested in a real-world application to observe and demonstrate measurable benefits in FE. The powertrain of this vehicle was converted from a standard ICE based powertrain to a Parallel-3 (P3) Plug-In Hybrid Electric Vehicle (PHEV) based powertrain with minimal information given on the vehicle from the manufacturer. A control system also had to be developed to control both the ICE and EM as well as the additional components required for the vehicle to function as a PHEV.

Chapter 2

Vehicle Development

For this project, a P3 HEV was developed from a stock 2019 Toyota Tacoma [23, 24]. In a P3 HEV, the EM is located in between the transmission and the differential. This vehicle configuration was chosen for a number of reasons. The first was the relative ease of manufacturing in comparison to the other types of hybrid configurations. This type of powertrain was also comparatively easier to implement a supervisory controller in. The P3 configuration allowed the main structure of the vehicle to remain relatively unchanged.

This modification added an electric motor between the transmission and the differential as well as the necessary components to support the electrified powertrain. This included the Inverter, Battery, Battery Management System (BMS), Hybrid Supervisory Controller (HSC), Toyota Gateway, On-Board Charger (OBC), and the needed 12 volt powered components to control and provide thermal regulation of the components.

2.1 Hybrid Supervisory Controller Overview

This control system functioned as a supervisory control system to combine the ICE part of the powertrain with the EM part of the powertrain. A 112 pin Woodward Motohawk (ECM-5644-112 SECM-112) was used as the hardware for the HSC. Matlab's Simulink software was the development environment for this program allowing for the use of Woodward's Motohawk rapid controller development software. This software was used to build the HSC code that was compatible with the Motohawk hardware. It also allowed for values within the controller to be viewed and calibrated in real time, on the vehicle.

This controller hardware consisted of input and output ports, which could receive and send analog, digital, and Controller Area Network (CAN) signals to the other parts of the vehicle. It also included a microcontroller that housed the Random-Access Memory (RAM), Read-Only Memory (ROM), and Central Processing Unit (CPU) [25].

The HSC operated as an Input-Processing-Output (IPO) model where it converted input signals to output signals to control the vehicle's behavior. These Input/Output (I/O) signals were received or transmitted in one of three forms: CAN, digital, or analog.

2.1.1 I/O Signals

CAN Signals Controlling the components in the electric powertrain involved communication through the two custom CAN buses that were added to the vehicle: a CSU CAN network and a battery CAN network. DBC files were developed to transmit and identify information from the CAN dataframe. The HSC received CAN signals from the Toyota Gateway, BMS, and Inverter. It transmitted signals to the BMS and the Inverter.

Digital Signals The oil pump, radiator fan, and Anti-Freeze (AF) pump were controlled by boolean output signals. The relays for the BMS, OBC, and Inverter were also controlled by boolean output signals. The digital pins allowed the listed components to be turned on and off as determined by the HSC logic.

Two switches were added to the vehicle's dashboard as a way to send commands to the HSC without using a laptop. These switches were connected to physical input pins. Also, the wire that provided power to the radio was spliced and routed into the HSC to be used to monitor the vehicle's power state.

Accelerator Pedal Position (APP) Signals To control the ICE in this vehicle, the APP signal was intercepted on its path to the Engine Control Module (ECM) and rerouted into the HSC analog input ports. This signal was then modified and output to the ECM from an analog port. The APP signals were different from the other I/O signals since these signals used both analog and digital pins.

The APP was input as an analog value since it consisted of two Accelerator Pedal Position Sensor (APPS) signal wires, called Accelerator Pedal Position 1 (APP1) and Accelerator Pedal Position 2 (APP2), that transmitted a voltage value that was proportional to an APP percentage.

The HSC could not, however, output varying analogue voltages, so signal conditioners were used to convert the digital, Pulse-Width Modulation (PWM), signals to the necessary voltages. To control the voltage, the PWM frequency values were manipulated while the duty cycle remained constant. Figure 2.1 shows the frequency and voltage conversion values as well as the minimum and maximum values for APP1 and APP2.

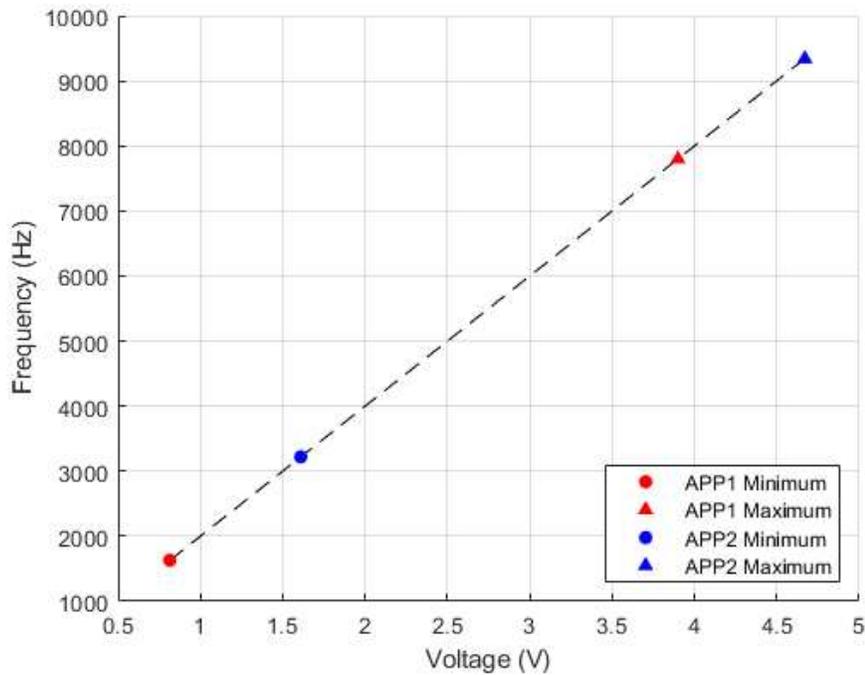


Figure 2.1: Frequency versus Voltage for APP1 and APP2

As the previous figure shows, the frequency to voltage conversions for APP1 and APP2 were almost the same. The only difference between the two were the maximum and minimum voltage values that the signals reached. Figure 2.2 shows a simplified flowchart of the APP signal through the HSC and to the ECM.

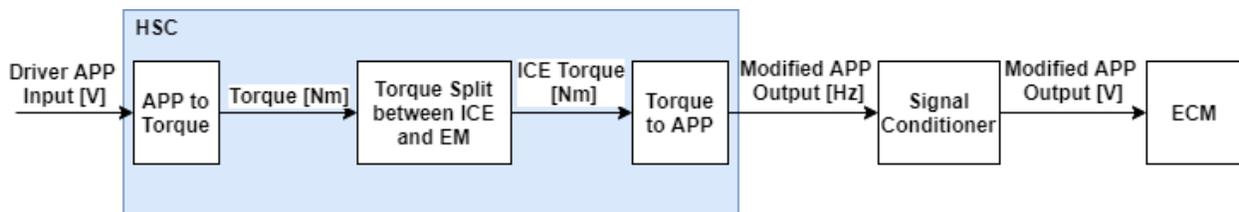


Figure 2.2: Simple APP Signal Flowchart

From this figure, it can be seen that the APP value input was converted to a torque within the HSC. This torque was then split between the ICE and EM. After this split, the torque was converted back to an APP to be sent to the ECM via the signal conditioner. From there, the controllers within the vehicle converted the APP signal to a torque output.

2.2 Toyota Gateway

Minimal information was provided from the manufacturer about the vehicle to incorporate into the controller. A Toyota Gateway was provided with the purpose of forwarding specific signals from the vehicle's CAN bus to the HSC. These signals included those shown in Table 2.1.

Table 2.1: Toyota Gateway Signal Output List

Signal Name	Description
ShiftPosition	Gives the gear value of the vehicle when in Sport Mode [0, 1, 2, 3, 4, 5]
BrakeSignal	Boolean value signaling whether or not the brake pedal is pressed [0, 1]
EngRPM	Engine speed value [RPM]
TqReqNoBrk	Torque value [Nm]
EngTq	Engine torque value [Nm]
AccPedPct	Accelerator pedal percent value [%]
VehSpd	Vehicle speed [km/h]
FuelConsumption	Fuel consumption value [1/(L/100km)]
TqReqBrk	Torque value [Nm]

At Toyota's request, no other signals from the base vehicle were used within the HSC. The HSC did not use information from the other Toyota CAN buses.

Chapter 3

Predictive Acceleration Event Model Development

The overall plan of the PAE Torque Split Strategy was to predict the optimal torque split between the ICE and EM for a defined AE. In this scenario, the optimal torque split was one that produced the best FE. It balanced commanding the ICE torque value that consumed a low amount of fuel, via the Brake Specific Fuel Consumption (BSFC), with the EM torque output that would allow for the start value of the High Voltage (HV) battery's State of Charge (SOC) to equal the end value.

It required a trace of the AE to be processed through a DP algorithm that used a model of the vehicle to calculate this optimal strategy. The DP algorithm worked backwards through the trace to find the optimal torque split strategy for each time step in the AE to keep the engine operating in as fuel efficient of a state as possible while also ensuring that the HV battery's SOC ends at the same value as it began at. This strategy could then be implemented within the HSC.

Since the PAE torque split strategy required the use of a model to predict the optimal torque output matrix [15], it was necessary to develop a model of the modified vehicle before the PAE torque split method could be included in the HSC. This model was based on the model developed in [26]. It included simple, physics-based, models of the ICE, EM, battery, and transmission. Constraints were added into the model to keep the SOC, EM, and ICE within their allowable boundaries. It was important to develop a vehicle model that approximated the behavior of the actual vehicle while still keeping the model simple enough to allow for the necessary data to be able to be collected through real world testing.

Vehicle velocity traces were input into the developed model to obtain the necessary outputs to add into the HSC code: the optimal control matrix, ICE torque trace, and EM torque trace. The optimal control matrix contained the data to output the on-vehicle torque split values based off of the vehicle speed, transmission gear value, and battery SOC. The ICE and EM torque traces were used to directly control the vehicle, bypassing the on-vehicle torque split. This model could also simulate vehicle behavior like the vehicle's speed, SOC, EM torque, ICE torque, EM power, and EM speed. These values could be used to compare to the real world.

It required real-world vehicle data to be collected and converted into maps that could be used in the calculations to generate the output data. The BSFC map was one of the necessary maps for this model. Data collected by driving the vehicle around Fort Collins, CO was used to generate a BSFC map to model engine efficiencies at various engine speed and torque combinations. Figure 3.1 shows the BSFC map derived from data points gathered during vehicle testing. To increase the ICE's efficiency, the vehicle should operate in the dark blue regions of the map.

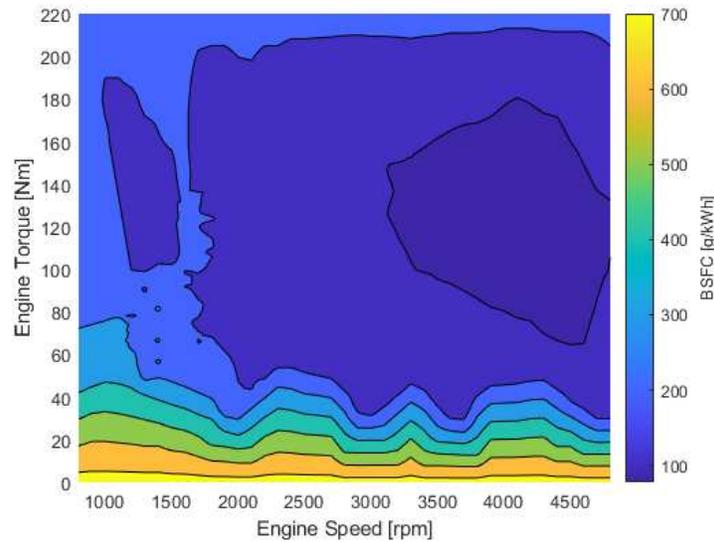


Figure 3.1: BSFC Map Developed From Collected Data

The figure shows the most efficient BSFC values occurred at high engine speeds and medium levels of torque output. Low ICE torques resulted in the least efficient BSFC values. This information could then be used to generate torque values for the PAE torque control method.

3.1 Model Validation

The PAE model was validated against data obtained through testing the vehicle. To obtain the data to compare the PAE model against the vehicle performance, the vehicle was physically driven through a 0 to 70 kph AE. Then, the vehicle speed trace was input into the PAE model, and the model behavior was compared to the vehicle's behavior. Figure 3.2 shows comparison of on-vehicle data to model data through the following values: vehicle speed (3.2a), SOC (3.2b), ICE Torque (3.2c), EM Torque (3.2d), EM Power (3.2e), and EM Speed (3.2f).

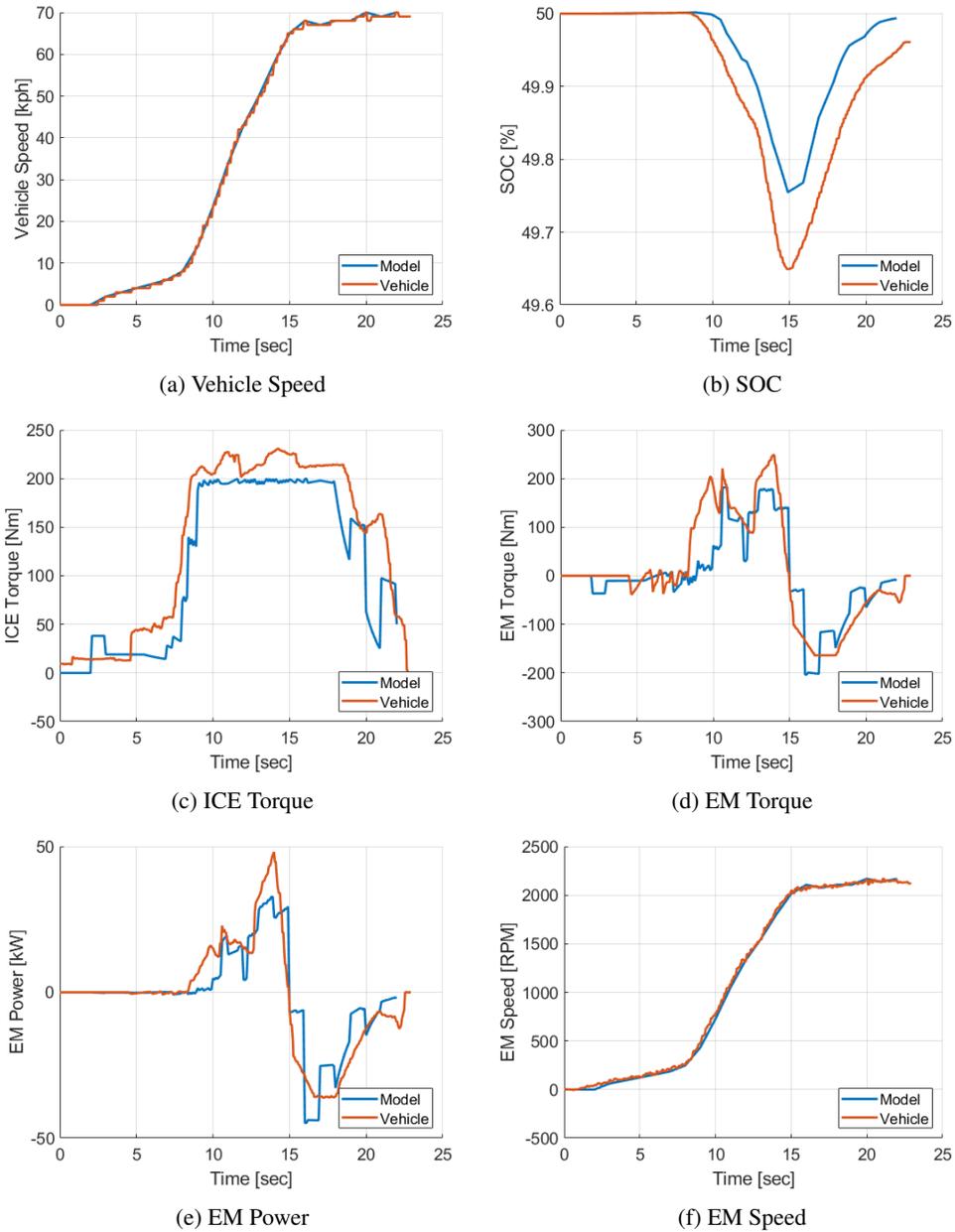


Figure 3.2: PAE Model Comparison to On-Vehicle Testing

These figures show that the torque output values from the model followed the same general trends as the actual values that the vehicle output during the AE. This illustrates the ability of the simple model combined with real-world test data to provide vehicle behavior that was similar to reality.

Chapter 4

Controller Development

When developing the controller, the following constraints were observed:

- The HSC's storage capacity was too small to contain a full optimal control matrix.
- The signals from the Toyota Gateway were the only values that could be used from the base vehicle.
- The BMS was an unreliable source of information, especially with producing SOC values.
- The driver could not consistently reproduce exact AEs under manual control using the accelerator pedal input.
- The HSC had no signal to differentiate between the vehicle's accessory mode and a fully on mode.
- The engine could not be controlled directly by torque requests.
- The Brake Pedal Position Sensor (BPPS) did not output non-binary values.

Figure 4.1 shows a simplified structure of the HSC and the flow of signals through the controller. It shows the signals that occurred when the vehicle was turned on and off as well as the signals that occurred continuously while the HSC was on.

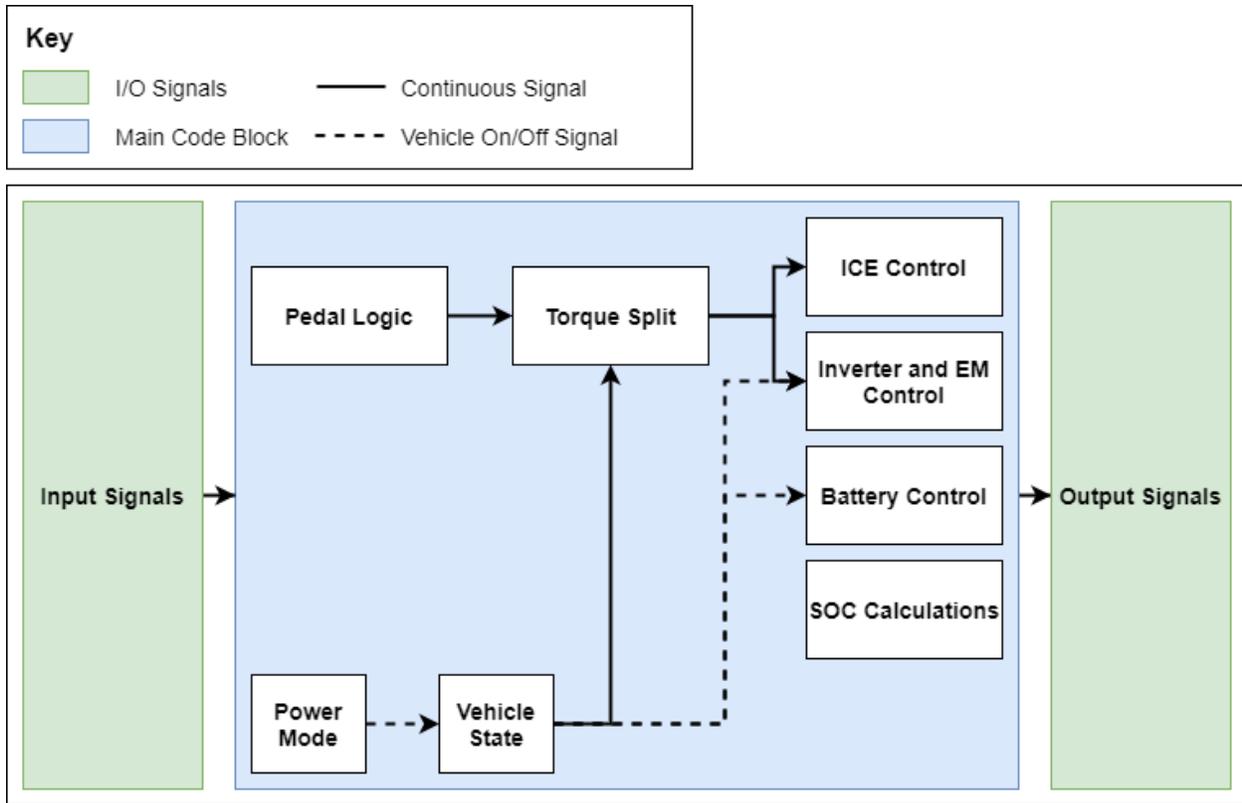


Figure 4.1: Simplified Controller Flowchart

This flowchart illustrates the basic outline of the HSC code. It begins with the input signals and ends with the output signals. The flowchart shows the connection between power mode, vehicle state, pedal logic, torque split, ICE control, inverter and EM control, battery control, and SOC calculations.

4.1 Power Mode

The power mode section of the controller was used to differentiate between the vehicle's Off mode, Accessory mode, and On mode. To do this, a state machine was used with radio power and engine rpm as the two inputs. The radio power was used since it was on in both the Accessory and On states and off when the vehicle was off. Engine rpm was used since it was only greater than zero when the vehicle was on. The output of this state machine is the power mode. Figure 4.2 shows the simplified power mode flowchart.

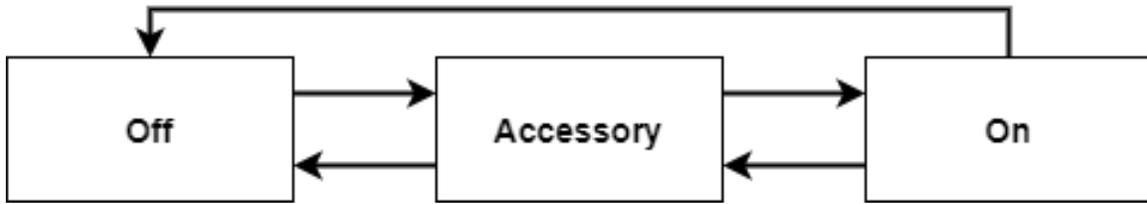


Figure 4.2: Power Mode Flowchart

As seen in the figure, this flowchart had three states: Off, Accessory, and On. These states were modeled after the fundamental power states of the vehicle.

Off State While in the Off state, the vehicle was completely off. In this state, the radio must be off and the engine rpm must be zero. When the radio was turned on, the controller moved into the Accessory state.

Accessory State In the Accessory state, the radio signal had to be greater than zero and the engine rpm had to be zero. If the radio signal returned to zero, the controller moved back to the Off state. If the radio signal remained greater than zero, and the engine rpm was also greater than zero, the controller moved to the On state.

On State In the On state, the radio must be on, and the engine rpm must be greater than zero. When leaving this state, the state machine allowed the controller to go back to the Accessory state or into the Off state. To go to the Accessory state, the engine rpm would need to drop to zero while the radio signal remained greater than zero. To go to the Off state, the radio signal must drop to zero.

4.2 Vehicle State

The Vehicle State section of the code consisted of a large state machine that was used to define the overall vehicle process state. It provided the commands to wake components, close the battery contactors, generate torque split values, and shut down components. Figure 4.3 shows a simplified overview of this state machine.

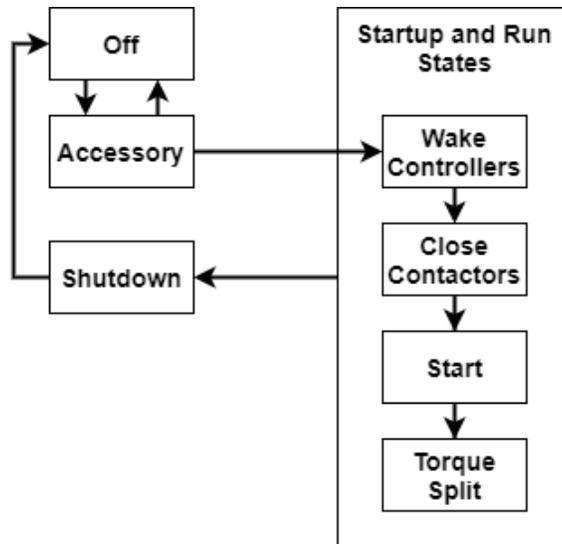


Figure 4.3: Simplified Vehicle State Machine Flowchart

Off and Accessory States The controller started in the Off state and moved into the Accessory state when the power mode state machine from Section 4.1 moved to the Accessory state. It also moved back to the Off state with the state machine from that section. When the power mode state machine moved into the On state, this state machine moved into the Wake Controllers state within the Startup and Run state group.

Startup and Run State Group This state grouped together four states that related to starting the vehicle up and running the vehicle. During this group of states, the vehicle’s ICE has turned on as the vehicle was keyed on. This group handled turning the components on that were related to the electric powertrain. At any point during the Startup and Run states, the state could change to the Shutdown state if the Power Mode state changed. While in this state, the controllers that were woken would be shut down and the contactors would be opened.

Wake Controllers State In the Wake Controllers state, the state machine sent commands to the inverter and BMS to wake them. After they were online and ready for more commands, the controller moved to the Close Contactors state.

Close Contactors State While in the Close Contactors state, commands were output from the state machine signaling the need to close contactors on the battery. Once they were closed, the controller moved to the next state, the Start state.

Start State This state was added to the controller to allow it to pause to ensure the previous commands were successful and that no faults occurred before continuing onto the Torque Split state. The Start state was necessary to

account for BMS faults that would occur immediately after the contactors closed.

Torque Split State The Torque Split state was the state that the controller spent the most time in. It contained the necessary code to split the driver torque input into an ICE torque and an EM torque, with the different torque split methods. To leave this state, the power mode from Section 4.1 would need to change from the On state to the Accessory or Off state. Leaving this state put the controller into the Shutdown state.

Shutdown State In the Shutdown state, commands were output within the HSC to open the battery’s contactors and put the inverter and BMS to sleep. The torque output from this state was zero. After this was completed, the controller moved back into the Off state to start the cycle again.

4.3 Pedal Logic

In Section 2.1.1, a brief overview of the APP signal flow was discussed that placed an emphasis on the APP through the scope of the HSC I/O. This section will take a more in depth look at what happened to the APP with respect to the flow internal to the HSC. As discussed previously, the APP signals were important to determine the driver’s intended torque and to manipulate the torque output of the ICE in accordance with the torque split strategy in place within the HSC.

Figure 4.4 shows the pedal logic flow through the HSC. It included the different units that the APP signals were converted to within the controller since the APP signals went through the most unit conversions of any signal in order to get these signals in usable states for the HSC code and a readable state for the ECM.

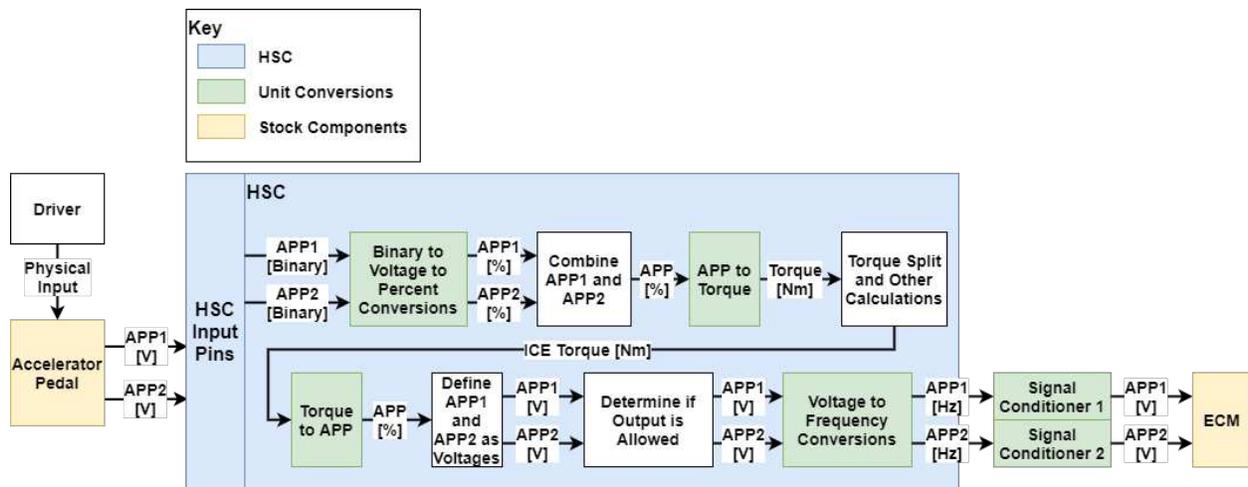


Figure 4.4: Simplified Pedal Logic Flowchart with Unit Conversions

The flowchart begins with the driver’s physical input to the accelerator pedal, to change the APP value. This then

resulted in a change in voltage along the APP1 and APP2 wire inputs to the HSC input pins. The HSC read these values as a binary value, whose conversion to a voltage can be seen in Figure 4.5a with the voltage to percent conversion in Figure 4.5b.

To obtain these conversion values, varying voltage values were applied to the HSC's APP1 and APP2 input pins. The binary values were recorded, plotted, and can be found in Figure 4.5a. Then, the voltage values for APP1 and APP2 were obtained by moving the accelerator pedal between 0% and 100% and recording the voltage value and the APP value reported by the Toyota Gateway (Figure 4.5b).

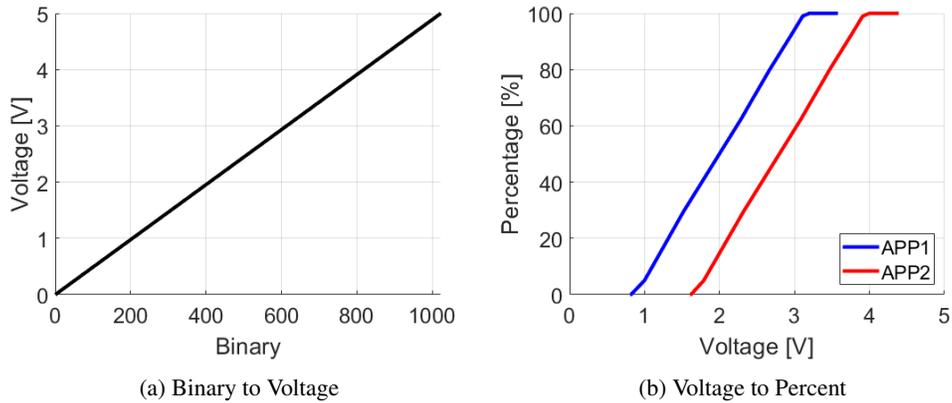


Figure 4.5: APP Conversion Charts

As Figure 4.5 shows, APP1 and APP2 had slightly different conversion values for the voltage to percent but used the same conversion for the binary to voltage. After the unit conversions needed to bring the APP percent values into the HSC, APP1 and APP2 needed to be combined before the next conversion could take place. This was done by taking the average of the two values. This average, called APP in the figure, was then converted to torque using the state machine shown in Figure 4.6.

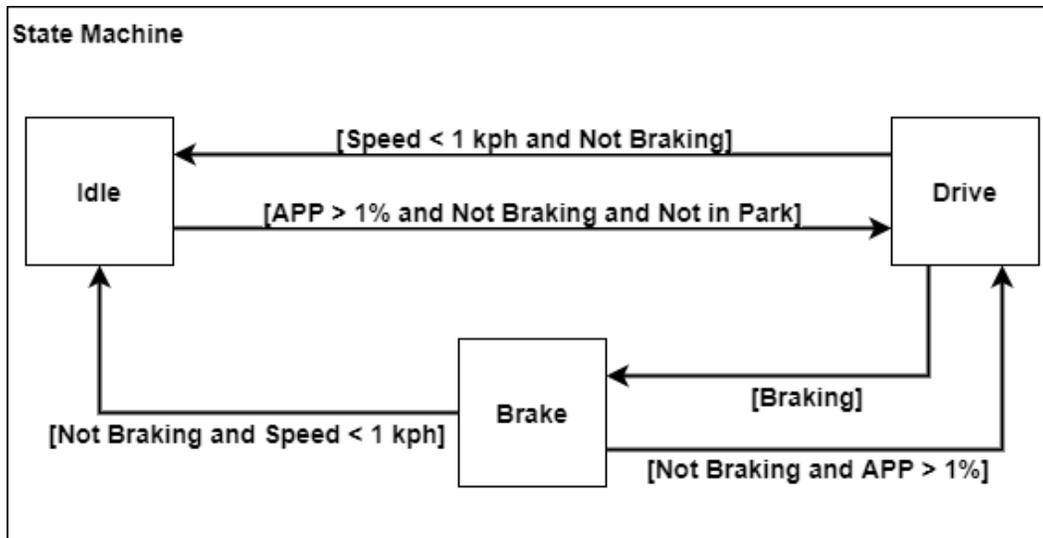


Figure 4.6: State Machine used to Convert APP to a Torque

This state machine was used to differentiate between the three main driving states that the vehicle went through: Idle, Brake, Drive. In Idle and Brake, the axle torque command was always zero. Though these could be combined into one state, since they had the same output, they were split up for clarity and to allow for the possibility of regenerative braking. However, this functionality was not added into the vehicle due to the lack of a non-binary BPPS value provided by the Toyota gateway device. In the Drive state, vehicle speed and pedal position were input into a pedal map to generate a torque value that fed into the rest of the controller.

This torque was then split between the ICE and EM using torque splitting methods that will be discussed in Torque Split Section 4.4. The ICE torque was then converted to an APP value. This conversion will be discussed more in ICE Control Section 4.5.

The ICE APP value was then split back into APP1 and APP2 voltages using the conversion from Figure 4.5b. The voltages were then converted to a frequency, using PWM pins, and output to the signal conditioners to be converted back to voltages for the ECM to read. Figure 4.7 shows a comparison of the HSC APP output value with the value that the vehicle reported through the Toyota gateway.

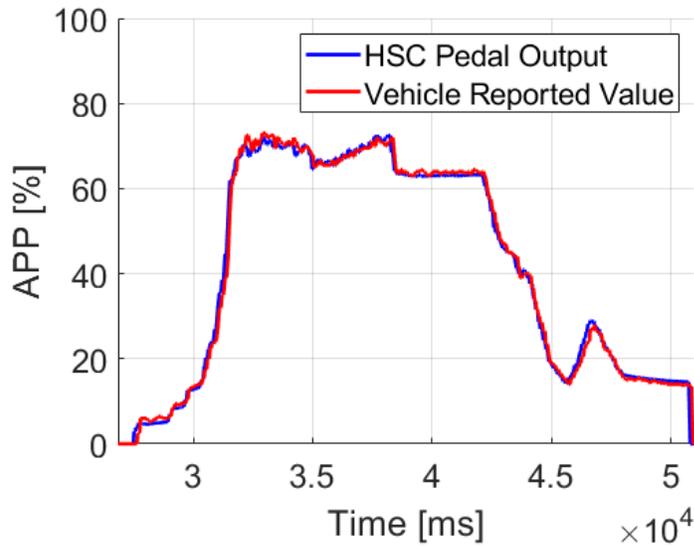


Figure 4.7: Comparison of HSC APP Output to the Actual, Vehicle Reported APP Value

As seen in Figure 4.7, there was a difference between the commanded APP value and the value reported by the vehicle, in terms of both the offset in time and to APP. To quantify the difference between these two values, the scaled Mean Squared Error (MSE) values were calculated for this and 17 other datasets, totalling 38,523 datapoints. The results of this can be found in Figure 4.8, where Error Percentage quantifies the difference in the APP values and the Time Shift quantifies the difference in time values.

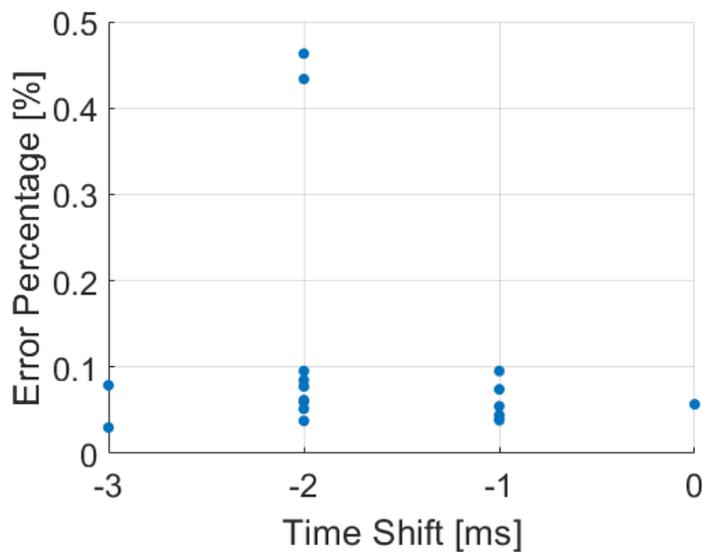


Figure 4.8: Resulting MSE of the Data With a Timeshift

From this plot, it can be seen that APP values output from the HSC and APP values reported by the Toyota gateway

were very similar. None of the 17 datasets used for this calculation had a higher error percentage than 0.5%. This means that the overall difference between what the HSC commanded was less than 0.5% different than what the Toyota gateway reported. Also, the Time Shift axis suggests that there was less than 3 milliseconds between the HSC commanding an APP value and the Toyota gateway reporting the APP value.

4.4 Torque Split

Two types of torque split calculations were used within the HSC calculations: Baseline Torque Split Control and PAE Torque Split Control. These two methods were chosen to compare against each other to determine the ability of the optimal torque split method to improve the vehicle's FE.

4.4.1 Baseline Torque Split Control

A load following torque split strategy was chosen to act as the Baseline Torque Split Method. The load following method calculated the ICE torque first, compared that torque value to the driver's requested torque, and filled in the difference with EM torque. This method worked in all driving scenarios, and was calculated on the vehicle. No calculations needed to happen before the vehicle was driven. However, this strategy did not provide an optimal torque split for the AE. The flowchart for the Baseline strategy can be found in Figure 4.9.

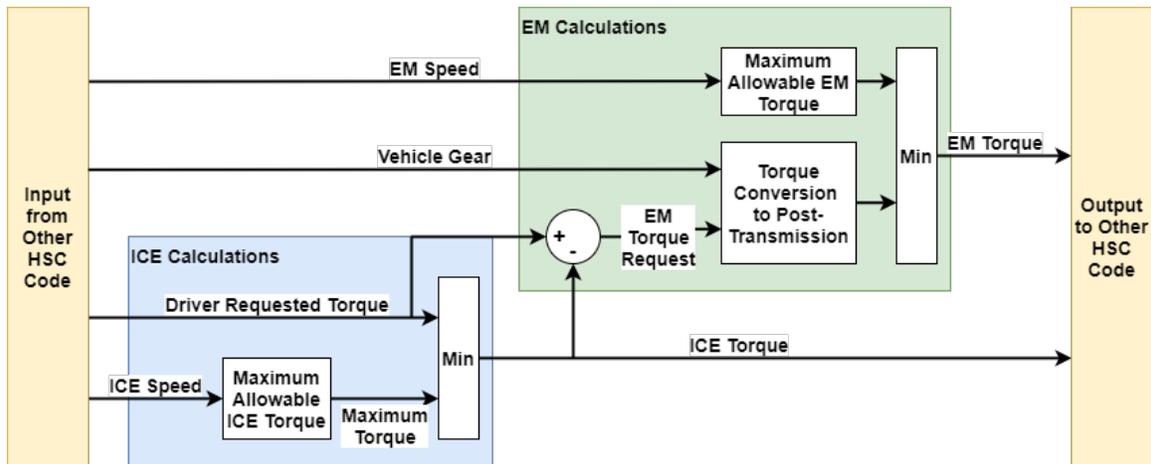


Figure 4.9: Baseline Torque Split

This strategy was composed of two main components: ICE calculations and EM calculations. The ICE calculations started with the minimum of the driver requested torque and maximum allowable torque for the given engine speed being selected for the ICE torque output. The ICE torque output was then subtracted from the driver requested torque to obtain the EM torque request value.

Since this was a P3 HEV, and the ICE and EM were located on opposite sides of the transmission, the EM torque request value must be multiplied by the vehicle gear ratio to obtain the post-transmission EM torque request value. The post-transmission value and maximum allowable EM torque are then compared, with the smaller value selected to be the EM torque output.

4.4.2 PAE Torque Split Control

Two PAE methods were used to calculate torque split values: on-vehicle torque computation through an optimal torque split matrix and pre-computed torque traces. The optimal torque split matrix, used for the on-vehicle torque split computations, was obtained from the PAE model, as discussed in Section 3. The pre-computed torque traces were also obtained as an output from this model.

Optimal Control Matrix Torque Computation This torque split method required the AEs to be pre-computed to generate an optimal control matrix of torque split output values. This optimal control matrix was then added to the controller, so that the engine torque could be determined from the SOC and vehicle speed while performing an AE. Figure 4.10 shows the simplified flowchart of this method.

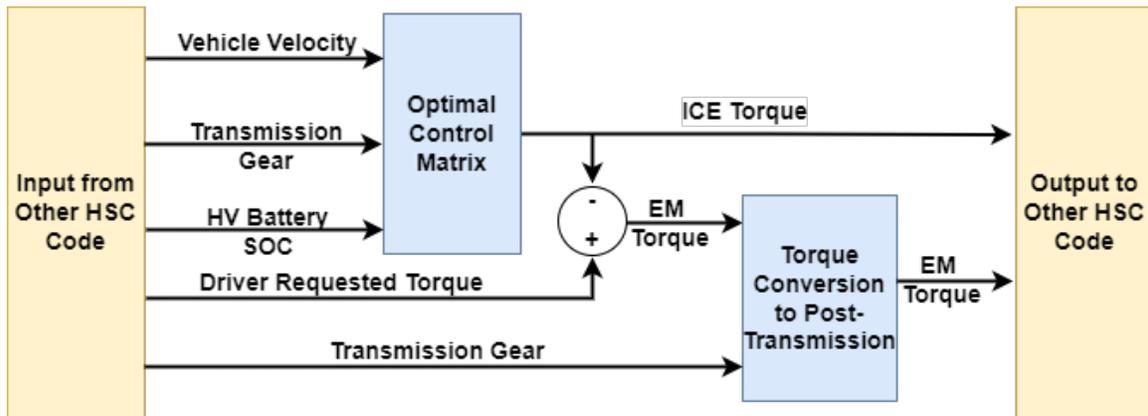


Figure 4.10: Optimal Control Matrix Torque Computation Flowchart

In this figure, it can be seen that the optimal control matrix requires three inputs: vehicle velocity, transmission gear, and the SOC of the HV battery. This matrix then output the ICE torque which, when subtracted from the driver requested torque, produces the pre-transmission EM torque value. This value then was multiplied by the transmission gear to convert to a post-transmission EM torque value.

Pre-Computed Torque Trace The Pre-Computed Torque Trace involved generating the ICE and EM PAE torque traces in advance and using the traces to control the torque instead of the driver's APP input. Figure 4.11 shows the flowchart for this PAE method.

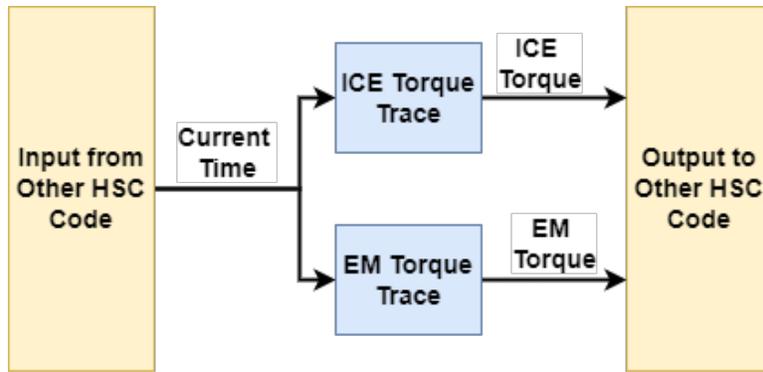


Figure 4.11: Pre-Computed Torque Trace Flowchart

As seen in the figure, the current time was input into lookup tables that contained ICE and EM torque traces which converted the time value to a torque output. After each AE, the driver reset the current time value back to zero via a manual switch, located on the vehicle’s dashboard. A second dash switch was used to trigger the block that contains the torque trace logic.

4.5 ICE Control

The calculations within the controller system were completed as torques. However, controlling the engine required the torque to be converted an APP. To convert the engine torque value to an accelerator pedal position, a three input regression equation was created using data collected by driving the vehicle around Fort Collins, Colorado. The three inputs used to create the engine model included vehicle speed, engine speed, and engine torque.

Both vehicle speed and engine speed were used to account for gear, since the provided gateway signals did not include a reliable gear value. The gateway only reported the value shown to the driver when the vehicle was in sport mode.

Over 200,000 datapoints were initially used to develop this model of the engine. This data was reduced by filtering to improve the uniformity of the engine torque data. The data was filtered to only include the data where the vehicle speed and engine torque values were greater than zero. As can be seen in Figure 4.12, doing this greatly reduced the span of the data.

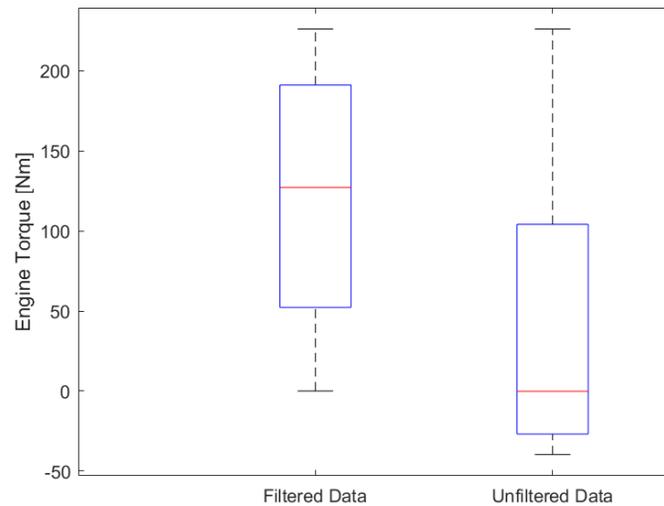


Figure 4.12: Comparison of the Distribution of Filtered and Unfiltered Engine Torque Data

In the figure, it can be seen that the unfiltered data contained ICE torque values that were negative while the filtered data only contains data that is positive. These negative values were filtered out to because the APP signal to the ECM would only need to command positive torque values from the ICE. The figure also shows the reduced span of the datapoints.

When using the regression to generate a pedal value on the vehicle, vehicle speed and engine speed were replaced with the instantaneous vehicle speed and engine speed. The engine torque was replaced with the torque value calculated in the controller model. The results of the engine model can be found in Figure 4.13.

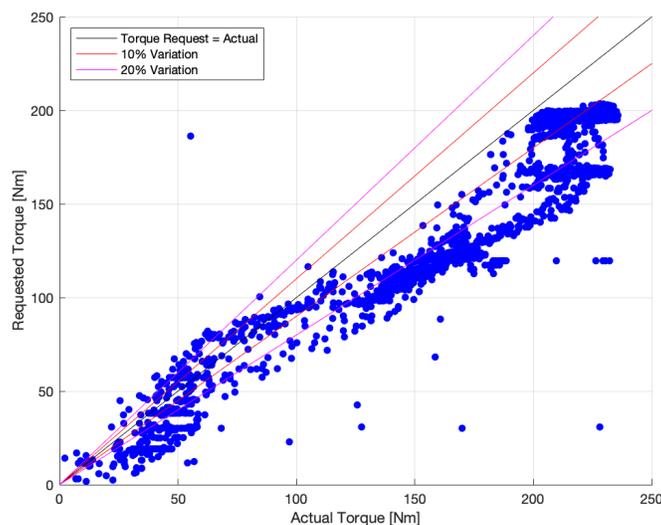


Figure 4.13: Plot of Requested Torque and Actual Torque with Variation Lines

This resulting engine model was used to translate engine torque request to APP for the purpose of controlling the engine torque request. As the figure shows, the torque value that the HSC was attempting to request mostly matched the value that the ICE output within a 20% range. When the values did not exactly match, the actual value of the ICE was greater than the value that the HSC was requesting.

4.6 Inverter and EM Control

4.6.1 CAN Input Signals

To control the inverter and EM, the HSC needed to monitor signals output from the inverter. These signals can be found in Table 4.1.

Table 4.1: Signals Received by the HSC from the Inverter

Signal Name	Description
Vehicle State Machine (VSM) State	Internal state machine output (states range between 0 and 15 as seen in Table 4.2). Indicates the internal state of the inverter.
EM Speed	Current speed of the EM.
Commanded Torque	Torque command read by the inverter.
Torque Feedback	Current EM torque output.
Enable State	Boolean value indicating the enable state of the inverter.
DC Bus Voltage	Current battery voltage, as seen by the inverter.
DC Bus Current	Battery current as seen by the inverter.

One of the most important signals in this list was the VSM State signal [27]. It was used to indicate the internal state of the inverter and the readiness of it to output torque commands. The state numbers and their meanings can be found in Table 4.2.

Table 4.2: VSM State Signal Descriptions

State Number	Description
0	Start up - 12 volt power up
1	Pre-charge initiated
2	Pre-charge active
3	Pre-charge complete
4	Waiting for EM direction activation (clockwise or counterclockwise)
5	Ready state - begins energizing the EM
6	EM running normally
7	Fault state
14	Shutdown in progress
15	Recycle power state - power to the controller needs to be cycled

4.6.2 CAN Output Signals

The inverter was used as an intermediary device to send commands to the EM. It was one of two devices that received CAN messages from the HSC. The signals sent from the HSC to the inverter can be found in Table 4.3.

Table 4.3: Signals Transmitted from the HSC to the Inverter

Signal Name	Description
Inverter Enable	Turns inverter and EM on and off.
Direction	Clockwise or counterclockwise rotation.
Speed Command	For use if speed mode is enabled. Commands motor speed output.
Torque Command	For use if torque mode is enabled. Commands motor torque output.
Inverter Discharge	On/Off signal to command inverter discharge.
Torque Limit Command	Limits the amount of torque. that can be commanded to the EM from the inverter.
Speed Mode Enable	Turns speed mode on and off. If speed mode if off, torque mode is enabled.

All of these signals were constant during Drive mode except for inverter enable and torque command. These two signals were used to turn on the inverter and EM and send the required torque commands to the EM. The direction command remained constant since the HSC did not output torque commands to the EM in any other transmission state outside of sport mode. Since the transmission did not change direction in sport mode, the direction command did not need to change.

The inverter discharge command remained constant because there was no requirement to discharge the inverter while using the vehicle. The speed command and speed mode command were set as constant values since the HSC output torque commands to the inverter and EM. The torque limit command was a constant value because the EM maximum torque was predetermined and preprogrammed into the inverter's software [28].

4.6.3 Inverter States and Control Flow

The inverter control system mainly consisted of a state machine to determine the necessary state of the inverter as well as the required output signals to the inverter based off of that state. Figure 4.14 shows an overview of the state machine used to control the inverter.

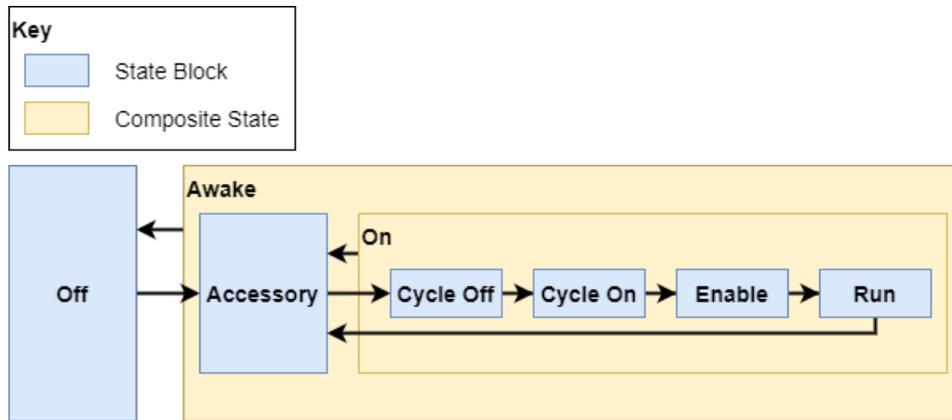


Figure 4.14: Inverter State Machine

Two signals were output from this state machine: the inverter enable signal and the EM torque command. The enable and torque command signals were sent over CAN to the inverter to enable and disable the device and command torques. The different states changed the output of these signals.

Off State The Off state was the state that the HSC defaulted to when the controller woke. To leave the state, the state machine needed to receive the wake request (Wake Request = 1) given by the Wake Controllers state from the Vehicle State Section (4.2). After leaving this state, the machine moved back into it when the wake request turned off (Wake Request = 0). The state machine output constant zero values for the inverter enable signal and the EM torque command.

Accessory State After the Off state, the state machine moved into the Accessory state. In this state, the inverter was awake, but not enabled, so it could not command torques from the EM. To leave this state and move into the Cycle Off state, the battery contactors had to be closed and the wake request had to equal one. This state output constant zero values for the enable signal and torque command. To re-enter this state from any of the On states, the contactors would need to reopen.

Cycle Off and Cycle On States In the next two states, Cycle Off and Cycle On, the inverter relay was turned off and back on. This was added in to reduce the occurrence of inverter faults since the procedure to remove the fault involved cycling power to the inverter. After cycling the power, the state machine moved into the next state. This state output constant zero values for the enable signal and torque command.

Enable State After the inverter's power cycle, it was ready to be enabled. In this state, the torque output remained zero, but the inverter enable output changed to one. Then, when the inverter reported a VSM state value equal to six, the state machine moved to the Run state.

Run State In the Run state, the torque command signal changed to equal the value calculated during the Torque Split Section (4.4). To leave this state, and enter the Accessory state, the VSM state much change to be a value other than six.

4.7 HV Battery Control

The main purpose of the HSC's battery control section was to open and close the contactors on the HV battery and indicate status of the battery's contactors to the rest of the control system. It also was used to indicate the fault state of the BMS. Figure 4.15 shows the battery state machine flowchart.

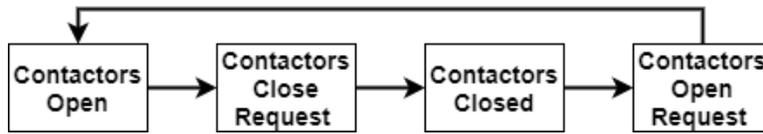


Figure 4.15: HV Battery State Machine

Contactors Open This was the default state of the battery state machine that the HSC started in. Otherwise, to enter this state, the HV battery's contactors would have to be opened. To leave this state, the state machine received a command from the Wake Controllers and Close Contactors states, in the Vehicle State Section (4.2).

Contactors Close Request In this state, the state machine output a command to the BMS to close the contactors on the battery. The controller left this state when the contactors closed.

Contactors Closed The controller would stay in this state until it received a command to open the contactors. Then, it moved to the next state.

Contactors Open Request In this state, the state machine would output a command to the BMS to open the contactors. It would remain in this state until they opened, and then the controller would move into the first state, the contactors open state.

4.8 State of Charge Calculations

After vehicle testing started, it was quickly realized that the BMS was an unreliable source of information for the battery's SOC. Instead of using the BMS's internal algorithms, Equation 4.1 [29] was used to estimate the SOC.

$$SOC(t) = SOC(t - 1) - \frac{I(t)}{C} * \Delta t \quad (4.1)$$

Where $I(t)$ was the instantaneous bus current, C was the battery capacity in coulombs, and Δt was the time between SOC calculations within the HSC. The initial value was assumed to be 50% for this calculation, and the initial value was reset in between AEs.

4.8.1 BMS and Calculated SOC Comparison

When comparing this calculation to the BMS reported value, it was clear that the calculated SOC was a better representation of the SOC than the reported value. Figure 4.16 contains a plot of Vehicle Speed (4.16a), EM Torque Output (4.16b), Calculated SOC (4.16c), and BMS Reported SOC (4.16d).

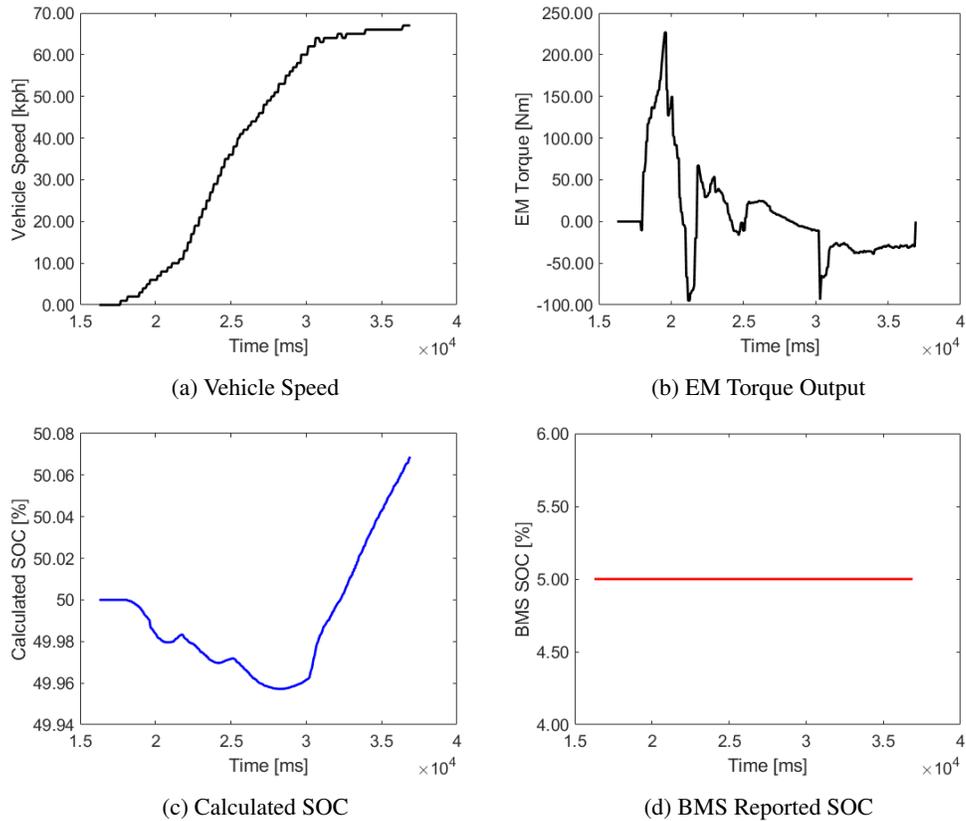


Figure 4.16: Comparison of Calculated SOC versus BMS Reported SOC

The figure shows that, for the given AE and EM torque output, the BMS did not report a change in SOC while the calculated SOC reported a dip in the SOC when the torque output was positive and an increase in SOC when the output was negative and regenerating charge. Due to the unreliability of the BMS's SOC algorithm, all SOC reported in this paper are calculated by the HSC using Equation 4.1.

Chapter 5

Testing

While testing the vehicle, it was necessary to prioritize the following: the safety of those near the vehicle, the preservation of the vehicle, and the validation of the HSC's code. To do this, the vehicle was put through many rounds of testing to incrementally validate it before testing everything together.

First, on-lift component testing was completed to test the individual components without letting the vehicle touch the ground. Then, on-lift vehicle testing was completed to test the functionality of the components as they worked together while keeping the vehicle off of the ground. After the on-lift testing was complete, the vehicle was tested on the ground in a parking lot and at an airfield.

By doing this, the people and vehicle were kept safe, and the controller was able to be tested. When bugs and faults were discovered, the issues were able to be resolved without injury or vehicle damage occurring.

5.1 On-Lift Component Testing

Before the components could be tested together, each section of the control system had to be tested individually to ensure each component performed as expected and to modify the control system to obtain the desired behavior. Performing these tests protected the vehicle from malfunctions that could have damaged the vehicle. To perform these tests, the vehicle was lifted into the air, just high enough off of the ground that the wheels did not touch the floor. This allowed for torque commands to be sent to the ICE and EM while remaining in control of the vehicle's location and environment.

5.1.1 HV Battery Control

The main point of testing the battery individually was to obtain a reliable method of communicating with the BMS to open and close the contactors on the battery. The DBC file that came with the BMS did not include the signal required to open or close the contactors, so that signal had to be found and added to the DBC.

Also, the status of the contactors and fault states were transmitted as an integer in one signal called the Traction Battery State signal. This testing allowed for the development of the method used to accurately convert this integer into the bits needed to indicate these statuses. The list of bits and their states can be found in Table 5.1.

Table 5.1: Battery State List

Bit Number (Intel)	Signal	Description
Bit 0	Fault State	Binary signal indicating the presence of a fault state in the battery.
Bit 1	Contactors K1	Indicates contactor K1, status as (open or closed).
Bit 2	Contactors K2	Indicates contactor K2, status as (open or closed).
Bit 3	Contactors K3	Indicates contactor K3, status as (open or closed).
Bit 4	Relay Fault	Binary signal indicating the presence of a relay fault in the battery.

Knowing these statuses allowed the HV battery control section of the HSC to signal contactor status to the other sections of the controller. It also provided information on when faults occurred and what type of faults occurred during the other tests performed.

During this and other tests it was discovered that the BMS was very unreliable as a source of information. It failed to balance the battery cells, lost communication with the cell boards, had a thermal event, and provided information that was less accurate than information that was reported by the inverter or that could be calculated in the HSC. Figure 5.1a shows the voltage signals reported by the inverter and BMS for the EM torque output (Figure 5.1b).

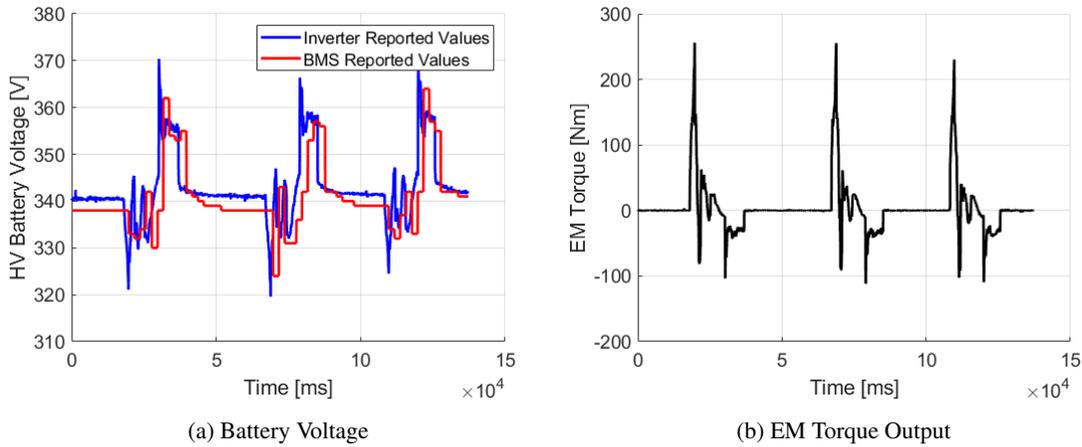


Figure 5.1: HV Battery Voltage for the EM Torque Output as Reported by the Inverter and BMS

From this figure, it can be seen that the BMS reported values were different than the values reported by the inverter. The values reported by the BMS did not fluctuate as much as the inverter. The BMS was also slower than the inverter at reporting changes in voltage. This information caused the HSC code to be modified to rely less on the values reported by the BMS and more on the values reported by the inverter.

5.1.2 EM Control

After solidifying control of the BMS, the EM and inverter tests could begin. To test, the contactors on the battery had to be closed and the transmission had to be shifted into neutral. Without the contactors being closed, the EM would

not be able to apply the commanded torque. If the transmission was not shifted into neutral, it could be damaged by the application of torque from the EM.

While trying to perform these tests, it was discovered that the inverter produced a large amount of Electromagnetic Interference (EMI) [23, 24] when the contactors were closed on the HV battery. This interfered with CAN communication between the HSC and added controllers and caused both the inverter and HV battery to produce faults. The inverter was moved to a second CAN bus to prevent this from occurring and to increase the reliability. Also, both CAN buses were shielded and grounded to further reduce noise. It was also found that cycling the power to the inverter before enabling it to run increased the reliability of the inverter and reduced faults.

The first EM control tests performed were limited to sending a constant torque value to the inverter. It took a command of 8 Nm to break the static friction, then the value could be dropped lower to keep the EM outputting a reasonable amount of torque while on the lift. The torque was kept low for the safety of those around the vehicle and for the protection of the vehicle while it was on the lift. Adding or removing too much torque too quickly could have caused the vehicle's center of gravity to shift and could have resulted in the vehicle falling off of the lift.

Doing this test allowed the direction command to be finalized before the EM torque was combined with the ICE torque. It also allowed the HV battery voltage and current to be monitored in a controlled setting. A variable torque command was sent to the inverter after constant command testing was completed. The HSC torque output was capped to a very low value. After ironing out the problems with the inverter with the previous testing, no problems were discovered with this test.

5.1.3 APP Control

The APP testing was completed to ensure the APP outputs from the HSC matched the values reported by the Toyota gateway. This was completed in accessory mode with all of the other components off.

5.1.4 ICE Control

Once the on-lift, APP testing was complete, the ICE testing could begin. To test the ICE, the contactors on the HV battery were opened, and all HV components were off. The APPS wires were configured so that the HSC intercepted the signals on their route to the ECM. This allowed the HSC to control the engine torque output through the APP signal.

The vehicle was keyed on to start the engine, the vehicle was shifted into sport mode, and the APP was changed through the use of the accelerator pedal. From this, the changing ICE output was used to validate ICE control.

While on the lift, the ICE torque output accuracy was not able to be determined due to the vehicle's traction control system limiting the ICE torque output. Therefore, this test was used only to validate that the ICE was able to

be controlled and not that the ICE was able to be controlled accurately.

5.2 On-Lift Vehicle Testing

The on-lift vehicle testing involved testing all of the components together. It meshed the EM torque and ICE torque outputs, at limited values. To limit these outputs, the APP torque output was capped at 10% and the EM torque was capped at 8 Nm. A driver was sitting in the driver's seat, changing the APP input.

The Power Mode section of the controller was modified to work as desired, then validated based off of the results of this testing to ensure the state machine would move into the Off, Accessory, and On states with the vehicle. The Vehicle State section of the controller was also modified and validated with this test by checking to make sure that the state machine moved from off to accessory mode with the vehicle, and then moved through the sequence to wake the controllers, close the contactors, and ended in the state that allowed for the torque to be split between the ICE and EM. Neither Baseline nor PAE was used to split the torques, instead a constant torque split was applied to the driver input torque value.

Then, once the vehicle was in a state where it could output torque commands to the ICE and EM, the torque was commanded through the use of the APP input signal from the driver. The on-lift tests that involved the ICE were used mostly to test that the ICE could output torque when commanded. This was not used to validate the accuracy of the torque output since the vehicle's traction control limited the amount of torque the engine could output while off of the ground.

5.3 Parking Lot Testing

Once the on-lift testing was completed, the vehicle could be moved off of the lift for the parking lot tests that were designed to observe the behavior of the vehicle in motion through low speed testing. This testing prevented the traction control issue from occurring since the vehicle's wheels were off of the ground.

One of the main goals of the parking lot tests was to validate the combination of EM and ICE torque. Two methods were used to complete this validation work: capping torque output and defining a specific torque split percentage between the ICE and EM.

To begin, the torque was split to send 90% to the ICE and 10% to the EM. The EM torque was also capped at 15 Nm to begin, as extra layer of protection to the system to ensure that the vehicle was behaving as expected before the EM torque commands were allowed to increase. Once the testing confirmed that the vehicle behavior was as expected, the controller's torque split changed to an 80/20 split (where 80% of the torque was commanded from the ICE and 20% was commanded from the EM). As the vehicle demonstrated capability, the torque output again changed to a

70/30 split, a 60/40 split, and then a 50/50 split.

After the 50/50 split, the validation work switched to EM regeneration, where the EM was sent negative torque commands to validate the range of regeneration that was expected. First, EM torque commands were limited to -15 Nm. Then, as the testing progressed, the limit decreased to -30 Nm, -50 Nm, and so on, until the limit reached -200 Nm.

5.4 Airfield Testing

The bulk of the Baseline and PAE comparison testing was completed at the Christman Airfield, a 4000 foot long runway that was used as a Colorado State University (CSU) testing facility. This airfield was used for closed-course, straight-line testing. It runs in a north-south line, and every test that was completed with this vehicle was in the north direction. This reduced the effects of the slight slope of the runway.

Different versions of both the Baseline and PAE control strategies were tested here to debug the vehicle and control strategies. This was necessary to ensure the vehicle would accelerate according to the defined AE without causing systems like the traction control to come online. It was discovered that accelerating too quickly would cause the traction control to interfere with the behavior of the ICE. To prevent this, the AE trace was modified to have a slower rate of acceleration.

The control strategies also had to be modified to ensure that they were being utilized to their fullest potential. The PAE control strategy was tested using the optimal control matrix method and the torque trace strategy. The optimal control matrix method, reduced in size and scope due to HSC memory limitations, was discovered to output torque split values that didn't prioritize recharging the battery to the starting SOC. Instead, it would command large, positive torque outputs from the EM.

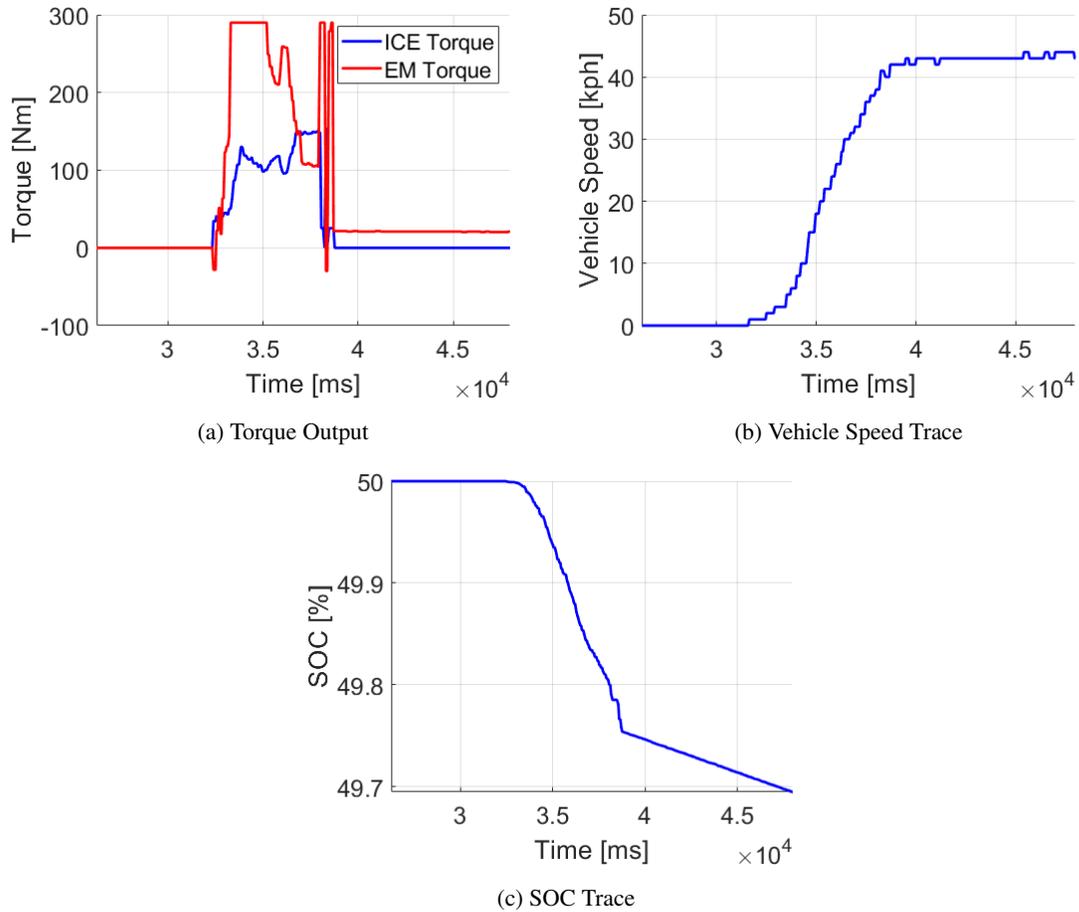


Figure 5.2: Optimal Matrix Example Plots

The figures above show one of the tests completed with the optimal control matrix generating the torque split between the EM and ICE (Figure 5.2a) for the AE (Figure 5.2b). As seen in the figure, the EM torque was applied at a much larger amount than the ICE torque. If the optimal control matrix was working correctly here, the EM torque output would have been negative during the AE to increase the SOC to ensure that the SOC at the beginning and end of the AE were equal.

The Optimal Control Torque Splitting Method was set aside in favor of more testing of the torque trace PAE method. This method allowed for the full, optimal control matrix to be used in the model to pre-calculate the ICE and EM torque values for the specified AE. The resulting SOC and Fuel Consumption (FC) output could be directly compared to the PAE model to compare the behavior to. Also, this allowed the PAE to be tested in a simple, repeatable manner so it could be compared to the Baseline Torque Splitting Method.

The AE that was created to avoid faults in the vehicle and to test the torque splitting methods repeatably can be found in Figure 5.3. The AE went from 0 to 65 kpm, and was able to be completed three times, with deceleration events to bring the vehicle speed back to zero, without driving off of the end of the runway. Within the length of the

runway, there was room to squeeze one more AE in, but doing this required decelerating very quickly. Adding a fourth AE without a fast deceleration, would result in the vehicle leaving the runway and driving onto the grass beyond it. Due to this, three AEs were performed in one test event.

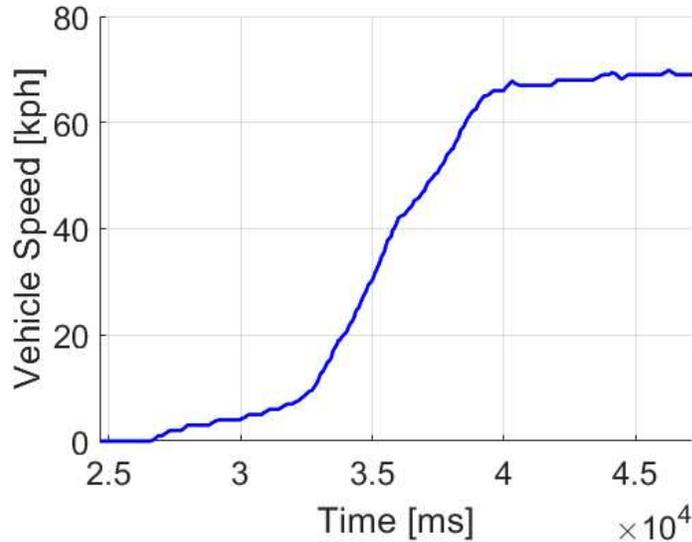


Figure 5.3: Vehicle Speed Trace

Both the Baseline and PAE torque splitting methods followed similar vehicle speed traces to the trace shown in the figure. To obtain repeatable results, the driver input was replaced with predetermined traces for both methods. For Baseline, the driver APP input was replaced with a trace. For PAE, the ICE and EM torques were replaced with pre-computed traces.

These were not replaced by the same type of signal due to the way that the torque split methods functioned. With Baseline, the torque split could be calculated on-vehicle, so the APP input was needed in order to compute the torques. To obtain the vehicle speed trace, a the vehicle was manually driven through the AE, and the APP was captured. This captured APP signal was used as the Baseline input.

However, since PAE must be precomputed, and the HSC did not have enough memory to contain the full optimal control matrix necessary for the torque split to properly occur, the torque traces had to be precomputed and fed into the HSC. The following steps were taken to begin an AE:

1. Shift the vehicle into sport mode.
2. Toggle the leftmost dash switch (Figure 5.4) on and then off. This resets the internal clock used to progress through the trace and sets the SOC back to 50%.
3. Flip the middle dash switch on and leave it on. This switch caused the internal clock to begin progressing. By leaving it on, the vehicle was able to complete the AE.

4. Wait until the vehicle has reached the maximum speed of the AE, and hold for 10 seconds.
5. Flip the middle dash switch off.
6. Press down on the brake pedal until the vehicle has come to a complete stop.



Figure 5.4: Dash Switches

The Baseline and PAE tests were driven back to back. The shift schedule shown in Table 5.2 was used for both test types. To create this shift schedule, the vehicle was driven through AEs in manual mode. It was shifted into the next gear when the vehicle indicated that it needed to shift. This usually happened through noises that the engine made. The shift points typically happened around the speed values shown in the table, so they were standardized among the vehicle's drive cycles.

Table 5.2: Shift Schedule

Gear	Speed Range
1	0-13 mph, 0-21 kph
2	13-23 mph, 21-37 kph
3	23-33 mph, 37-53 kph
4	33+ mph, 53+ kph

As the table shows, the first four gears of the transmission were needed to complete the acceleration event since the vehicle did not reach a high enough speed to require the fifth gear. Once all of the logistics of velocity trace, shift schedule, and course were tested and finalized, the two torque split methods could be tested, in back-to-back runs, gather the data to compare against each other.

5.4.1 Results

From the tests, it was found that the torque trace PAE method improved the FE by an average of 7.31%. To get to this number, the amount of energy used for the time that it took to complete the AE was converted into a gallon equivalent amount. This was then combined with the fuel usage of the ICE and was used to divide the distance driven to obtain

the Miles Per Gallon (MPG) value. This value could then be used for comparison. For two PAE AEs and two Baseline AEs, the engine torque was plotted against the engine speed on the BSFC map developed previously. This can be seen in Figures 5.5 and 5.6.

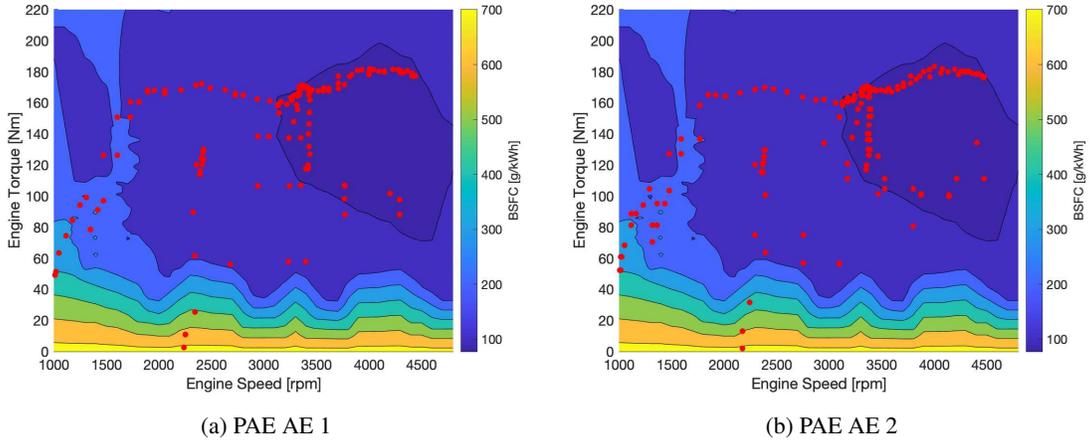


Figure 5.5: BSFC Plots for PAE AEs

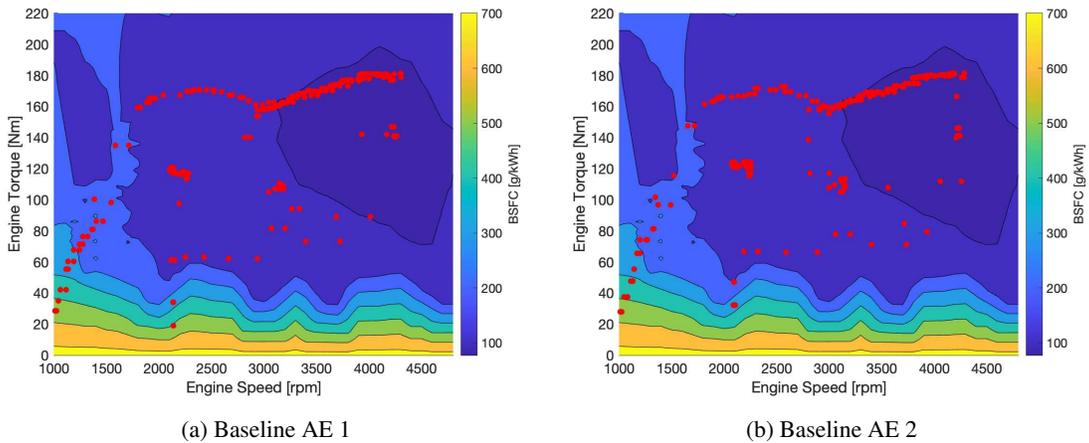


Figure 5.6: BSFC Plots for Baseline AEs

The PAE controlled AE shown in Figure 5.5a spent 42.9% of the event in the darkest blue, or most fuel efficient zone. Figure 5.5b shows the next PAE AE which spent 43.7% of the time in that state. Figure 5.6a's AE, which used Baseline control, spent 29.6% of the event in the dark blue zone. The Baseline controlled AE in Figure 5.6b spent 29.1% in the fuel efficient zone. Using this information with the BSFC plots, it can be seen that the AEs using PAE control spend more time with the ICE in the darkest blue, or most fuel efficient, section of the map than the Baseline AEs. The PAE control method was commanding the ICE into the more fuel efficient states during the AE allowing the vehicle to improve the FE by an average of 7.31%.

The vehicle used for this project was difficult to obtain results with due to the unreliability of base vehicle control

while quickly accelerating using the electric motor, so three tests each from baseline and PAE were determined to be acceptable for this analysis. These cases were removed due to the tendency of the engine to fault which caused the ICE torque output to equal zero while the APP commands were nonzero. An example of this can be seen in Figure 5.7.

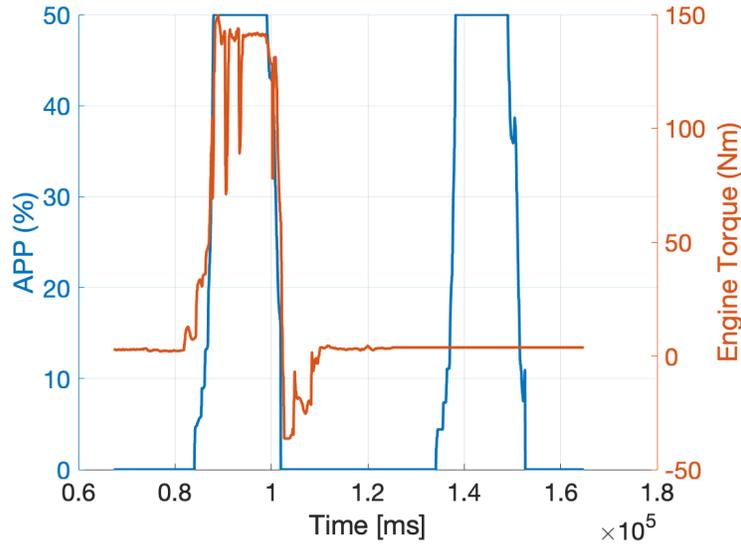


Figure 5.7: ICE Fault Causing ICE Torque Output To Equal Zero While APP Commands Were Nonzero

In the figure above, the ICE first outputs nonzero engine torque values while the APP value is nonzero. After the APP dropped back to zero, the ICE is then commanded to output torque using the same APP trace as the previous attempt. However, this second attempt produced no torque output from the ICE, and this attempt had to be removed from the test data. The second event was completed immediately after the first, and no vehicle settings were changed between the two events.

Since no directly controllable settings were changed between the two events, it was determined that this vehicle behavior was caused by a uncontrollable setting internal to vehicle. Testing was halted to attempt to find a solution to this problem. Further testing was prevented by the weather.

The following table shows the percentage increase of PAE’s FE over baseline for the 0-70 kph AE:

Table 5.3: Comparison of FE Change for PAE and Baseline

AE	PAE Percentage Increase Over Baseline
1	7.7%
2	5.8%
3	8.4%
Average	7.3%

After performing a Student’s t-test, comparing the samples from baseline and PAE, the $P(\zeta_t)$ value was 0.008975

meaning that the hypothesis that the two groups represent different populations can be accepted at 95% confidence.

Chapter 6

Discussion

6.1 Implications of PAE on Hybrid Vehicle Control

Currently, all of the HEVs on the road currently use instantaneously optimized control to be able to control the powertrain of the vehicle [13]. With this method, the calculations to optimize the vehicle's FE occur instantaneously on the vehicle. This control method is used due to its overall simplicity in implementation. This was seen in the relative ease in creating the baseline control method on the vehicle as compared to the PAE method of control. The instantaneous control method was easier to troubleshoot and required less information about the vehicle and the drive cycle than the PAE method.

Predictive optimal control methods take the entire drive cycle into account when calculating the optimal control for the vehicle. This was more difficult to build into the control method due to the increased knowledge of the vehicle and drive cycle that was needed. However, PAE was an implementable way of improving the FE of HEVs, using information from the predicted vehicle trajectory. This research demonstrated that these strategies were feasible and were able improve the FE of HEVs over the instantaneous control by 7.3%.

6.2 Implications of Robust, Map-Based Predictive Powertrain Control

Nearly every study of predictive powertrain control uses complicated and computationally expensive ways to optimize the operation of the powertrain [30, 13]. These studies include methods like DP, machine learning, and model predictive control to determine the optimal control for the vehicle.

In this study, a map-based control method, based on offline learning was used to realize a pseudo-optimal control that was robust to disturbances and realized measurable fuel economy gain. The control system was able to use the pre-computed PAE values as a lookup table to determine the optimal torque output for the ICE and EM. This meant that by pre-computing the PAE map, the PAE control could occur on the Motohawk control hardware with the rest of the supervisory control algorithm. On this vehicle, the map had to be simplified and reduced in size to fit the constraints of the controller hardware, but this simplified version of the map was effective at improving the vehicle's FE.

6.3 Implications for Vehicle Testing and Demonstration

Testing this vehicle marked the first time in literature that a test vehicle was able to demonstrate the FE benefits of predictive control algorithms [13]. The previous literature has calculated, modeled, and simulated predictive control and hypothesized the fuel economy benefits while this paper demonstrated the actual improvements in FE for a PAE control when compared to a baseline, instantaneous control.

From this, it can be seen the PAE algorithm improved the vehicle's FE over the baseline FE in a real world environment. It shows that, even with reduced knowledge of the vehicle, this type of control can improve the FE over baseline. This suggests that with a greater knowledge and control of the ICE, these FE improvements are likely to be even greater.

Chapter 7

Conclusion

As discussed previously, to reduce greenhouse gas emissions from the transportation sector, the sector with the largest share of emissions, vehicles should become more fuel efficient. HEVs can be used to improve the FE of vehicles over vehicles that only contain an ICE. Electrified vehicles can further improve their fuel economy through the use of their torque split control algorithms.

In this study, the PAE control method produced FE improvements of 7.3% over a baseline, instantaneous torque control method. This was done by using DP to precalculate the optimal torque split between the ICE and EM. This method used the ICE's BSFC map to deliberately place the ICE into fuel efficient states in order to improve the FE. Future research on this topic should be done on the FE improvements on a vehicle undergoing a full drive cycle, such as the US06. This would allow the vehicle to use more of the battery since a US06 provides more opportunities to apply both positive and negative torque from the EM while beginning and ending the drive cycle at a given SOC.

References

- [1] European Environmental Protection Agency, “CO2 emissions from cars: Facts and figures (infographics): News: European parliament.” <https://www.europarl.europa.eu/news/en/headlines/society/20190313ST031218/co2-emissions-from-cars-facts-and-figures-infographics>.
- [2] U.S. Department of Energy Office of Energy Efficiency and Renewable Energy, “Reduce climate change.” <https://www.fueleconomy.gov/feg/climate.shtml>.
- [3] W. F. Lamb, T. Wiedmann, J. Pongratz, R. Andrew, M. Crippa, J. G. Olivier, D. Wiedenhofer, G. Mattioli, A. Al Khourdajie, J. House, *et al.*, “A review of trends and drivers of greenhouse gas emissions by sector from 1990 to 2018,” *Environmental research letters*, 2021.
- [4] U.S. Environmental Protection Agency, “Sources of greenhouse gas emissions.” <https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions>.
- [5] H. Dong, M. Xue, Y. Xiao, and Y. Liu, “Do carbon emissions impact the health of residents? considering China’s industrialization and urbanization,” *Science of The Total Environment*, vol. 758, p. 143688, 2021.
- [6] D. Peters, J. Schnell, P. Kinney, V. Naik, and D. Horton, “Public health and climate benefits and trade-offs of us vehicle electrification,” *GeoHealth*, vol. 4, no. 10, p. e2020GH000275, 2020.
- [7] K. L. Ebi and J. J. Hess, “Health risks due to climate change: Inequity in causes and consequences: Study examines health risks due to climate change.,” *Health Affairs*, vol. 39, no. 12, pp. 2056–2062, 2020.
- [8] H. Dia, “Rethinking urban mobility: unlocking the benefits of vehicle electrification,” in *Decarbonising the built environment*, pp. 83–98, Springer, 2019.
- [9] J. Li, P. Zhang, K. Lo, M. Guo, and M. Wang, “Reducing carbon emissions from shopping trips: Evidence from China,” *Energies*, vol. 8, no. 9, pp. 10043–10057, 2015.
- [10] S. McLeod, J. Scheurer, and C. Curtis, “Urban public transport: Planning principles and emerging practice,” *Journal of Planning Literature*, vol. 32, no. 3, pp. 223–239, 2017.
- [11] A. Al-Samari, “Study of emissions and fuel economy for parallel hybrid versus conventional vehicles on real world and standard driving cycles,” *Alexandria Engineering Journal*, vol. 56, no. 4, pp. 721–726, 2017.

- [12] S. Sarvaiya, S. Ganesh, and B. Xu, “Comparative analysis of hybrid vehicle energy management strategies with optimization of fuel economy and battery life,” *Energy*, vol. 228, p. 120604, 2021.
- [13] Z. D. Asher, A. A. Patil, V. T. Wifvat, A. A. Frank, S. Samuelson, and T. H. Bradley, “Identification and review of the research gaps preventing a realization of optimal energy management strategies in vehicles,” *SAE International Journal of Alternative Powertrains*, vol. 8, no. 2, pp. 133–150, 2019.
- [14] Q. Wang and A. Frank, “Plug-in HEV with CVT: configuration, control, and its concurrent multi-objective optimization by evolutionary algorithm,” *International Journal of Automotive Technology*, vol. 15, no. 1, pp. 103–115, 2014.
- [15] D. A. Trinko, “Predictive energy management strategies for hybrid electric vehicles applied during acceleration events,” Master’s thesis, Colorado State University, 2019.
- [16] D. A. Trinko, Z. D. Asher, and T. H. Bradley, “Application of pre-computed acceleration event control to improve fuel economy in hybrid electric vehicles,” tech. rep., SAE Technical Paper, 2018.
- [17] C.-C. Lin, J.-M. Kang, J. Grizzle, and H. Peng, “Energy management strategy for a parallel hybrid electric truck,” in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 4, pp. 2878–2883 vol.4, 2001.
- [18] L. Serrao, S. Onori, and G. Rizzoni, “ECMS as a realization of pontryagin’s minimum principle for hev control,” in *2009 American Control Conference*, pp. 3964–3969, 2009.
- [19] Z. D. Asher, D. A. Trinko, J. D. Payne, B. M. Geller, and T. H. Bradley, “Real-Time Implementation of Optimal Energy Management in Hybrid Electric Vehicles: Globally Optimal Control of Acceleration Events,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, 03 2020. 081002.
- [20] J. Wu, J. Ruan, N. Zhang, and P. D. Walker, “An optimized real-time energy management strategy for the power-split hybrid electric vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1194–1202, 2019.
- [21] C. Liang, X. Xu, F. Wang, and Z. Zhou, “Coordinated control strategy for mode transition of DM-PHEV based on MLD,” *Nonlinear Dynamics*, vol. 103, no. 1, pp. 809–832, 2021.
- [22] A. Rabinowitz, F. M. Araghi, T. Gaikwad, Z. D. Asher, and T. H. Bradley, “Development and evaluation of velocity predictive optimal energy management strategies in intelligent and connected hybrid electric vehicles,” *Energies*, vol. 14, no. 18, p. 5713, 2021.

- [23] D. Adelman, "Post-transmission parallel hybrid vehicle design and validation for predictive acceleration event energy management strategies," Master's thesis, Colorado State University, 2021.
- [24] B. Mckenney, "Comparison of design and implementation of hybrid systems in prototype vehicles," Master's thesis, Colorado State University, 2021.
- [25] P. Subke, *Diagnostic communication with road-vehicles and Non-Road Mobile Machinery*. SAE International, 2019.
- [26] A. Rabinowitz, F. M. Araghi, T. Gaikwad, Z. D. Asher, and T. H. Bradley, "Development and evaluation of velocity predictive optimal energy management strategies in intelligent and connected hybrid electric vehicles," *Energies*, vol. 14, no. 18, 2021.
- [27] C. Brune, A. Khan, and T. Gintz, *RMS PM Software User Manual*. Cascadia Motion, 7929 SW Burns Way Suite F, Wilsonville, OR 97070, 3.6 ed., 1 2021.
- [28] C. Brune and A. Khan, "Rms pm software user manual," Jan 2016.
- [29] W.-Y. Chang, "The state of charge estimating methods for battery: A review," *International Scholarly Research Notices*, vol. 2013, p. 7, 2013.
- [30] M. Sabri, K. A. Danapalasingam, and M. F. Rahmat, "A review on hybrid electric vehicles architecture and energy management strategies," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 1433–1442, 2016.

Appendix A: PAE Figures

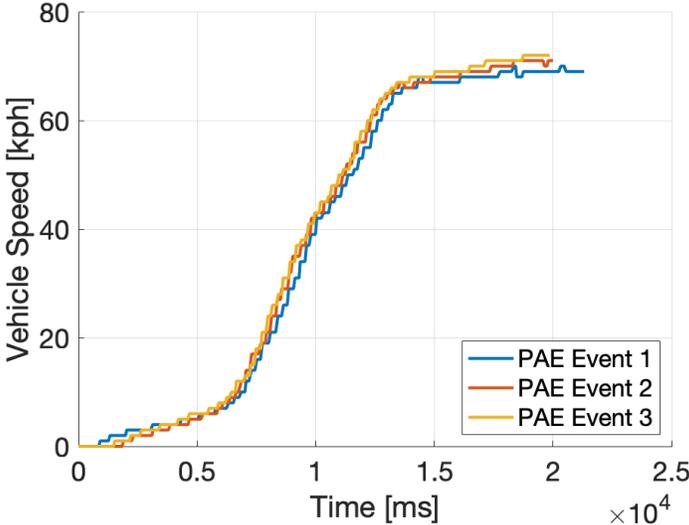


Figure A1.1: Vehicle Velocity Comparison for Three PAEs

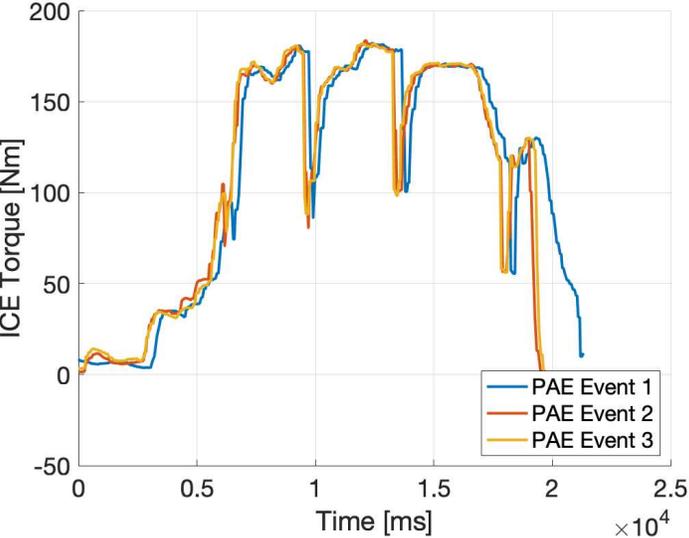


Figure A1.2: ICE Output Comparison for Three PAEs

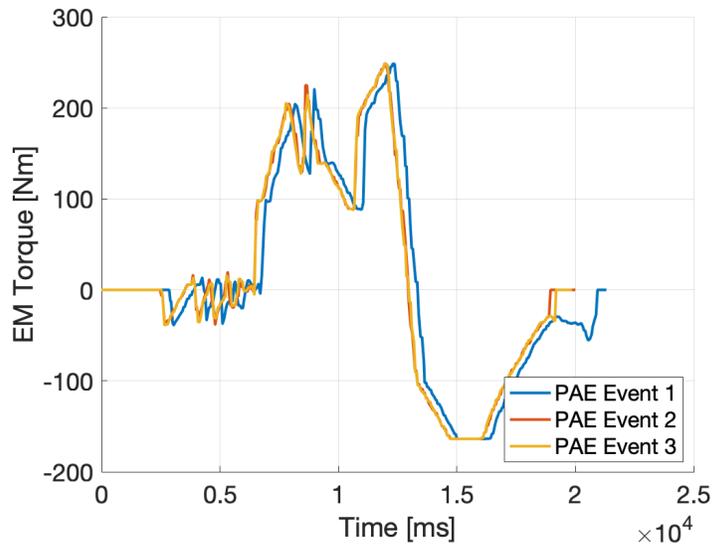


Figure A1.3: EM Output Comparison for Three PAEs

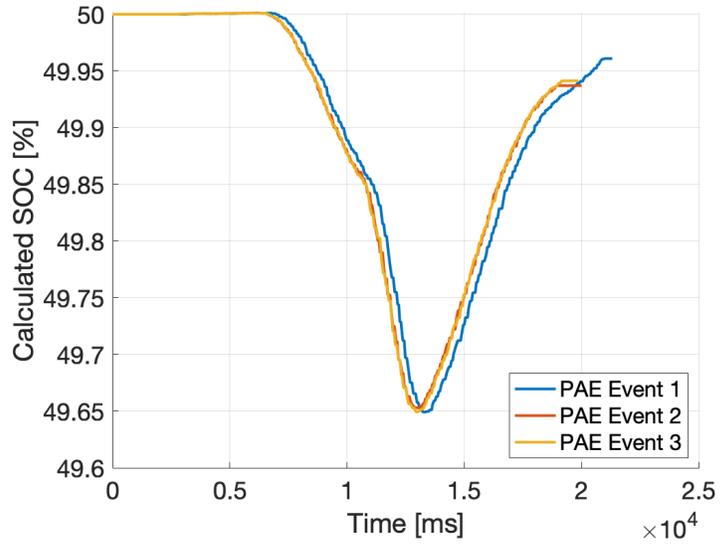


Figure A1.4: HV Battery SOC Comparison for Three PAEs Events with the SOC Reset to 50% After Each AE

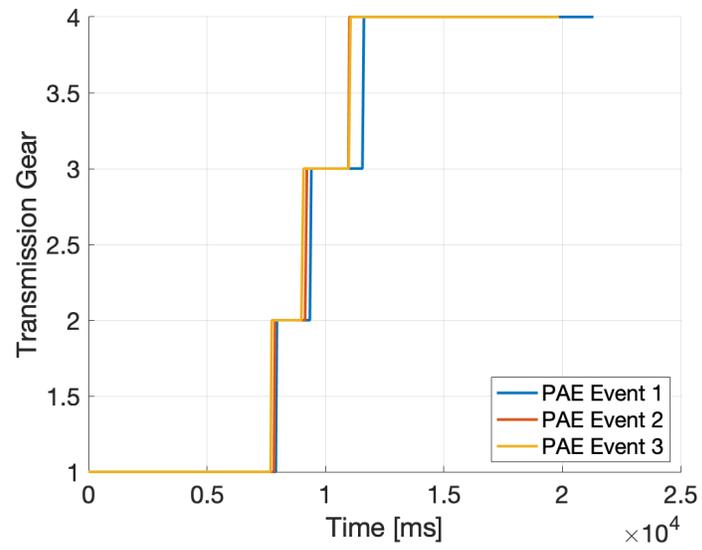


Figure A1.5: Transmission Gear Value Comparison for Three PAEs

Appendix B: Baseline Figures

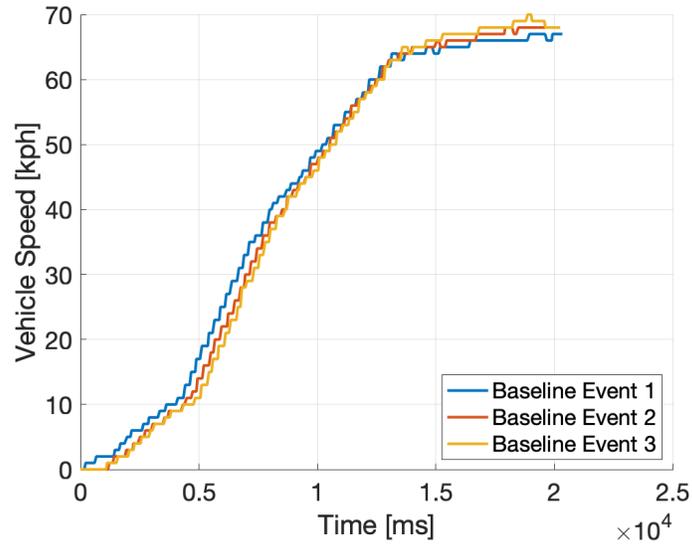


Figure B1.1: Vehicle Velocity Comparison for Three Baseline AEs

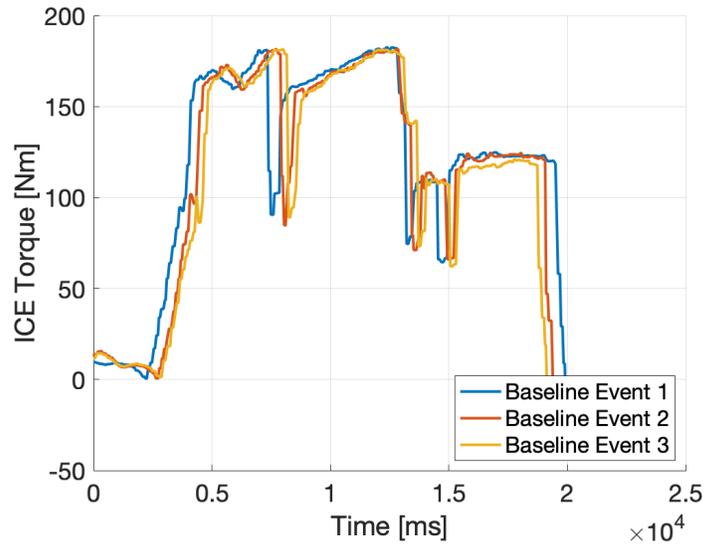


Figure B1.2: ICE Output Comparison for Three Baseline AEs

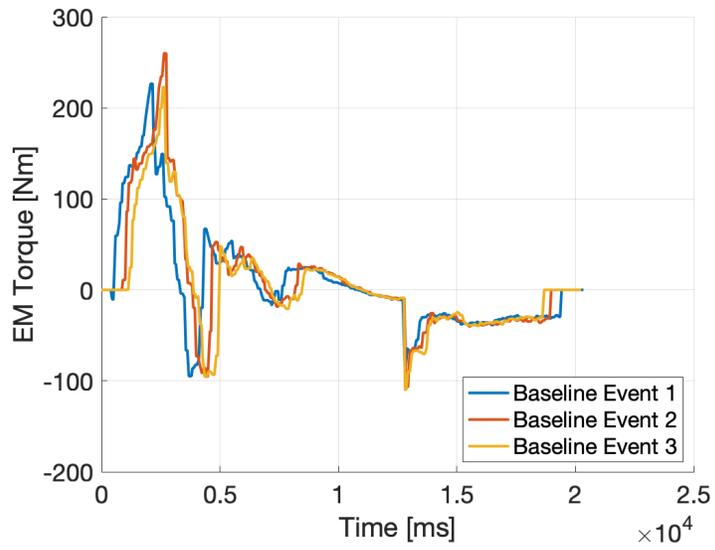


Figure B1.3: EM Output Comparison for Three Baseline AEs

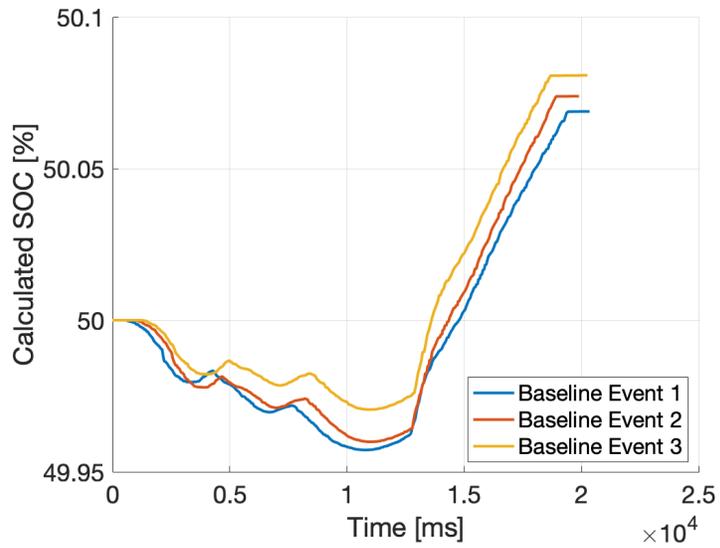


Figure B1.4: HV Battery SOC Comparison for Three Baseline AEs with the SOC Reset to 50% After Each AE

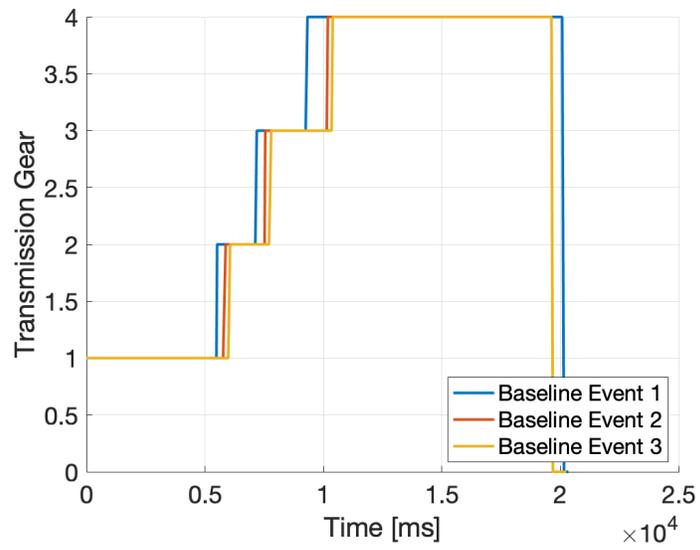


Figure B1.5: Transmission Gear Value Comparison for Three Baseline AEs