

## DEVELOPMENT OF A ROBOTIC VEHICLE CONTROL SYSTEM



# DEVELOPMENT OF A ROBOTIC VEHICLE CONTROL SYSTEM

By

Matthew C. Johnson, B.Eng

A Thesis

Submitted to the School of Graduate Studies

in partial fulfilment of the requirements for the degree of

Masters of Applied Science

Department of Computing and Software

McMaster University

© Copyright by Matthew C. Johnson, September 2013

## DESCRIPTIVE NOTE

MASTERS OF APPLIED SCIENCE (2013) McMaster University

(Software Engineering)

Hamilton, Ontario

TITLE: DEVELOPMENT OF A ROBOTIC VEHICLE CONTROL SYSTEM

AUTHOR: Matthew C. Johnson, B.Eng. (McMaster University)

SUPERVISOR: Dr. M. v. Mohrenschildt, P.Eng

NUMBER OF PAGES: xvi, 169

## ABSTRACT

This thesis presents the design stages in creating a robotic driving system for performing in-lab driving schedule playback using a chassis dynamometer. This equipment is intended to assist research in improving the power train system in Hybrid Electric Vehicles (HEV). The design stage layout contributes to showing how to effectively breakdown a mechatronics related project into manageable steps. The process includes background research, system requirements, system design and validation. Design stages are further broken up into three subsystems, mechanical, electrical and software.

Two actuators control the gas and brake pedals of the test vehicle. An active control system allows the vehicle to follow a speed vs. time driving schedule. The control feedback loop uses two cascading Proportional-Integral (PI) controllers (vehicle speed and pedal position). Feedback signals come from the onboard diagnostics (OBD-II) port. The control software is implemented on a dSPACE MicroAutoBox capable of multiple inputs and outputs including a built in CAN Bus controller to receive messages from the OBD-II port. The control software is implemented in Simulink and provides a modular, maintainable architecture for future development.

The system design steps lead to a practical system obtained through a systematic approach. Design documentation will allow for further development of this test system to meet future requirements.

## ACKNOWLEDGEMENTS

I would like to thank Dr. Martin von Mohrenschildt for his patience and support through the past 4 years of this 2 year degree. It was a very rewarding experience to work with Martin. A thank you goes to the Ford Motor Company for providing the test vehicle and support with communication to the vehicles ECU's. Thank you to my friends and family for their support. A big thanks goes out to Alan Sawula for the coffee breaks and being there to bounce ideas off of. Finally, thank you, the reader, for taking the time to view this thesis. – Good luck and Godspeed

---

## CONTENTS

1	Introduction.....	1
1.1	Thesis Motivation.....	1
1.2	System Components.....	2
1.3	Thesis Layout.....	7
2	Background.....	9
2.1	Hybrid Vehicles.....	9
2.1.1	Serial and Parallel Hybrid Vehicles.....	10
2.1.2	E-CVT.....	12
2.2	CAN Bus in Automobiles.....	15
2.3	Dynamometer.....	18
2.4	Combined Resistive Forces while Driving.....	20
2.4.1	Dynamic Drag.....	22
2.4.2	Rolling Resistance.....	22
2.4.3	Equations.....	23
2.5	Commercial Driving Robots.....	25
2.5.1	Stahle SAP-2000.....	25
2.5.2	Horiba ADS 7000.....	28
2.6	Driving Cycles.....	30
3	Requirements.....	33
3.1	Assumptions.....	33
3.2	Mechanical Requirements and Constraints.....	34
3.2.1	Force Output for Gas.....	34

---

- 3.2.2 Force Output for Brake.....34
- 3.2.3 Size Constraints.....36
- 3.2.4 Actuation Distance for Gas Pedal.....39
- 3.2.5 Actuation Distance for Brake Pedal .....39
- 3.2.6 Minimum Speed for Actuation.....39
- 3.2.7 Summary of Mechanical Requirements .....41
- 3.3 Electrical System Requirements .....42
  - 3.3.1 E-Stop.....42
  - 3.3.2 Manual Control.....42
- 3.4 Controller Requirements.....42
  - 3.4.1 Monitored Quantities.....43
  - 3.4.2 Controlled Quantities .....43
- 3.5 Safety Requirements .....44
  - 3.5.1 Modular design.....44
  - 3.5.2 Operation.....44
  - 3.5.3 Battery Detection.....44
- 4 System Design.....45
  - 4.1 Overview.....45
  - 4.2 Mechanical Design .....45
    - 4.2.1 Objective .....45
    - 4.2.2 Initial Design Ideas.....45
    - 4.2.3 Design Decisions.....48
    - 4.2.4 First Prototype Design.....53
    - 4.2.5 Second Prototype Design .....55

---

4.2.6	Final Design .....	57
4.2.7	Preliminary Design Verification .....	58
4.2.8	Cost .....	61
4.3	Electrical Design .....	62
4.3.1	Objective .....	62
4.3.2	System Module Breakdown .....	62
4.3.3	Vehicle Interfaces.....	67
4.3.4	dSPACE Interface Description .....	69
4.3.5	Detailed dSPACE Connections.....	71
4.3.6	Power Supply .....	76
4.3.7	Schematic Design for Custom Hardware.....	78
4.3.8	Cost .....	89
4.4	Software Design .....	91
4.4.1	Objective .....	91
4.4.2	Data Flow Breakdown.....	91
4.4.3	User Interface Design.....	95
4.4.4	Implementation Level 1 – System_Testing.mdl .....	97
4.4.5	Implementation Level 2 .....	101
4.4.6	Implementation Level 3 .....	124
4.4.7	Implementation Level 4 .....	131
4.4.8	Interface Implementation .....	134
5	System Validation.....	137
5.1	In-Lab Tests.....	138
5.1.1	Mechanical .....	138

---

- 5.1.2 Electrical.....139
- 5.1.3 Software.....145
- 5.2 Field Testing .....149
  - 5.2.1 Mechanical .....149
  - 5.2.2 Electrical.....150
  - 5.2.3 Software.....150
  - 5.2.4 Driving Cycle Recording.....152
  - 5.2.5 Dynamometer Setup .....153
  - 5.2.6 Ease of use.....153
- 5.3 Validation Matrix.....153
- 6 Conclusion .....155
  - 6.1 Accomplishments .....155
  - 6.2 Future Work.....157
    - 6.2.1 Mechanical .....157
    - 6.2.2 Electrical.....157
    - 6.2.3 Software.....158
- 7 Appendix.....159
  - 7.1 Mechanical Drawings .....159
  - 7.2 Printed Circuit Board Design.....164
- 8 Bibliography.....167

## LIST OF FIGURES

Figure 1.1 2010 Ford Escape Hybrid .....	3
Figure 1.2 Vehicle Spy 3 - <a href="http://intrepidcs.com/VehicleSpy/index.html">http://intrepidcs.com/VehicleSpy/index.html</a> .....	5
Figure 1.3 dSPACE MicroAutoBox 1401/1501 .....	6
Figure 1.4 System Overview.....	7
Figure 2.1 Parallel powertrain used in Ford Hybrids (Miller, 2006) .....	11
Figure 2.2 Continuous Variable Transmission Representation (Mitsubishi, 2010).....	13
Figure 2.3 Power split e-CVT of the input split type (Miller, 2006) .....	14
Figure 2.4 CAN Physical Layer ISO 11898-2 (CiA, 2011).....	16
Figure 2.5 CAN Bus Frame .....	16
Figure 2.6 Dynamometer Setup at McMaster Innovation Park .....	19
Figure 2.7 Forces acting on a vehicle (Ehsani, 2005).....	20
Figure 2.8Stahle SAP-2000.....	25
Figure 2.9 Horiba ADS 7000 (Horiba, 2010) .....	28
Figure 2.10 European Combined Drive Cycle (NZ Government, 2009).....	31
Figure 2.11ADR drive cycle speed tolerances (Namik, et al.).....	31
Figure 2.12 Japanese 10-15 mode drive cycle (NZ Government, 2009) .....	32
Figure 3.1 Recorded Brake Feedback from Normal Driving Conditions .....	35
Figure 3.2 Gas Pedal Dimension.....	36
Figure 3.3 Brake Pedal Dimensions.....	37
Figure 3.4 Distances Between Pedals (inches) .....	37
Figure 3.5 Gas Pedal Position During Regular Driving.....	40

---

Figure 3.6 Rate of Change of Gas Pedal .....	40
Figure 4.1 Preliminary Sketch 1 .....	46
Figure 4.2 Preliminary Sketch 2 .....	47
Figure 4.3 Preliminary Sketch 3 .....	48
Figure 4.4 First Prototype Model .....	53
Figure 4.5 Second Prototype Model .....	55
Figure 4.6 Final Design Model .....	57
Figure 4.7 Actuator Dimensioned Drawing .....	60
Figure 4.8 Robot 3 Compressed .....	60
Figure 4.9 Module Breakdown .....	63
Figure 4.10 dSPACE MicroAutoBox .....	64
Figure 4.11 dSPACE Computer Interface .....	65
Figure 4.12 NEO Fire VI <a href="http://intrepidcs.com">http://intrepidcs.com</a> .....	68
Figure 4.13 OBDII to db25 Pin Out .....	69
Figure 4.14 Wire Harness .....	72
Figure 4.15 1401/1501 Datasheet (dSPACE GmbH, 2009) .....	73
Figure 4.16 External Hardware - Motor Controller Schematic .....	82
Figure 4.17 VNH2SP30 High Current Motor Driver IC Schematic .....	83
Figure 4.18 Pololu Motor Controller Break-out Board ( <a href="http://www.robotshop.ca">www.robotshop.ca</a> ) .....	83
Figure 4.19 External Hardware - Safety Circuit Schematic .....	86
Figure 4.20 Current Calculations for Voltage Detection .....	88
Figure 4.21 Voltage Detection Schematic .....	89

---

Figure 4.22 Software Data Flow Breakdown.....	92
Figure 4.23 System_Testing.mdl .....	100
Figure 4.24 CAN Bus Signals.....	103
Figure 4.25 Encoder Feedback.....	105
Figure 4.26 Mode Selection .....	107
Figure 4.27 Source Selection .....	108
Figure 4.28 Source Control.....	110
Figure 4.29 Actuator Control Gas.....	111
Figure 4.30 Actuator Control Brake.....	113
Figure 4.31 Output Control Accel.....	114
Figure 4.32 Output Control Brake .....	116
Figure 4.33 Manual Control.....	117
Figure 4.34 Error Output.....	119
Figure 4.35 E-Stop .....	120
Figure 4.36 dSPACE to Accelerator .....	122
Figure 4.37 dSPACE to Brake .....	124
Figure 4.38 Pedal Position Controller.....	125
Figure 4.39 Speed Controller .....	127
Figure 4.40 Conflict Prevention.....	128
Figure 4.41 Encoder Position Decoder .....	130
Figure 4.42 Error to Motor.....	132
Figure 4.43 Discrete PI .....	133

---

Figure 4.44 Safety Limits .....	134
Figure 4.45 User Interface.....	135
Figure 5.1 Robotic Driving System.....	137
Figure 5.2 Mechanical Additions .....	139
Figure 5.3 Motor Controller Circuit V1 .....	140
Figure 5.4 Safety Circuit Bench Testing.....	143
Figure 5.5 Equipment Case Assembly .....	144
Figure 5.6 Motor with Encoder .....	146
Figure 5.7 FTP Schedule with Position Controller .....	147
Figure 5.8 User Interface.....	148
Figure 5.9 Testing Modes.....	148
Figure 5.10 Final Implementation Testing .....	150
Figure 5.11 Field Testing .....	151
Figure 7.1 L-Bracket Modification Drawing .....	159
Figure 7.2 Bottom Plate Drawing.....	160
Figure 7.3 Pusher Assembly Drawing.....	161
Figure 7.4 Standoff Drawing.....	162
Figure 7.5 Limit Switch Plate Drawing.....	163
Figure 7.6 Controller Board PCB Layout .....	164
Figure 7.7 Safety Circuit PCB Layout .....	165
Figure 7.8 Voltage Detect PCB Layout.....	166

---

## List of Tables

Table 2.1 Variable Description of Figure 2.7 .....	21
Table 2.2 Dynamic Drag Coefficients (Ehsani, 2005).....	22
Table 2.3 Rolling Resistant Coefficients (Ehsani, 2005).....	23
Table 2.4 Rolling Resistance Equation Variables.....	23
Table 2.5 Dynamic Drag Equation Variables .....	24
Table 2.6 Stahle SAP-2000 Actuator Specs.....	27
Table 2.7 Horiba ADS 7000 Actuator Spec.....	29
Table 3.1 Brake Pedal Force Measurements.....	36
Table 3.2 Summary of Mechanical Requirements.....	41
Table 3.3 Monitored Quantities .....	43
Table 3.4 Controlled Quantities .....	43
Table 4.1 Geared DC Motor Specs .....	50
Table 4.2: Mechanical Cost per Actuator .....	61
Table 4.3 dSPACE Connections to External Hardware – Accelerator Control.....	74
Table 4.4 dSPACE Connections to External Hardware – Brake Control.....	74
Table 4.5 dSPACE Connection to Vehicle Pin Description .....	75
Table 4.6 dSPACE Connection to Encoder .....	75
Table 4.7 dSPACE to Dynamometer Connections .....	76
Table 4.8 Electrical Cost Breakdown.....	90
Table 4.9 Required Display Information .....	95
Table 4.10 User Control Required .....	96

---

Table 4.11 System Testing.mdl Description .....	99
Table 4.12 CAN Bus Signal Module Guide/Design .....	102
Table 4.13 Encoder Feedback Module Guide/Design.....	104
Table 4.14: Mode Selection Module Guide/Design .....	106
Table 4.15 Source Selection Module Guide/Design .....	108
Table 4.16 Source Control Module Guide/Design .....	109
Table 4.17 Actuator Control Gas Module Guide/Design.....	111
Table 4.18 Actuator Control Brake Module Guide/Design .....	112
Table 4.19 Output Control Accel Module Guide/Design.....	114
Table 4.20 Output Control Brake Module Guide/Design .....	116
Table 4.21 Manual Control Module Guide/Design.....	117
Table 4.22 Error Output Module Guide/Design.....	118
Table 4.23 E-Stop Module Guide/Design .....	120
Table 4.24 dSPACE to Accelerator Module Guide/Design .....	121
Table 4.25 dSPACE to Brake Module Guide/Design .....	123
Table 4.26 Pedal Position Controller Module Guide/Design.....	125
Table 4.27 Speed Controller Module Guide/Design.....	126
Table 4.28 Conflict Prevention Module Guide/Design.....	128
Table 4.29 Position Decoder Module Guide/Design.....	129
Table 4.30 Error to Motor Module Guide/Design.....	131
Table 4.31 Discrete PI Module Guide/Deisgn .....	132
Table 4.32 Safety Limits Module Guide/Design.....	134

Table 5.1 Test 1 - Gas Position Controller..... 151

Table 5.2 Test 2 - Gas Position Controller..... 152

Table 5.3 Requirements Validation Matrix..... 154

Table 5.4 System Overview Validation Matrix ..... 154



# 1 Introduction

Car manufacturers are continuously trying to push the boundaries for fuel efficiency and performance from their vehicles. Motivated by the rising price of fuel in the last 10 years, companies are developing new approaches to reduce energy losses and decrease fuel consumption. Today, advancements in Hybrid vehicle technology are allowing motorists to get the most out of every drop of fuel. This is made possible by combining efficient electrical drive systems with techniques to reclaim and store energy. Hybrid vehicles continue to develop in a number of different ways including using lightweight materials, advanced electronic drive train controls and smoother running tires. These is a sophisticated power management system, to control the separate power sources efficiently. As limits on material selection and engine design are being approached, the power train control is now becoming a target for increasing efficiency. Current research at McMaster University involves analyzing a Hybrid vehicle's power train system, looking for improvements using repeatable and comparable lab tests.

## 1.1 Thesis Motivation

The Green Auto Power Train project involves numerous streams of research in various engineering departments across a number of Universities in Southern Ontario. Part of this research involves analyzing a Ford Hybrid vehicle in order to determine current efficiency and find methods to reduce overall energy waste in the power train system. This thesis introduces equipment for conducting in-lab testing, that is beneficial to researchers. The design consists of two mechanical-electrical actuators that control the

gas and brake in the test vehicle, controlled from customizable software. This thesis documents the development of the proposed testing equipment and provides details to further advance the software for future development. The final implementation is comparable to the functionality of commercial solutions, while also having the additional benefit of being on a more flexible platform potentially incorporating additional hardware modules for testing and data collection.

## 1.2 System Components

The design of an in-lab testing suite for vehicle control, involves different subsystems working together. This section will give a brief overview of the components involved.

### The Vehicle

A 2010 Ford Escape Hybrid has been made available to McMaster University for performing experiments on. This is a parallel hybrid vehicle using Ford's electronically-controlled continuous variable transaxle (e-CVT) transmission (see Section 2.1.2). These are 2 key features that must be considered while designing a robotic vehicle control system. The parallel hybrid design enables 2 power sources to simultaneously provide power to the wheels. The e-CVT transmission maintains an optimal rev per minute rate while accelerating and changing vehicle speeds. Traditional methods of using engine speed sensors as a feedback signal to determine vehicle speed will not work. More information on Hybrid vehicle design can be found in Section 2.1.

We are able to record data from the car's computer using the On-board Diagnostic (OBD) connection located beneath the dashboard on the driver's side. The dimensions of the interior, around the driver's pedals, were collected for design size requirements.



**Figure 1.1 2010 Ford Escape Hybrid**

## Dynamometer Setup

At McMaster's Innovation Park, there is a Chassis Dynamometer set up for full-speed in-lab tests with the Ford Hybrid Vehicle. The Dynamometer was originally for Ontario Drive Clean Emissions Tests. It's comprised of a MD-100 dyno from Mustang Dynamometer connected to a computer terminal running the emission analysis software. The Dyno, is used to simulate road friction while in a controlled testing environment. For initial tests, the Dyno is used in a free run mode. To use the Dyno as a controllable component in this testing suite, research and testing to become familiar with the dyno still

needs to be done. The availability of this equipment has been unpredictable, as new facilities are being constructed for the Green Auto research. The availability has had an impact on the progress made to develop this component.

## Robotic Driver

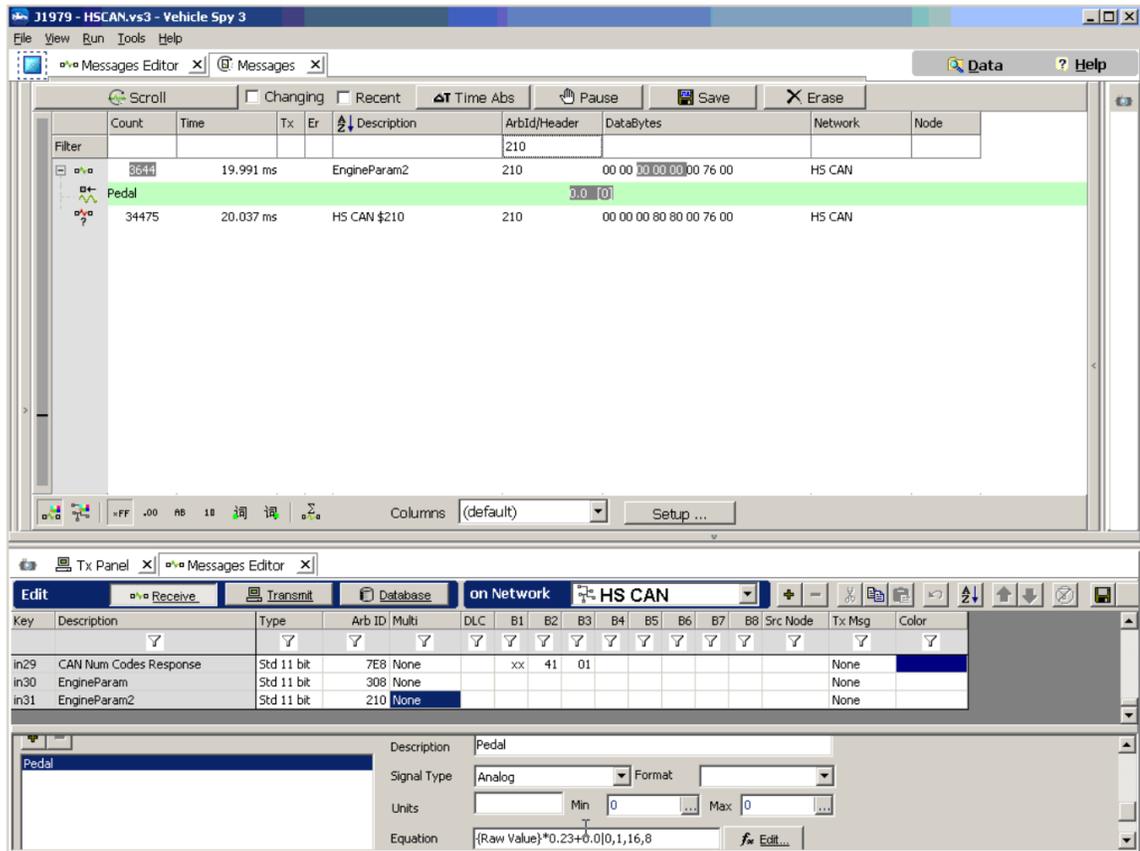
The main focus of this thesis is the mechanical-electrical device with feedback control from the car's computer. The requirements, system design and final verification sections document the design of a driving robot to control the 2010 Ford Escape Hybrid.

## Closed-Loop Control System

A closed-loop control system is used to command the actuators to follow specified driving schedules, which may include standard industry tests or custom user recorded driving cycles. Bench testing is conducted using position feedback sensors that simulate pedal position information from the car. After there's confidence with the controller functionality, full scale testing, using the chassis dyno, verifies the final design.

## Vehicle Interfacing

Making a connection to the car is done using a device capable of receiving CAN Bus messages. The NEO VI Fire from Intrepid Control Systems Inc. is capable of sending and receiving CAN Bus messages through a USB connection and displayed using the software suite Vehicle Spy 3, seen in Figure 1.2. Becoming familiar with the car's various control computers helps understand how a vehicle's control is implemented.



**Figure 1.2 Vehicle Spy 3 - <http://intrepidcs.com/VehicleSpy/index.html>**

The main control hardware being used is the dSPACE MicroAutoBox 1401/1501. This box is capable of receiving CAN messages as well as a variety of other digital and analog inputs. Control code can be developed in Matlab/Simulink, python or C.



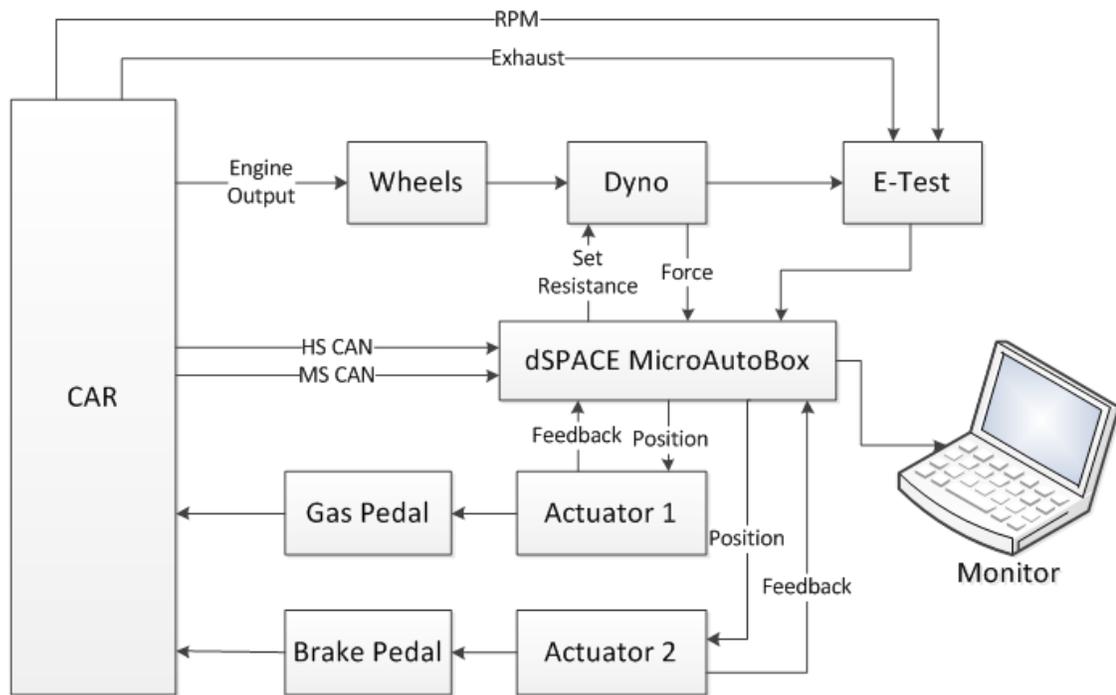
**Figure 1.3 dSPACE MicroAutoBox 1401/1501**

## Driving Cycle Collection

Collected driving cycles are used to accurately record a human driver's control of the car and replay these actions using the robotic driver. This collection is done by recording vehicle speed vs. time feedback from the OBDII port to a memory device. For the purposes of this thesis, the dSPACE Micro Auto Box is used to collect these cycles

## System Overview

Figure 1.4 provides a general system block diagram to help explain the proposed thesis. The dSPACE is the central controller, communicating with the vehicle, actuators, dyno and operator monitor. The feedbacks from the actuators provide information on position, current draw and limit switch detection to ensure operation limitations are not being exceeded.



**Figure 1.4 System Overview**

### 1.3 Thesis Layout

This document goes through the design stages involved for implementing an electrical-mechanical actuator with an embedded real-time controller. Topics are broken up into Mechanical, Electrical and Control when possible. The reader will understand the design motivations as well as have an understanding of functionality for the final product. This document is a stepping-stone for future research and development of the in-lab testing suite.

The Background information section provides information to support ideas in later chapters. This covers topics related to project specific technologies, existing research and products developed which inspired this thesis project.

The Requirements chapter is a summary of specific functional requirements required to achieve a complete system. This outlines criteria for the system design chapter and provides a basis for System Validation in chapter 5.

The System Design chapter explains design decisions and overall functionality of the final product. Design decisions are related to stated requirements whenever possible.

The System Validation chapter summarizes in-lab testing for the final system implementation. Tests are conducted in an incremental method to test and debug the hardware and software. Using a spiral method, the final design is continually updated and tested.

The Conclusion chapter summarizes the accomplishments achieved during this thesis. Future Work will give ideas for further research, for which there was not enough time to complete during this research term.

## 2 Background

This section provides background information to support this thesis. The GreenAuto Power Train project involves exploring the capabilities of the Ford Escape Hybrid and its many sub systems. This thesis follows the development of a testing suite to control the accelerator and brake in order to assist in conducting repeatable tests with the provided vehicle. The testing suite reduces the need for lengthy and potentially dangerous road tests which cost researchers valuable time away from the office. The intention is to simulate these real-world driving scenarios with repeatability, for full-scale system testing. The following sections in this chapter will introduce the Ford Escape Hybrid technology; the available communication systems within the vehicle; the Dynamometer being used for in lab testing; a description of resistive forces present on the vehicle while driving; commercially available systems available for in-lab vehicle testing; and finally current test cycles for in-lab testing in industry.

### 2.1 Hybrid Vehicles

This section gives an overview of Hybrid vehicle technology, specifically for the 2010 Ford Escape hybrid. This vehicle is a gasoline-electric hybrid with a parallel hybrid design, meaning that both the electric and gas powered engines work together to propel the car. At low speeds and from takeoff, only the electric motor is engaged when possible. The electric motor power is ideal for urban driving as it has enough power to get the vehicle moving and maintaining speeds under 60 km/h. The gasoline engine is intended to provide additional power to the drive wheels when required and charge the battery.

Additional power is reclaimed from regenerative braking and descending a hill. Regenerative braking works by using the back electromotive force (EMF) from the electric motor to provide resistance to the rotation of tires and reclaim energy (Ford Motor Company, 2009). The hydraulic brakes will engage when extra braking force is required. The gasoline engine is also used to heat and cool the cabin and battery compartment. Maintaining ideal battery temperatures helps to extend the battery life. The gasoline engine is run in free run state (not consuming gasoline) when descending a hill to help control the vehicles speed and produce electric energy as a generator (Ford Motor Company, 2009). These features are important to consider when designing an automated vehicle control system for a hybrid vehicle. Methods, such as monitoring the engine's speed, to determine the vehicle speed cannot be used because of the e-CVT and electric parallel power input. The following sections give further information into how these systems work.

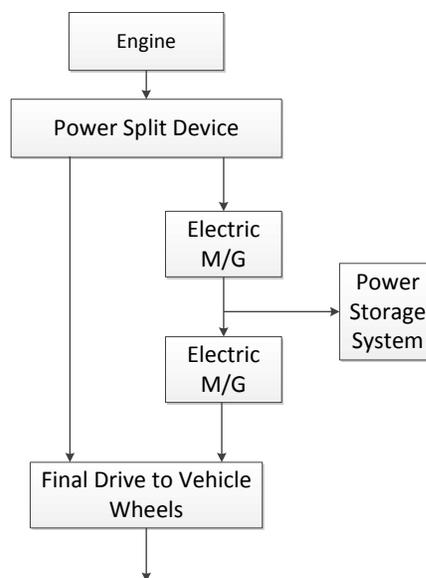
### 2.1.1 Serial and Parallel Hybrid Vehicles

There are 2 different types of hybrid vehicle designs on the market today. The configurations are called Parallel and Serial Hybrid. This section will focus specifically on Gasoline-Electric Hybrid vehicles.

In a serial hybrid vehicle, the car is powered by the electric motor. The gasoline engine is used to generate electricity to recharge the batteries. The benefit of this is that a smaller gasoline engine is needed to power the batteries meaning less energy is lost due to excessive weight. Electric motors also run more efficiently than combustion engines

because of fewer moving parts and less heat loss (Ehsani, 2005). These vehicles also do not require a sophisticated transmission for combining drive power between propulsion sources.

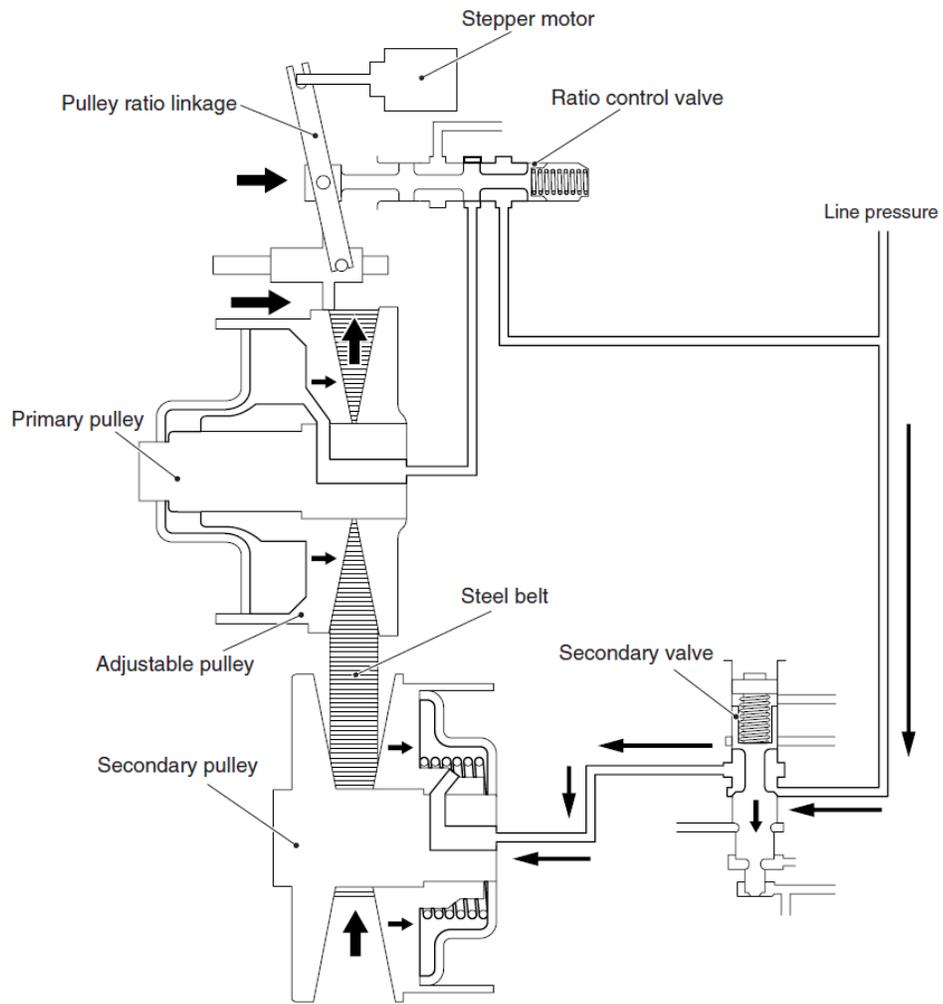
The parallel hybrid design has the ability to use both electric and gas motors to provide propulsion. This means that the gasoline engine can assist the electric motor when more power is needed. Since the gasoline engine is only providing a portion of the power, it does not have to be big enough to fully power the vehicle, which reduces overall weight. Parallel vehicles have more sophisticated transmissions than serial because they need to be able to engage and disengage the gasoline engine at any time (Ehsani, 2005). Figure 2.1 shows a top level functionality of the parallel hybrid transmission configuration. There are two electric motors/generators used to produce power and reclaim energy; one connected to the drive wheels and one connected to the gas engine.



**Figure 2.1 Parallel powertrain used in Ford Hybrids (Miller, 2006)**

### 2.1.2 E-CVT

Parallel hybrids require a sophisticated transmission in order to efficiently transfer power to the drive wheels. The Ford Hybrid uses an electrically-controlled continuous variable transaxle (e-CVT) transmission. This transmission allows for a smooth transition between gear ratios. Figure 2.2 illustrates the inner workings of a continuous variable transmission. Figure 2.3 illustrates the top level view of a parallel hybrid's split e-CVT. The gaps of the two pulleys are adjusted inward and outward to change the gearing ratio between the engine and the drive wheels. The two sides of the pulleys squeeze a metal belt which causes it to rise on the V-shaped groove. These transmissions are typically used in large machinery, such as farming equipment. Implementations commonly use hydraulically controlled CVT's which are heavy and require hydraulic reserves. The Ford Escape Hybrid uses an electrically actuated system that reduces overall weight.



**Figure 2.2 Continuous Variable Transmission Representation (Mitsubishi, 2010)**

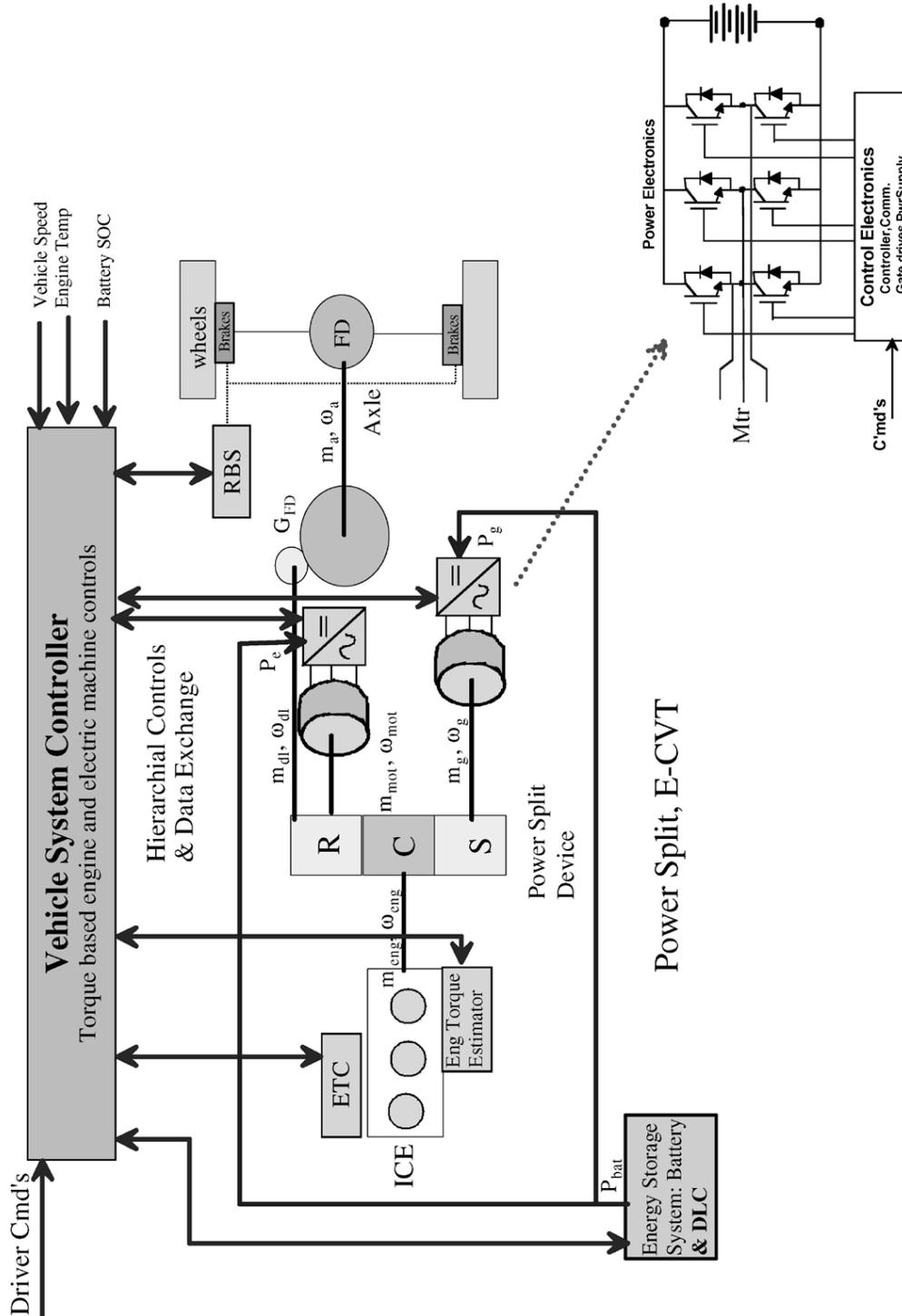


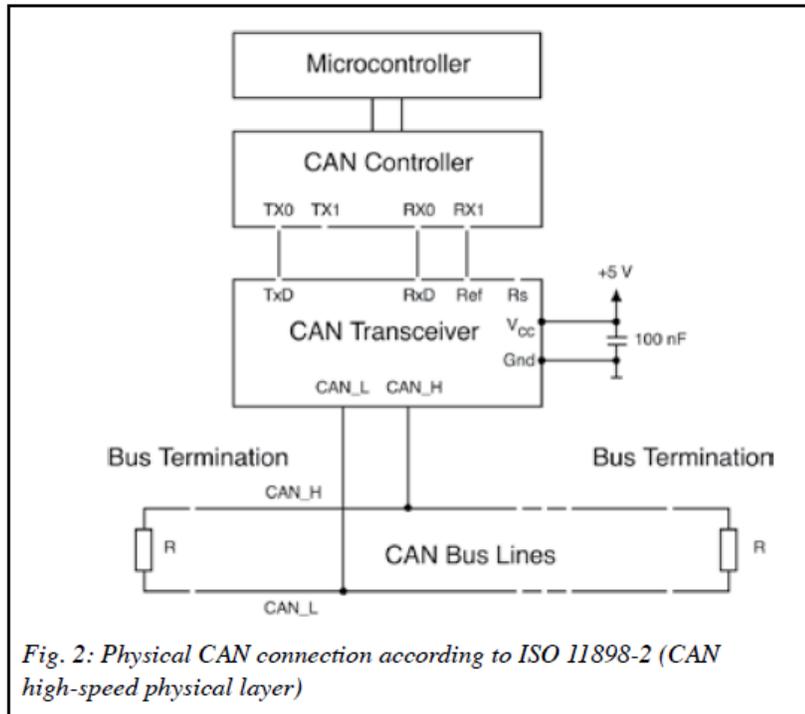
Figure 2.3 Power split e-CVT of the input split type (Miller, 2006)

## 2.2 CAN Bus in Automobiles

Automobiles today have an increasing reliability on Electronic Control Units (ECU) to control functionality of the vehicle. The ECU's are dependent on one another for information in order to make decisions. It is necessary to have a secure communication network in place, designed to be reliable in harsh conditions. Requirements include dealing with environmental noise, message transmission speed, message priority, collision detection, and simplicity. In order to achieve the hard real-time system needed, car manufacturers use CAN (Controller Area Network) Bus communication protocol to ensure reliable data transmission.

The CAN Bus protocol was developed by Robert Bosch GmbH for use in automobiles as an alternative to more expensive wiring harnesses. (Vahid, et al., 2002) Today it is used for a variety of industrial communication applications which require secure serial communication in a noisy environment. Examples include elevator controls, production line equipment and medical instruments.

The physical layer is defined by ISO 11898 for high speed applications and ISO 11519-2 for low speed applications. (Vahid, et al., 2002) The layout within vehicles uses a differential electrical bus connecting the nodes together. The messages are in a standard format that each node can receive and interpret. Figure 2.4 shows a can connection, defined by ISO 11898-2.



**Figure 2.4 CAN Physical Layer ISO 11898-2 (CiA, 2011)**

Figure 2.5 illustrates the structure of a CAN base frame.



**Figure 2.5 CAN Bus Frame**

The frame begins with the start bit “Start of frame (SOF)”, followed by the arbitration field and the remote transmission request bit (11 + 1 bits). CAN 2.0 uses an extended arbitration identifier using an additional 18 bits, (29 bits total). (CiA, 2011) The data field length is specified in the control field, up to 8 bytes. (CiA, 2011) Transmission

errors are checked using the Cyclic Redundancy Checksum (CRC) field. ACK is sent as a recessive acknowledgment bit and overwritten as a dominant bit by the module(s) receiving the information. Finally EOF indicates the end of frame which consists of a 7 recessive bit intermission. Another sender can begin after the EOF is completed. When there is nothing being sent, the line remains in a recessive state.

The CAN Bus Protocol uses unique identifiers to control priorities of messages being transmitted over the bus. Within the vehicle, messages that are more important, such as engine speed, should have priority over messages like engine temperature. When two messages begin to transmission, the first thing sent is an arbitration identifier. Logic states are defined as dominant and recessive. The CAN protocol defines the bit “0” as dominant and “1” as recessive. The recessive node will see logic 0 when it expects to see logic 1 on the line and stop transmission. The message with a higher arbitration value will yield to a message with a lower value. This is known as non-destructive bitwise arbitration and guarantees that neither information nor time is lost.

The High Speed CAN line within new vehicles employs the CAN 2.0 scheme, which can transmit at speeds up to 1Mbps, 8 times faster than CAN 1.0. (CiA, 2011) A Ford Engineer at the Oakville Assembly plant provided the transmission speeds when the NEO VI Fire vehicle monitoring hardware was setup. The High Speed CAN bus has a transmit speed of 500kbps. Medium Speed CAN lines have a transmit speed of 125kbps. A database descriptor file is used to interpret the 8 bytes of data being transmitted between nodes. The descriptor file provides information to break up messages into

multiple signals, based on the identifier; it then scales the data and provides units. Each node on the bus needs this file to understand one another. The files are proprietary information to auto manufacturers and had to be provided by Ford under a lengthy non-disclosure agreement.

## 2.3 Dynamometer

A dynamometer is used as a measurement tool to capture force and torque outputs of a rotational power system. Specifically, an automotive chassis Dynamometer, like the one pictured in Figure 2.6, is used to measure the transferred force from the vehicles drive train, through the tires and on to a set of rollers (Environmental Systems Products Holding Inc., 2002). The rollers are connected to an electric generator to capture a power output. The rolling resistance can be adjusted by changing the resistive load on the generator. By changing this electrical resistance, more power is required to obtain the same rate of rotation through the rollers. Driving simulations can be performed in the lab using a controller to continuously change the resistance in order to recreate forces that the car would experience in different driving scenarios. Some test scenarios could be ascending and descending of a hill, simulating wind resistances, and comparing gravel surfaces to asphalt.



**Figure 2.6 Dynamometer Setup at McMaster Innovation Park**

## 2.4 Combined Resistive Forces while Driving

For replaying driving scenarios, an overall net resistive force on the vehicle can be established using external sensors. This force can then be applied to the vehicle using a controllable chassis dynamometer during in-lab tests. In order to calculate resistances experienced by a vehicle, it is important to understand the monitored variables required in order to get an accurate prediction of the resistive force. The following background discussion provides necessary information for dynamic testing with a dynamometer, to be used for future development of the testing suite. Figure 2.7 illustrates the forces acting on a vehicle.

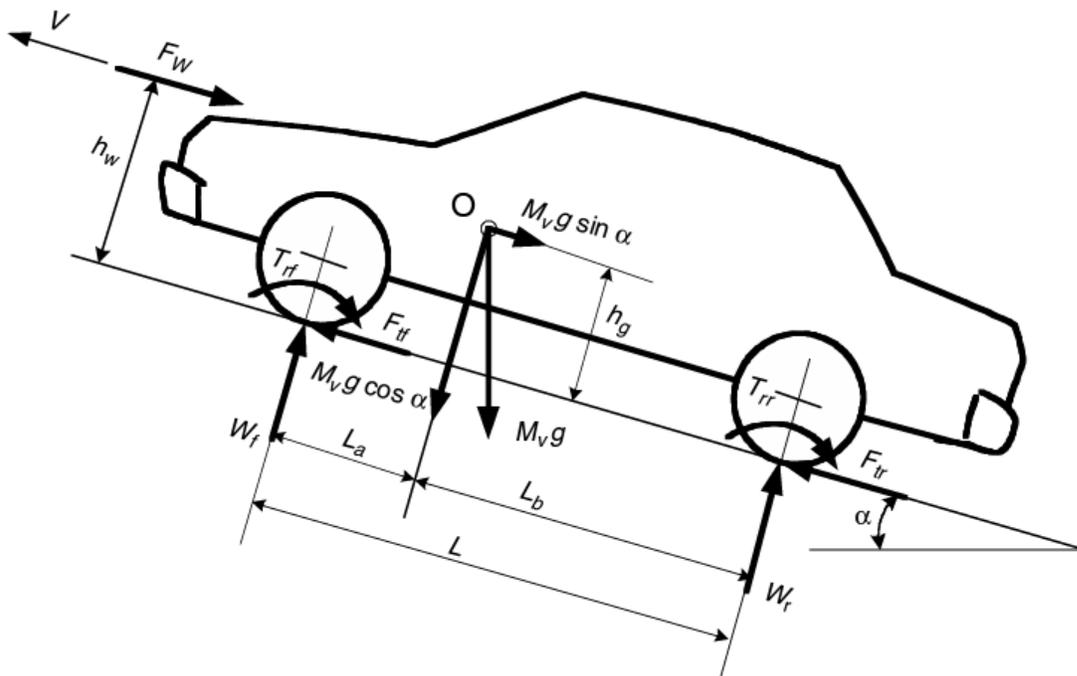


Figure 2.7 Forces acting on a vehicle (Ehsani, 2005)

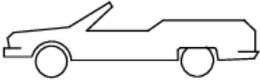
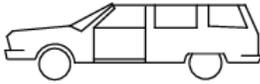
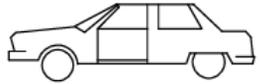
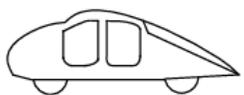
Variable	Unit	Description	Type
V	m/s	Vehicle speed	Monitored
$F_w$	N	Aerodynamic drag	Calculated
$h_w$	m	Height of effective drag force (approx. $h_g$ )	Constant
$T_{rf}$	N-m	Front tire rolling resistance moment	calculated
$T_{rr}$	N-m	Rear tire rolling resistance moment	calculated
$F_{tf}$	N	Tractive force of front drive wheels	calculated
$F_{tr}$	N	Tractive force of rear drive wheels	calculated
$M_v$	kg	Total mass of the vehicle	Constant
$\alpha$	degrees	Angle of grade	Monitored
$h_g$	m	Height of center of mass	Constant
$W_f$	N	Normal load of front axle	Calculated
$W_r$	N	Normal load of rear axle	Calculated
L	m	Distance between wheels	Constant
$L_a$	m	Distance to center of mass from front tire	Constant
$L_b$	m	Distance to center of mass from rear tire	Constant

**Table 2.1 Variable Description of Figure 2.7**

The effective resistive force can be calculated by monitoring the angle of inclination of the vehicle and the current speed. Other factors that affect the resistance of the vehicle are the dynamic drag (2.4.1) and the rolling resistance (2.4.2). These calculations depend on weather conditions and the road surface during data collection. The following sections explore the monitored variables needed to get a complete approximation of the vehicles overall resistive force, followed by the equations needed in section 2.4.3.

### 2.4.1 Dynamic Drag

Dynamic Drag Force is a measure of how the vehicle travels through the environment. The main contributing factors are the vehicle's shape and the medium the vehicle is travelling in. Table 2.2 shows ranges of drag coefficients associated with different shapes of vehicles.

Vehicle type	Coefficient of aerodynamic resistance
 Open convertible	0.5...0.7
 Van body	0.5...0.7
 Ponton body	0.4...0.55
 Wedged-shaped body; headlamps and bumpers are integrated into the body, covered underbody, optimized cooling air flow	0.3...0.4
 Headlamp and all wheels in body, covered underbody	0.2...0.25
 K-shaped (small breakaway section)	0.23
 Optimum streamlined design	0.15...0.20
Trucks, road trains	0.8...1.5
Buses	0.6...0.7
Streamlined buses	0.3...0.4
Motorcycles	0.6...0.7

**Table 2.2 Dynamic Drag Coefficients (Ehsani, 2005)**

### 2.4.2 Rolling Resistance

Rolling resistance of the tires will change depending on roadway conditions. This is a result of imperfections in different road surfaces as well as the physical deformity of

the tires on the road, which causes losses in energy (Ehsani, 2005). The resistance can be calculated using rolling resistant coefficients seen in Table 2.3. The values in Table 2.3 do not take into account for variations in speed.

### Rolling Resistance Coefficients

Conditions	Rolling resistance coefficient
Car tires on concrete or asphalt	0.013
Car tires on rolled gravel	0.02
Tar macadam	0.025
Unpaved road	0.05
Field	0.1–0.35
Truck tires on concrete or asphalt	0.006–0.01
Wheels on rail	0.001–0.002

Table 2.3 Rolling Resistant Coefficients (Ehsani, 2005)

### 2.4.3 Equations

The equation for rolling resistance is:

$$F_r = M_v g f_r \cos \alpha$$

Variable	Unit	Description	Type
$F_r$	N	Rolling resistance	calculated
$M_v$	kg	Total mass of the vehicle	Constant
$\alpha$	degrees	Angle of grade	Monitored
$f_r$	-	Rolling resistance coefficient (Table 2.3)	Constant
$g$	$m/s^2$	Gravity ( $9.81m/s^2$ )	Constant

Table 2.4 Rolling Resistance Equation Variables

Aerodynamic drag uses the following equation to calculate the effective force (Ehsani, 2005):

$$F_w = \frac{1}{2} \rho A_f C_D (V + V_w)^2$$

Variable	Unit	Description	Type
$F_w$	N	Dynamic drag force	Calculated
$\rho$	kg/m <sup>3</sup>	Air density	Monitored
$A_f$	m <sup>2</sup>	Frontal area of vehicle	Constant
$C_D$	-	Aerodynamic drag coefficient (Table 2.2)	Constant
$V$	m/s	Vehicle speed	Monitored
$V_w$	m/s	Wind speed	Monitored

**Table 2.5 Dynamic Drag Equation Variables**

The gravitational force which will affect the vehicles resistive force when ascending or descending a hill can be expressed as:

$$F_g = M_v g \sin \alpha$$

The equation for the effective resistive force on a vehicle is:

$$\sum F = F_r + F_w + F_g$$

$$F_{eff} = M_v g (f_r \cos \alpha + \sin \alpha) + \frac{1}{2} \rho A_f C_D (V + V_w)^2$$

Variables to monitor are air density, vehicle speed, wind speed, and angle of inclination. Other variables are constants that are specified for each driving scenario being tested.

## 2.5 Commercial Driving Robots

The development of an autonomous driving simulator is not a new concept. There have been attempts as early as 1974 to develop robotic driving controls for dynamometer testing (Namik, et al.). Today there are commercially available driving robots developed for emissions tests, fuel consumption analysis and emission endurance testing. They can also be used for testing vehicles in extreme temperatures or mileage accumulation. This section will compare two commercially available units which are capable of operating different vehicles, including manual transmission cars. For the purposes of this research with hybrid vehicles, the transmissions are automatic, simplifying design requirements.

### 2.5.1 Stahle SAP-2000



Figure 2.8 Stahle SAP-2000

The SAP-2000 is a commercial driving robot designed to be a human analog for chassis dynamometer testing. Its purpose is to provide repeatable vehicle control for vehicle testing and verification. The following applications are listed on the Stahle website:

- Research and development
- Climate testing
- Mileage accumulation
- Endurance testing
- Acoustics testing
- Transmission testing
- Powertrain testing
- Engine calibration

This system is adjustable and works with most vehicles, including having add-ons to work with manual transmission vehicles. The manufacturer claims it can be installed within 15 minutes. The control software is run on a standard computer and is linked to the SAP-2000 via a RS-232 or LAN connection. (Stahle, 2010) The software will play back recorded driving cycles to the robot. Other features listed are the ability to connect to an exhaust analyzer, dynamometer control, data acquisition, and control from a remote computer. This system does not interface with the CAN Bus communication of the vehicle. It is able to collect feedback signals for vehicle speed and engine speed using other methods.

The SAP2000 system is able to calibrate its controls to the vehicle it is operating before simulation begins. This is done using a “Learn Cycle” that relates the engine speed (RPM) to the throttle position. This information is then used to improve simulation results. (Stahle, 2010)

The system is placed on the driver's seat and secured using ratchet straps. Telescopic actuators extend from the main body of the system to the pedals. The actuators are then physically attached to the pedals to ensure they do not slip off. There are no tools required for the installation. The main body of the system weighs 30 kg. (Stahle, 2010)

For autonomous vehicle control, safety is very important. The SAP-2000 system has incorporated 3 different safety features. The emergency off disables the main power, which results in all pedal actuators retracting. This also has the same result as an external power failure. There is also an emergency stop, which requires power, and will shift a manual vehicle in to neutral and compress the brake pedal. The control software is equipped with a watchdog monitoring system. (Stahle, 2010) The benefit of using watchdog timers in software design is that it guards against unexpected behavior by ensuring processes are completed on time. If the software became locked-up, it would trigger an emergency stop.

The following specs in Table 2.6 for the accelerator and brake actuators can be used as benchmarks for future system designs.

	Accelerator	Brake
Max Stroke	150mm	150 mm
Max Force	100 N	350 N
Max Speed	0.45 m/s	0.3 m/s

**Table 2.6 Stahle SAP-2000 Actuator Specs**

## 2.5.2 Horiba ADS 7000



**Figure 2.9 Horiba ADS 7000 (Horiba, 2010)**

The Horiba ADS 7000 is a commercial driving robot used primarily for testing new vehicles before they go on the market. (Horiba, 2010) Applications include emission, fuel consumption rating and endurance testing. Using this system, a vehicle can be evaluated and compared to other vehicles of similar class to determine an overall rating.

The ADS 7000 has been designed for easy installation to limit time between testing vehicles. It sits in the driver's seat and operates the pedals at a similar angle as a human driver. As seen in Figure 2.9, there are independently controlled actuators for each pedal. Its estimated installation time is four minutes plus a three minute automatic adjusting period. It weighs 30 kg all together and separates into parts weighing 10 kg each. (Horiba, 2010)

This is an electrically operated actuator controlling a vehicle which does not depend on an external power source. Safety measures are put in place to guard against power failure and unexpected operation. The system is equipped with an alarm and emergency stop. When activated, the accelerator actuator is released, the vehicle is shifted into neutral, and the ignition key is turned off. (Horiba, 2010) If there is a power failure, the throttle is completely released. (Horiba, 2010)

The control method for this system uses a feed-forward proportional-integral (PI) controller. (Horiba, 2010) The system can play back either vehicle speed versus time data or stroke control using recorded pedal positions. Standard driving cycles come included with the system; EPA, ECE and Japanese. This system is also capable of replaying custom driving cycles up to 8000 seconds long. When the system is initialized, it finds the starting position of the gas, brake and clutch pedals. After this is found, pedal position feedback can be monitored with internal position sensors. This eliminates the need to communicate with the vehicles computer for feedback signals. (Horiba, 2010)

The following characteristics in Table 2.7 are used as a reference for the driving simulator requirements in Chapter 3.

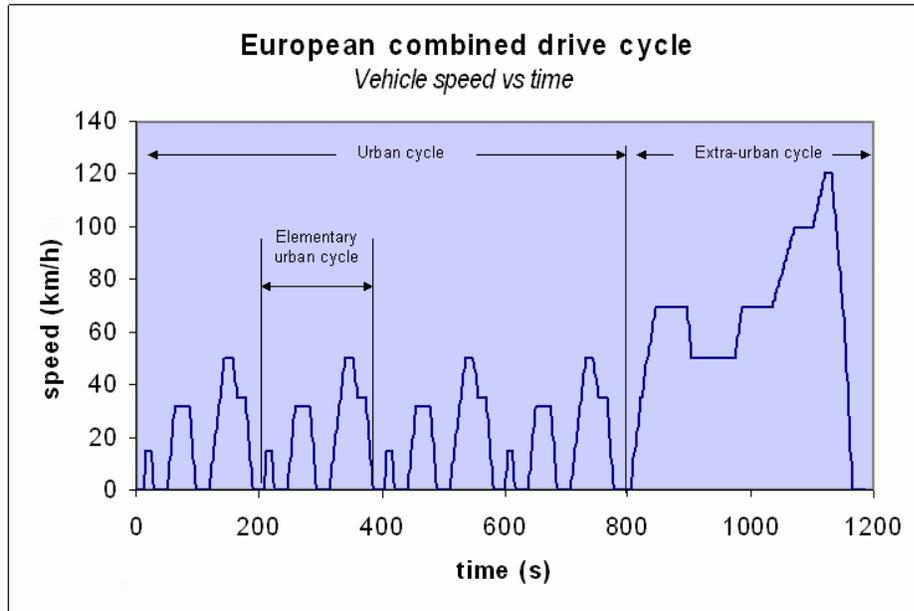
	Accelerator	Brake
Max Stroke	200 mm	200 mm
Max Force	160 N	220 N
Max Speed	N/A	N/A

**Table 2.7 Horiba ADS 7000 Actuator Spec**

## 2.6 Driving Cycles

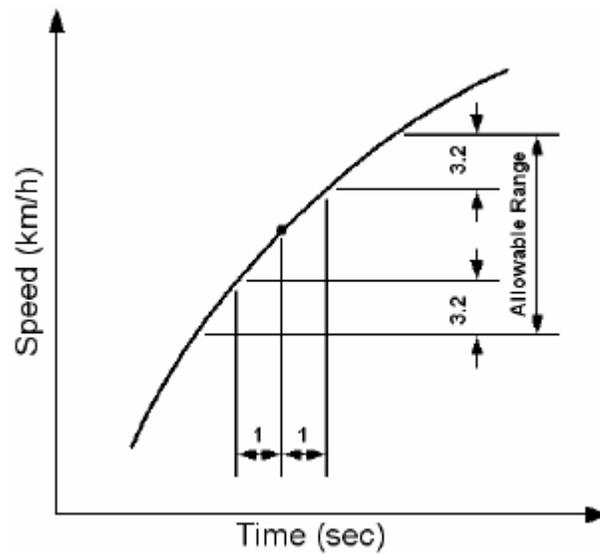
To test a car's efficiency, there are standard driving cycles which cars are put through. Their Fuel consumption can then be compared to other makes and models of similar classes. When dealerships advertise Highway and Urban fuel consumption, the figures produced come from these standard driving cycles. These cycles are a representation of a vehicle's Speed Vs. Time. The European Combined Drive Cycle (ECE), shown in

Figure 2.10, is used to verify that vehicles comply with the Australian Design Rules (ADR) (NZ Government, 2009). For a successful test, the vehicle must remain within a certain tolerance of the driving cycle input. This is illustrated in Figure 2.11. The vehicles speed must lie within  $\pm 3.2$ km/h and the settling time of the controller must reach the desired speed within 1 second. This tolerance can be out of range but for no more than a total of two seconds for the test (Namik, et al.).



*Drive cycle contained in UNECE Regulation 101*

**Figure 2.10 European Combined Drive Cycle (NZ Government, 2009)**



**Figure 2.11 ADR drive cycle speed tolerances (Namik, et al.)**

Another example of a standard driving cycle, illustrated in Figure 2.12, used in Japan for controlled benchmark testing. Custom driving cycles can be recorded for simulation of different driving scenarios in a controlled environment.

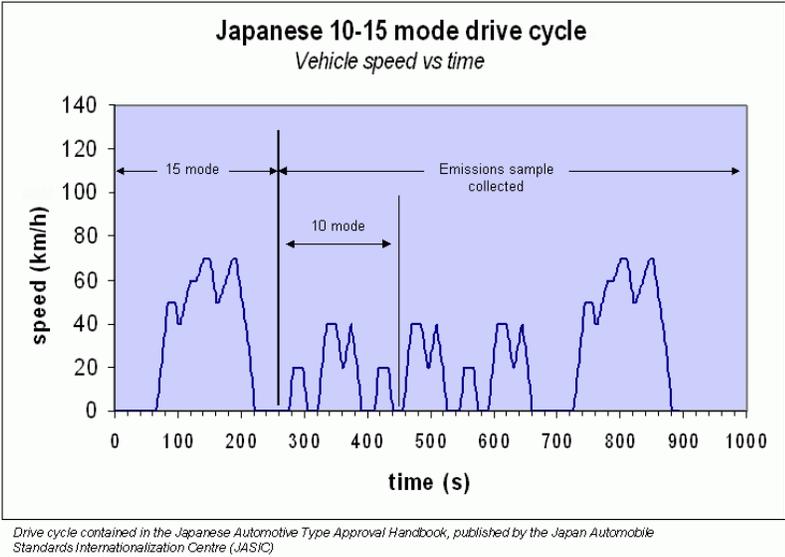


Figure 2.12 Japanese 10-15 mode drive cycle (NZ Government, 2009)

## 3 Requirements

This section will detail the requirements needed for the mechanical, electrical, control and safety design for the development of a robotic vehicle control system. The requirements and design are based on assumptions made in Section 3.1. The requirements specification is broken down into mechanical, electrical, control and safety to better outline each requirement. This information is used for design and validation in Chapters 4 & 5.

### 3.1 Assumptions

The following is a list of assumptions that the design of the automated robotic driving control system will be based on.

1. This device is to be used on a 2010 Ford Escape Hybrid with automatic transmission.
2. The actuator control uses the dSPACE MicroAutoBox, which has been provided for this project.
3. The design is for performing in-lab driving simulations limited to everyday scenarios. It does not have to perform fast accelerations or breaking of the vehicle.
4. The final design assumes constant user supervision of operation by someone who can identify unsafe, out-of-range behavior.
5. Equipment will operate mainly in a climate-controlled environment.
6. External AC electrical power will be available for in-lab tests

## 3.2 Mechanical Requirements and Constraints

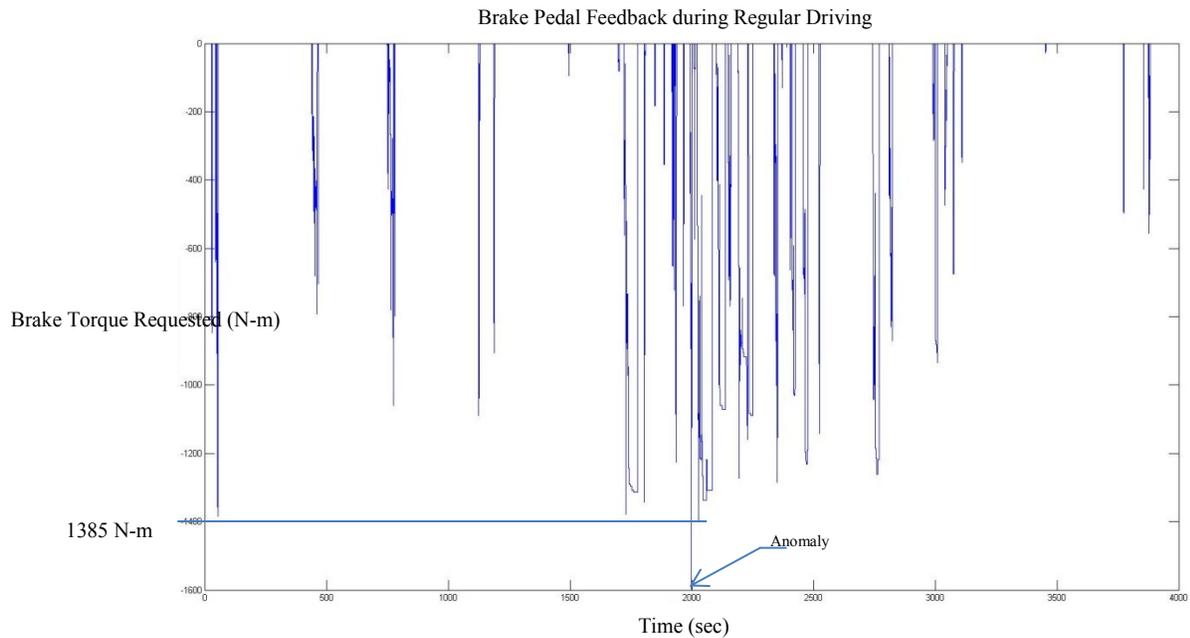
The mechanical requirement section will be limited to the physical system being implemented. The following headings detail the characteristics critical to the design.

### 3.2.1 Force Output for Gas

The force output from the actuator must exceed the resistance force of the vehicles accelerator pedal. Because the magnitude of force required is difficult to predetermine, this requirement shall be verified through testing.

### 3.2.2 Force Output for Brake

A rough estimation for the exerted force on the brake pedal can be made using a bathroom scale. This scale is calibrated for a normal force under earth's gravity to give a mass (in lbs. or kilograms) reading of a human. This can be used to calculate a linear force in newtons by multiplying a kilogram reading by  $9.81 \text{ m/s}^2$ .



**Figure 3.1 Recorded Brake Feedback from Normal Driving Conditions**

Figure 3.1 shows a recorded brake feedback over a typical driving period. The value recorded is a requested braking torque to the wheels measured in Newton-meters. This value is directly proportional to the position of the pedal. The brake pedal force feedback is also proportional to the requested breaking torque value. The full braking power is not used on a typical drive. The actuator being designed is required to compress the brake pedal to a point within normal operation boundaries. There are limits to the amount of braking force used during testing in order to prevent dangerous operation of the vehicle, such as locking the wheels or engaging ABS while on the dynamometer. The maximum recorded value is -1385 N-m (negative sign is for convention). The following measurements demonstrate the relationship to the brake feedback and the force required to compress the brake.

Requested Torque (N-m)	Measurement (Kg)	Force (Newtons)
-1385	2.0	19.62
-2750	5	49.05
-3600	10	98.1

Table 3.1 Brake Pedal Force Measurements

The required force output of the actuator should exceed 20 Newtons.

### 3.2.3 Size Constraints

The actuator size constraints directly come from measurement of the driver side floor area. The following diagrams are rough measurements taken with a carpenter’s tape measure to approximate the size restrictions. Dimensions are subject to a maximum tolerance of ±0.125 inches (±0.3175 cm)

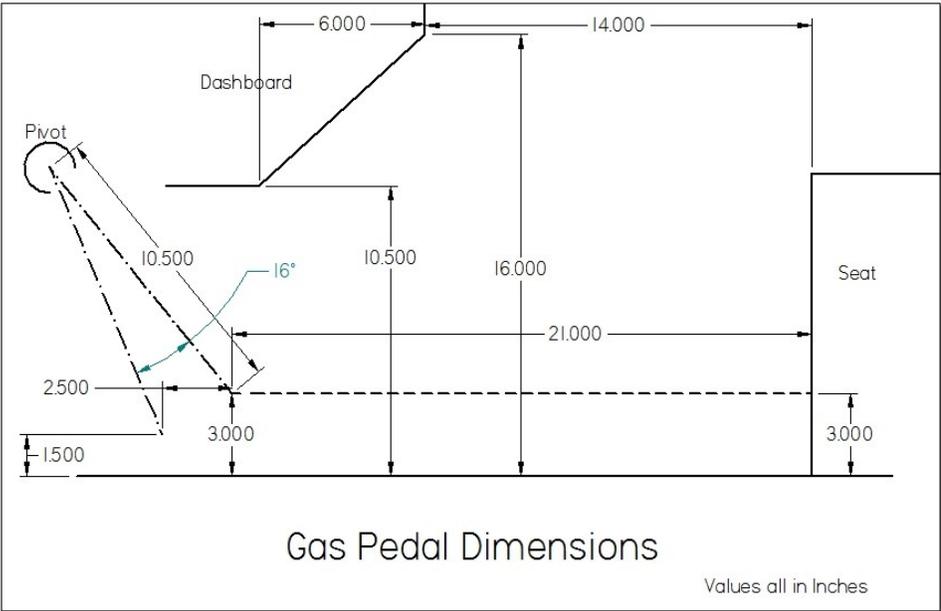


Figure 3.2 Gas Pedal Dimension



### 3.2.3.1 Length Constraint

As seen in Figure 3.3, the maximum length of a typical implementation is constrained by the brake pedal's distance to the driver's seat. This distance is measured at **20 inches (50.8 cm)**. This distance is taking into account the shortest distance between driver's seat and brake pedal when seat is moved back fully. It shall be noted that there is extra available room for a lower profile design that sits beneath the seat cushion.

### 3.2.3.2 Width Constraint

In Figure 3.4, the measurements of the distance between pedals and also the distance from the gas pedal to the center console restrict the width of each actuator. The actuator width should be no greater than **5 inches (12.7cm)**. This will allow sufficient contact with each pedal and also leave enough room to clear the center console.

### 3.2.3.3 Height Constraint

The height of the design is restricted by the dashboard. This can be seen in Figure 3.2 and Figure 3.3. The design may contact the dashboard for stability, but should not leave any marks or scratches. Designs will stay within the **10.5 inch (26.67cm)** distance from the floor to the dashboard.

### 3.2.3.4 Contact Tolerances

The actuator contact with the pedal needs to have a tolerance built in to incorporate the change in vertical height as the pedal is compressed. Figure 3.2 shows a **1.5 inch (3.81 cm)** change in height of the gas pedal and Figure 3.3 shows a **0.5 inch (1.27 cm)** change in height of the brake pedal. The actuator should remain in contact with

the pedal surface at all times during compression and should limit the vertical motion's effect while compressing and decompressing the pedals.

### 3.2.4 Actuation Distance for Gas Pedal

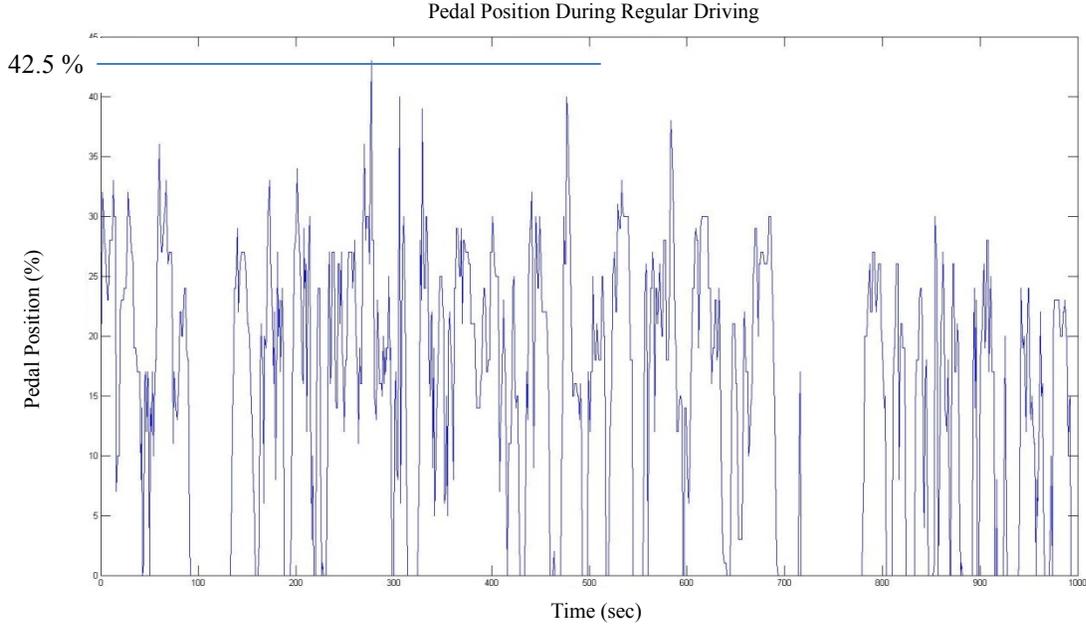
Figure 3.2 illustrates a rough linear approximation for the actuation distance of the gas pedal. The estimated linear actuation distance is **2.5 inches (6.350 cm)**. The design should allow a tolerance to account for initial calibration of the robot to find a zero position as well as a margin of error on how the actuator is placed within the vehicle.

### 3.2.5 Actuation Distance for Brake Pedal

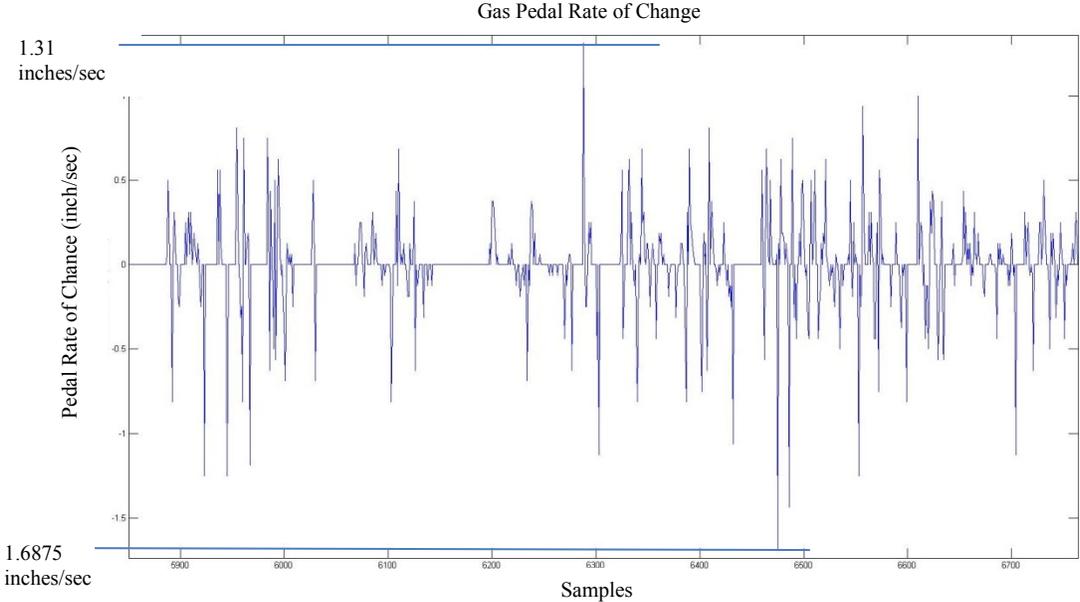
Figure 3.3 shows a rough linear approximation for the actuation distance of the brake pedal. The estimated linear actuation distance is **3.5 inches (8.89 cm)**. The design should allow a tolerance to account for initial calibration of the robot to find a zero position as well as a margin of error on how the actuator is placed within the vehicle.

### 3.2.6 Minimum Speed for Actuation

The actuators' required max speed can be approximated using recorded data from a human driver. This system will be simulating human behavior in the vehicle and should not be required to go beyond a normal roadway driving scenario. The following data was recorded during an hour of highway and city driving.



**Figure 3.5 Gas Pedal Position During Regular Driving**



**Figure 3.6 Rate of Change of Gas Pedal**

Figure 3.5 shows the recorded pedal position over the trip. The sampling time, determined by the sensor sampling rate on the vehicle, is 2.5 samples per second. Figure

3.6 shows the rate of change of the pedal position. This is a sample of the data that shows both the max and min rates of change. The vehicle returns a value of percent pedal position, 100% being fully compressed. From Figure 3.2, it is seen that the extent of the gas pedal is approximately 2.5 inches linear motion. The conversion is then 0.025 inches/% reading. The forward peak rate of change is **52.5%/sec** or approximately **1.31 inches/sec** linear motion. The reverse peak rate of change is **67.5%/sec** or approximately **1.6875 inch/sec** linear motion.

The required actuator speed should exceed the greatest recorded speed of **1.6875 inches/sec (4.286 cm/s)**.

### 3.2.7 Summary of Mechanical Requirements

The following table will summarize all the above mechanical requirements to a table with required quantified values where available.

<b>Description</b>	<b>Specific Requirement</b>	<b>Section Ref.</b>
Force Output to Gas	Negligible	3.2.1
Force Output to Brake	Exceed 20 Newtons	3.2.2
Length	Limit to 20 inches (50.8 cm)	3.2.3.1
Per Actuator Width	Limit to 5 inches (12.7 cm)	3.2.3.2
Height	Limit to 10.5 inches (26.67 cm)	3.2.3.3
Contact Tolerance	1.5 inches (3.81 cm)	3.2.3.4
Gas Actuation Distance	2.5 inches (6.35 cm)	3.2.4
Brake Actuation Distance	3.5 inches (8.89 cm)	3.2.5
Min. Actuation Speed	Exceed 1.6875 in/s (4.286 cm/s)	3.2.6

**Table 3.2 Summary of Mechanical Requirements**

### 3.3 Electrical System Requirements

This section will outline specific requirements of the electrical control system. This will include electrical safety systems as well as functional requirements.

#### 3.3.1 E-Stop

There must be an emergency override in between the main control and the accelerator actuator in order to release the gas pedal if there is a system failure. The override must include a manual trigger with an external Emergency Stop button and have system power loss detection. It is up to the user to recognize erratic behavior within the system and manually trigger an E-Stop. When triggered, main control is disengaged and the actuator is retracted from the gas pedal using a dedicated safety controller circuit. The safety circuit receives power from a secondary power source requiring a battery backup for the main system to operate.

#### 3.3.2 Manual Control

A manual control option will be implemented which will override automated control using a physical switch. Users will have control of the actuator's speed and direction. This feature is required for initial testing and calibration of hardware.

### 3.4 Controller Requirements

This section will outline controller requirements for the automated vehicle control system. Monitored and controlled variables as well as timing requirements determined by the Ford Escape Hybrid sensor sampling times.

### 3.4.1 Monitored Quantities

The following monitored quantities in Table 3.3 are required feedback variables for a closed loop controller implementation. The range of values are based on recorded data collected through the CAN Bus connection on the 2010 Ford Escape Hybrid and limitations imposed by the MD100 Dynamometer from Mustang Dyno. A brief description of each variable is also provided.

<b>Name</b>	<b>Unit</b>	<b>Range</b>	<b>Description</b>
Gas_Pos	Percent	0 to 100	Throttle Position Sensor
Brake_Pos	Torque	0 to -3748	Brake Position Feedback (Req.Torque)
Vehicle_Speed	km/h	0 to 100	Speed of the vehicle
Accel\Limit Fwd	Bool	0 or 1	Forward limit detection for throttle actuator
Accel\Limit Rev	Bool	0 or 1	Reverse limit detection for throttle actuator
Brake\Limit Fwd	Bool	0 or 1	Forward limit detection for brake actuator
Brake\Limit Rev	Bool	0 or 1	Reverse limit detection for brake actuator

**Table 3.3 Monitored Quantities**

### 3.4.2 Controlled Quantities

The following controlled quantities, in Table 3.4, are required for the implementation of an automated vehicle speed controller through pedal actuation.

<b>Name</b>	<b>Unit</b>	<b>Range</b>	<b>Description</b>
actuatorSpeedThrottle	Percent	0 to 100	Speed control
actuatorDirectionThrottle	Bool	0 to 1	Direction of Actuator
actuatorSpeedBrake	Percent	0 to 100	Speed control
actuatorDirectionBrake	Bool	0 to 1	Direction of Actuator

**Table 3.4 Controlled Quantities**

## 3.5 Safety Requirements

The testing environment involves electrically controlling a vehicle with its own power system onboard. The system is powered from an external power source independent of the vehicle. To ensure safety, the system must recognize when there is a power failure and take action to remove control from the vehicle. This will prevent the car from continuing to drive with no way to stop the vehicle remotely.

### 3.5.1 Modular design

The safety system should be separate from the control hardware. The safety system will have the ability to interrupt the control hardware in case of an emergency.

### 3.5.2 Operation

A mechanical dual pole switching circuit is required to break connections to the actuator motor when there is a power failure. When there is power detected, the switch will engage to normal operation mode. When there is a power loss or an E-Stop triggered, the mechanical switch will go into safety mode. A battery back-up system will take over and perform the required safety procedure.

### 3.5.3 Battery Detection

The system should not operate unless the battery backup is connected. The system must detect this connection to allow operation, and disengage main control if disconnected. The user must verify state of charge of battery before operation.

## 4 System Design

### 4.1 Overview

This chapter explains the system design to complete all requirements laid out in Chapter 3. The chapter is broken into mechanical, electrical and software design sections.

### 4.2 Mechanical Design

The mechanical design component of this project involves creating a linear actuator capable of compressing the accelerator and brake pedals of a 2010 Ford Escape Hybrid. Section 3.2 introduces the mechanical requirements and constraints. The following sections will break down the mechanical design stage.

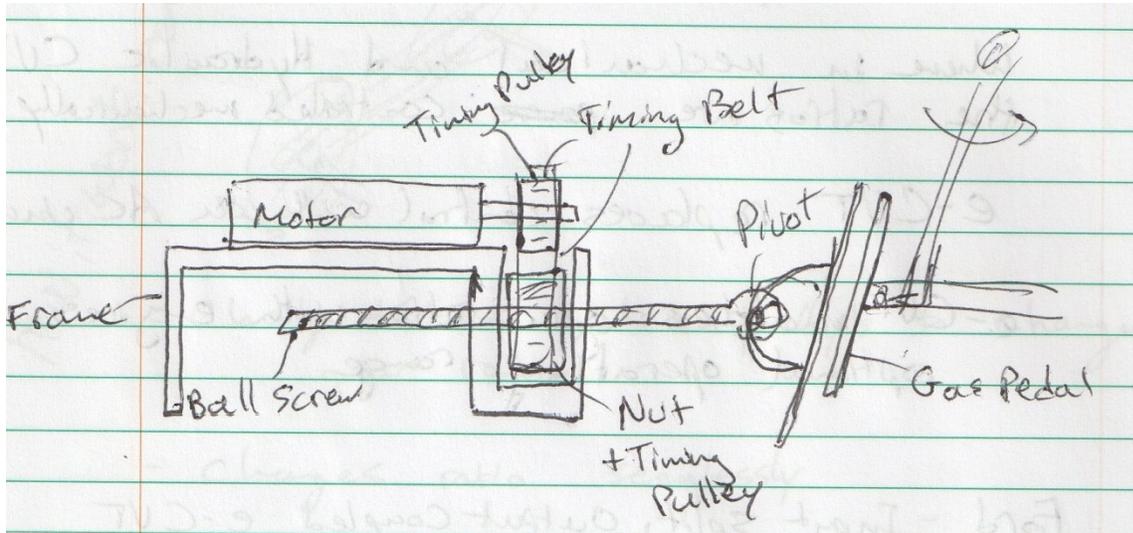
#### 4.2.1 Objective

The objective is to make a cost effective solution for electrically controlling the position of the accelerator and brake pedals of a 2010 Ford Escape Hybrid. The overall performance completes minimum requirements to operate an automatic transmission vehicle and does not have to meet specifications of commercial solutions introduced in Chapter 2.5. The actuator design will ideally be universal to each pedal (same design for both) and be able to be replicated in small quantities using standard parts, materials and simple machining steps.

#### 4.2.2 Initial Design Ideas

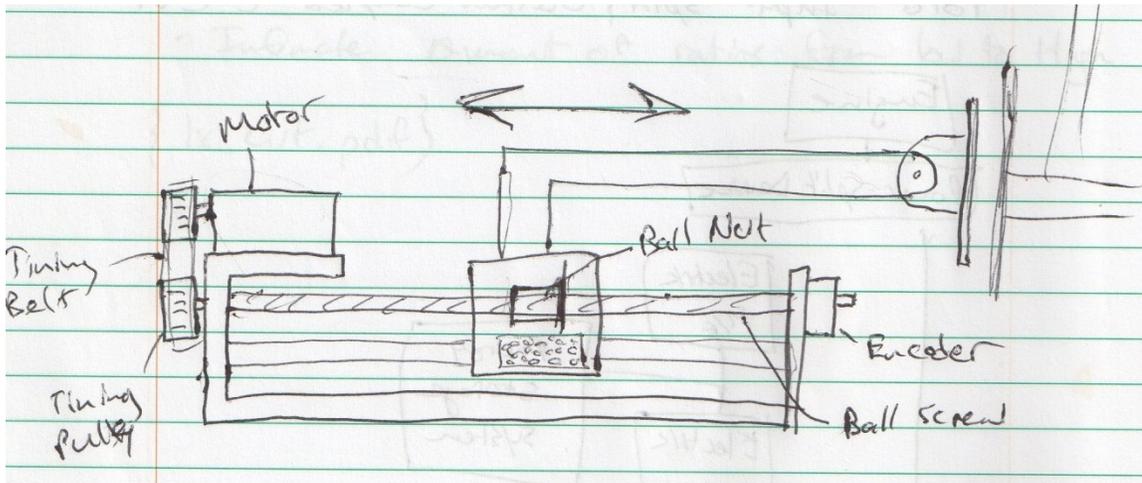
The initial design ideas started as sketches. The first idea involves a complex frame and to use a geared DC motor with a timing pulley to drive a ball screw forward.

The Motor turns a ball nut, which remains horizontally in place while the screw moves through it. The screw is fitted with a pivot joint attached to a plate making contact with the pedal. This is seen in the following illustration in Figure 4.1.



**Figure 4.1 Preliminary Sketch 1**

The next preliminary sketch involves a traveling nut on a ball screw. This design employs a geared DC motor driving a ball screw from one end and a traveling assembly to contact the pedal. There is a precision shaft to resist the traveling assembly from rotating as the ball screw turns. The ball screw is again driven by a timing pulley and belt. A contact plate is on the end of the traveling assembly, which makes contact with the pedal. This is illustrated in the following sketch in Figure 4.2.



**Figure 4.2 Preliminary Sketch 2**

The next design idea has a vertical support that attaches to the floor and the dashboard within the driver's cockpit. The frame is friction fitted, and adjusted to stay fixed in position. A rotational actuator uses a geared DC motor or Servo to compress and release the pedal, with an arc motion to closely match the arc of the pedal. The following sketch in Figure 4.3 illustrates this idea.

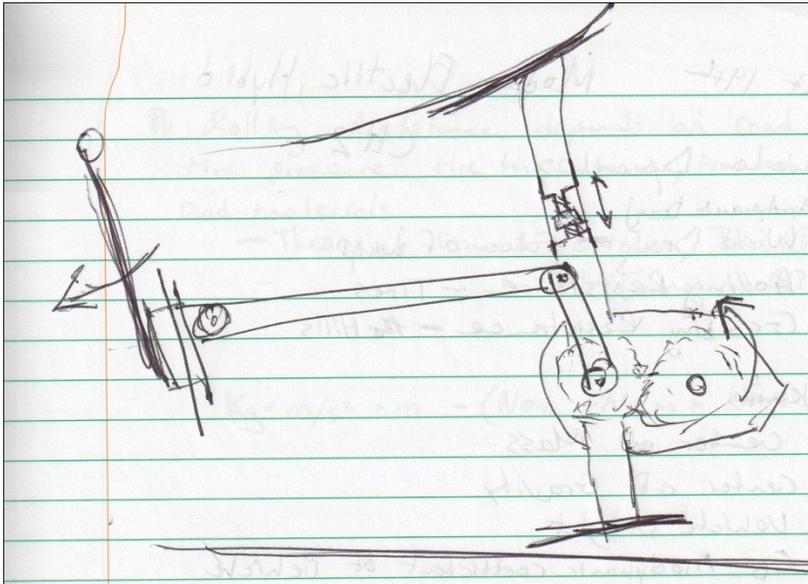


Figure 4.3 Preliminary Sketch 3

## 4.2.3 Design Decisions

This section is intended to explain design decisions related to the mechanical design component.

### 4.2.3.1 Actuation Method

It was determined that an electrical-mechanical actuation would be the simplest and most cost efficient solution. Other methods considered included hydraulic and pneumatic. Hydraulic has the benefit of extra power and accuracy at the cost of weight and potentially being messy. Pneumatic is a lighter option with off the shelf actuators available, but precision can be lost due to the difficulty of controlling position with pressure on light loads.

Using a geared DC motor for control is the simplest and most cost efficient way to achieve motion for this application. For the initial design, a high torque DC motor was selected that far exceeded the torque requirements. As outlined in Section 3.2.2, the max force required is 20 Newtons. Added torque comes with a trade-off of lower speed. This motor must meet the speed requirements outlined in Section 3.2.6. The motor selected for this design is the RS540 DC motor with 64:1 ratio gearbox from Banebots. This is a commonly used geared motor and available from Robotshop at a reasonable price. By selecting a hobbyist motor, it has a limitation of not having all the motor specs required to build a model for simulation. Table 4.1 outlines the motor characteristics given. A preliminary speed verification can be found in Section 4.2.7.2.

### Banebots 12V 263RPM 2527oz-in Planetary Gearmotor w/RS-540 Motor Specifications

#### Physical

Type	Planetary
Reduction	64:1
Gear Material	All Metal
Weight (Gearbox only)	7.2 oz (205g)
Weight (with motor)	12.6 oz (358g)
Length (Gearbox only)	1.9 in (49mm)
Length (with motor)	4 in (102mm)
Width (Square)	1.5 in (38mm)
Shaft Diameter	0.375 in (10mm)
Shaft Length	2.15 in (55mm)
Mounting Holes (2)	#10-32
Shaft End Tap	#8-32
Shaft Key	0.125 in (3.2mm)

#### Calculated Performance\*

Type	Planetary
Motor	RS-540
Operating v	4.5v – 12v
Nominal v	12v
No Load RPM	263
No Load A	1A
Stall Current	42A
Stall Torque	2527 oz-in / 17843 mN-m
Kt	60.2 oz-in/A / 425 mN-m/A
Kv	22 rpm/v

\*All performance numbers are calculated using motor data at Nominal Voltage. They do not factor in loss from the gearbox.

**Table 4.1 Geared DC Motor Specs**

#### 4.2.3.2 Linear Motion

A linear actuator was determined to be the easiest to implement. Other alternatives were to use an arc motion that followed the motion of the pedals arc. Through testing, it was found that a linear motion can maintain contact with the pedal through its arc.

There are different ways to achieve linear motion with a DC motor. Different methods include rack and pinion, lead screw, ball screw and belt driven. In order to reduce friction and wear on components, a ball screw was determined the best option. The cost of a ball screw assembly is higher than a lead screw but this also requires less torque and a less expensive motor can be used. The reduced friction will also make it easier to create a closed loop controller because of fewer contributions to steady state error. The actuator will be able to move in a smaller range without requiring as much proportional gain in the controller. With a ball screw, a higher pitch in thread can be selected which increases the linear speed we initially lost using a high ratio gearbox.

When comparing the two preliminary designs with linear motion, in Section 4.2.2, the second design is more feasible. This design requires less custom parts and off-the-shelf parts are more prone to have a stationary screw with a travelling ball nut (linear stage actuator).

To use a ball screw with a traveling nut, a guide rail must be added to the design. The guide rail serves the purpose of stabilizing the assembly as well as restricting the nut from rotating with the screw. Since this is another moving assembly, any added friction will affect the applied torque needed. Selecting a precision rod with a bushing for this

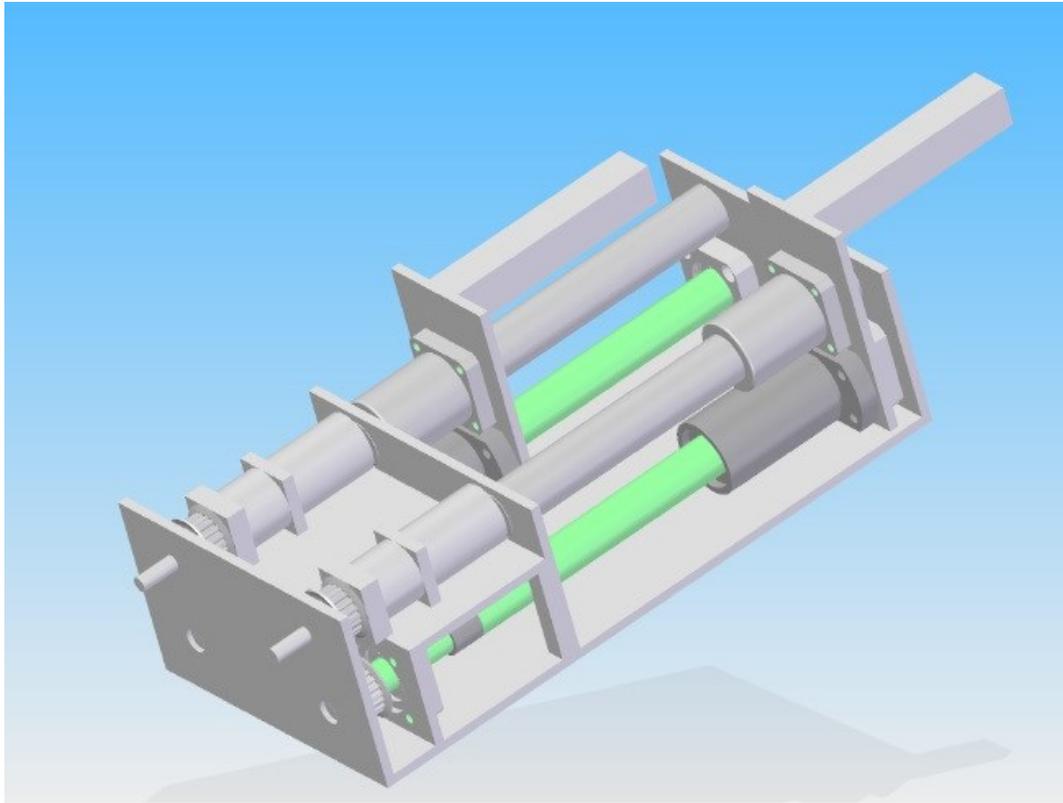
---

purpose reduces friction associated with imprecise machining. These are common parts and can be found in many sizes from mechanical component suppliers worldwide.

#### 4.2.3.3 Pedal Contact

After analyzing the pedal arcs within the test vehicle, it is found that each have a different motion. There is also a vertical component to the pedal's motion when a horizontal linear force is applied. Since the pedals have a rubber pad on them, simply compressing them with a metal plate or equivalent would create a vertical component of friction. By adding a caster to the plate, similar to what you find on a shopping cart, reduces the vertical friction without affecting the linear force applied. By carefully positioning the pedal contact point and selecting an appropriate sized caster, the design can be used on either pedal in the vehicle.

#### 4.2.4 First Prototype Design



**Figure 4.4 First Prototype Model**

##### 4.2.4.1 Description

The first prototype seen in Figure 4.4 was created as a proof of concept, and to get main components modeled. The motors are positioned on a stage above the ball screw and driven by a timing belt and pulley assembly. This design incorporates a pushing mechanism that is just a 1"x1" square tube welded to a plate connecting to the ball screw and guide bushing. There are very few off the shelf parts incorporated in this design. The base is a very complex piece consisting of multiple machined parts welded together.

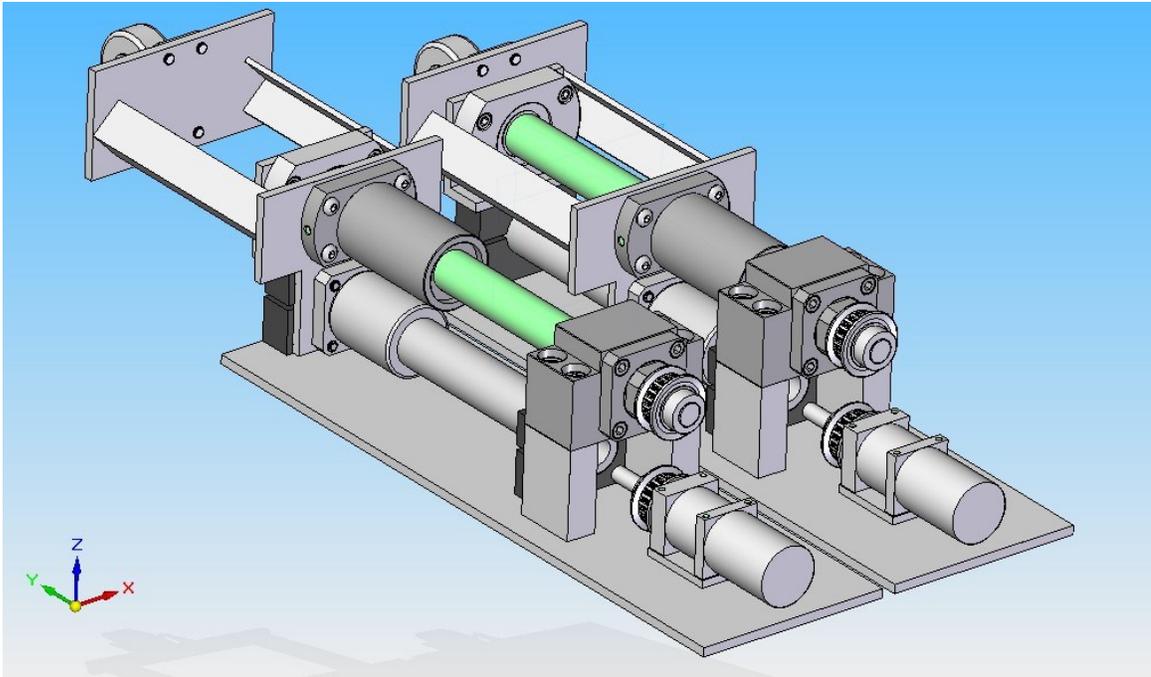
#### 4.2.4.2 Design Flaws

The major design flaws identified were that the ability to produce the complex base plate assembly would likely take a lot of time and not be very precise in the end (ie. it is hard to weld at perfect 90-degree angle butt joints). Having precision linear sliding parts anchored to pieces with a varying accuracy would defeat the purpose.

The other problem with this design was not having a great way of contacting the pedal. Using a flat-ended piece of metal contacting a rubber surface with vertical motion will create sliding frictions and possibly cause instability with the design. The contacting location of the pushing assembly is also too high for the pedal locations. They would have to be lowered from their current position.

Lastly, this design is made as one unit rather than being two duplicate actuators for each pedal. This makes the final assembly very heavy and creates larger pieces to machine.

## 4.2.5 Second Prototype Design



**Figure 4.5 Second Prototype Model**

### 4.2.5.1 Description

The second prototype design in Figure 4.5 was revised from the first based on comments and considerations. The main objective was to reduce the complexity of custom-machined parts and to use as many catalog parts as possible. Also creating one assembly for prototyping allows flaws to be worked out and corrected for the second iteration. Notable differences are that the base plate is now just one flat sheet of steel with machined holes for locating parts. Next, is that there are now catalog configurable parts from Misumi being used where possible. These parts are very precise and cost effective. The ball screw assembly with bearings and guide assembly are all configurable and selected to integrate into the design. The supports used for the ball screw are catalog

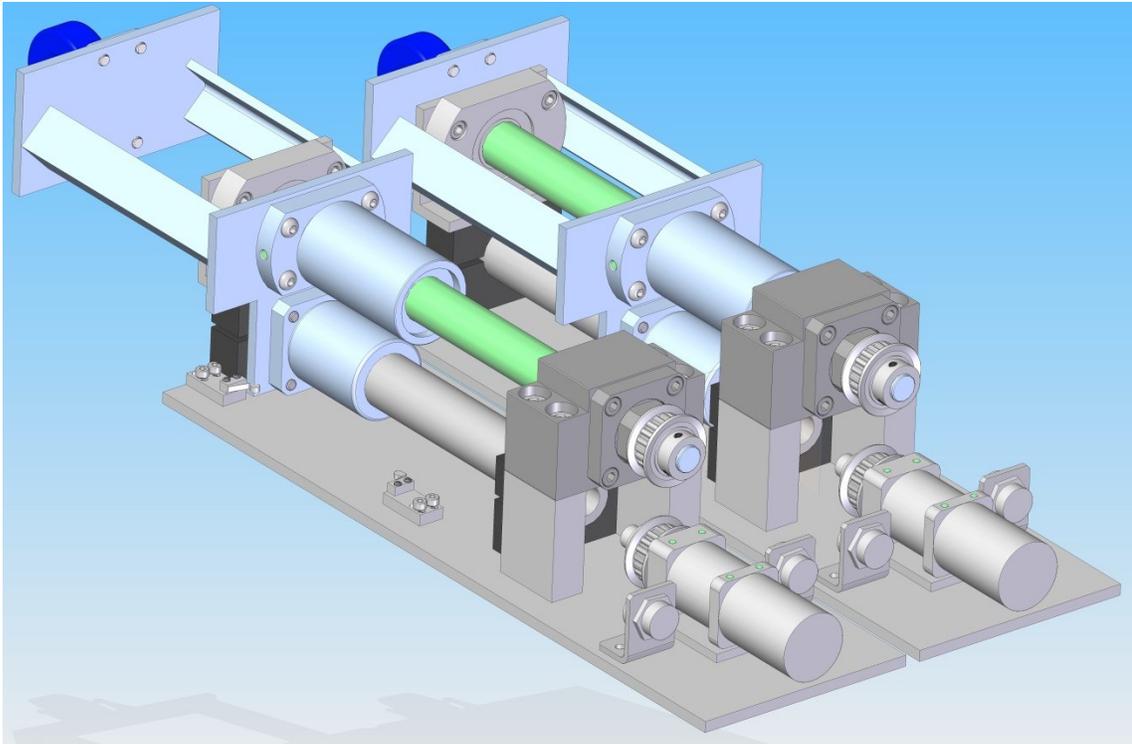
pieces and provide better precision than welded supports. The motor mounts on the base plate and uses a timing belt with pulley to drive the ball screw. The pedal contact assembly has been changed so that contact is lower. A caster wheel was added to reduce vertical friction when either pedal moves in its arc motion, as discussed in Section 4.2.3.3. This travelling assembly is welded together using angle iron between the two plates. Precision is not required for these weld joints and should not affect the functionality if the dimensions vary a bit.

#### 4.2.5.2 Design Flaws and Considerations

No major flaws were associated with the functionality of this design. One obstacle was that the motor supplier discontinued selling the geared dc motor originally purchased. The second iteration uses a new motor mount design to accommodate the next generation motor.

Some considerations were to add limit switch mounts as opposed to gluing or taping on lever limit switches. In addition, brackets to mount connectors for electronic connections (limit switches, motors power, and encoder feedback) should be considered make it easier to disassemble and transport the actuators.

## 4.2.6 Final Design



**Figure 4.6 Final Design Model**

### 4.2.6.1 Description

For the final design seen in Figure 4.6, electrical connector mounting brackets are added as well as limit switch supports for both forward and reverse directions. Two separate connectors are added in order to keep low current control signals separate from motor's main power. New gearboxes were selected for this design to replace the discontinued version. They are still 64:1 ratio with an RS-540 motor from Banebots, but have a different mounting pattern and different output shaft size. Appropriate changes were made to the configurable parts to accommodate this change.

## 4.2.7 Preliminary Design Verification

Using Solid Edge, prototyping time can be reduced by verifying some design requirements before parts are ordered. The following sections are a preliminary design review discussing mechanical functionality of the design in Figure 4.6.

### 4.2.7.1 Motion Simulation

In order to verify speed operation of the design, Solid Edge includes motion simulation tools. This is used to check for clearances and collisions during movement. A rotational speed is applied to the timing pulley on the geared DC motor, which then creates motion of the movable assembly through specified relationships in the Solid Edge assembly file. By inputting a rotational speed, specified by the geared motor datasheet, realistic motion can be simulated.

### 4.2.7.2 Max Speed Requirement Verification

The calculation of rotational speed through the gear ratios, a top speed can be calculated. The pulleys used create a 1:1 ratio. The ball screw selected has a pitch equal to one inch per revolution travel. The Motor's gear ratio used is 64:1 and the DC motor being used is an RS-540, rated at a top no load speed of 263 RPM after the gear box. This would indicate a top linear speed of 263 inches per minute or 4.38 inches per second. This number will decrease, as there will be a load added. With the ball screw having low friction, a conservative estimate would be about 3.75 inches/second top speed, which exceeds the required speed of 1.7 inches/second specified in Section 3.2.6.

#### 4.2.7.3 Actuation Distance Requirement Verification

From the simulation, we find that actuation distance is approximately 4.5 inches. This exceeds the required distance for both the gas and the brake pedal actuators of 2.5 and 3.5 inches, respectively, stated in Sections 3.2.4 and 3.2.5 in the requirements chapter. Verification can be seen by comparing Figure 4.7 and Figure 4.8.

#### 4.2.7.4 Size Requirement Verification

As outlined in Section 3.2.3, the size of the overall design is constrained by the physical dimensions of the vehicle. Figure 4.7 and Figure 4.8 show dimensioned drawings of the version 3 actuator design in the fully extended and compressed positions. Because of the low profile design, the actuators will fit in the car despite being about .5 inches longer than the required max dimension stated in Section 3.2.3. The extra length fits underneath the seat cushion.

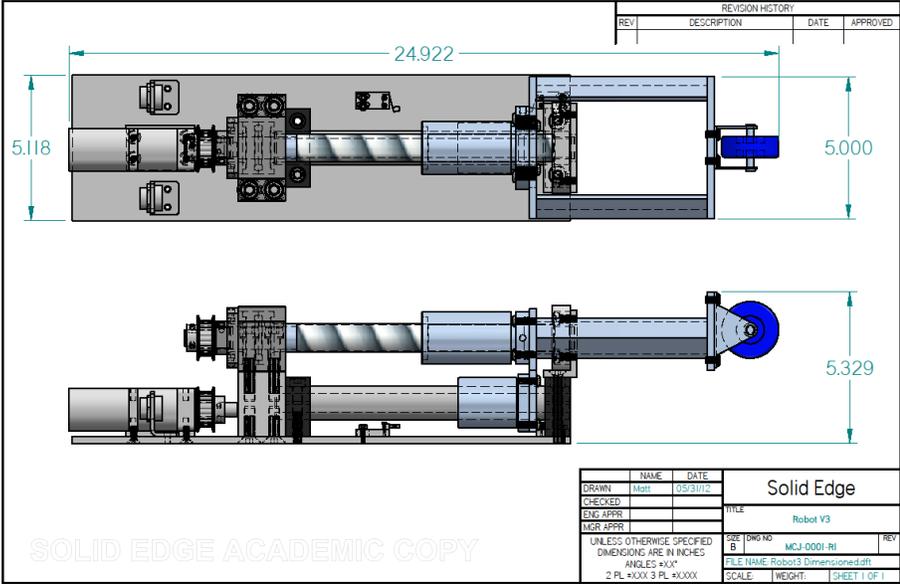


Figure 4.7 Actuator Dimensioned Drawing

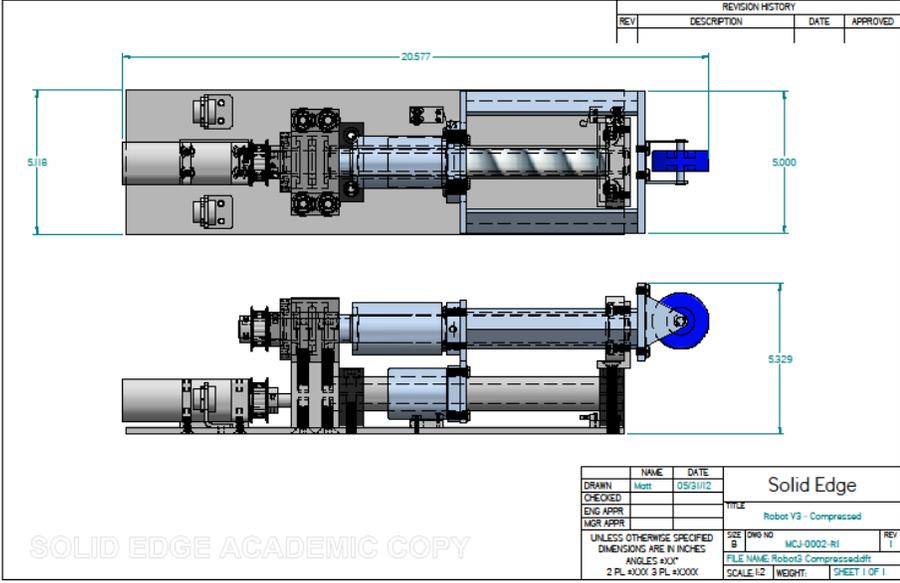


Figure 4.8 Robot 3 Compressed

## 4.2.8 Cost

Table 4.2 shows a breakdown for the cost of one actuator. These prices are for one actuator. The total cost of implementing 2 actuators is **\$3070.60**. The initial budget called for \$1000 for custom machining, and \$1400 for mechanical parts, totaling \$2400. To save costs, custom machining should be done by a volunteer student when possible.

Mechanical Components				Quantity	Unit Price	Total
Item	Description	Supplier	Part Number			
1	Ball Screw Fixed End	Misumi	BSW20	1	117.3	117.3
2	Ball Screw Support End	Misumi	BGCNB6204ZZ	1	18.3	18.3
3	BALL SCREW	Misumi	BSSZ2525-348-RLC-KLC-K10-S4	1	290.9	290.9
4	TIMING PULLEY - BALL	Misumi	ATP20XL037-B-P15	1	10.4	10.4
5	TIMING PULLEY - MOTO	Misumi	U-ATP20XL037-B-P0.50	1	18.4	18.4
6	PRECISION SHAFT	Misumi	SPJ30-254	1	21.3	21.3
7	BUSHING	Misumi	LHTS30	1	25.8	25.8
8	SHAFT HOLDER	Misumi	SHMWB30-30	2	45.9	91.8
9	MOTOR SPACER	Misumi	KPLNF6-A61-B40-X7-Y9.5-F47-G19-N6-NA6	1	24.8	24.8
10	BEARING L-BRACKET	Misumi	KPANU6-A60-BC28-L80-X40-Y35-F16-G17-V48-Q59-R34-M6-NA8-D24	1	58.6	58.6
11	TIMING BELT	Misumi	TBN90XL037	1	4.6	4.6
12	Connector Bracket	Misumi	KBANT3.2-A40-B25-L30-X15-Y25-F7-G15-V16-D19.5-NA5	2	9	18
<b>Fasteners</b>						
13	M6 1x18 soc cap screw	R.E. Glover	488.006.0018	8	0.11	0.88
14	10-32 x5/8 flat soc screw	R.E. Glover	195.011.0062	4	0.08	0.32
15	M10 1.5x55 soc cap screw	R.E. Glover	488.010.0055	4	0.66	2.64
16	M6 1x16 soc cap screw	R.E. Glover	488.006.0016	4	0.08	0.32
17	1/4-28 x1/2 button soc c/s	R.E. Glover	188.025.0050	4	0.12	0.48
18	M8 1.25x50 soc cap screw	R.E. Glover	488.008.0050	2	0.35	0.7
19	M10 1.5x25 flat soc screw	R.E. Glover	491.010.0025	4	0.46	1.84
20	M8 1.25x70 flat soc screw	R.E. Glover	491.008.0070	2	1.25	2.5
21	M4 12mm	Digikey	29324K-ND	4	0.47	1.88
22	M2 8mm lg	Digikey	H739-ND	1	8.1	8.1
<b>Metal</b>						
23	Standoff	Misumi	PSFPN-58-42-20	2	8.12	16.24
24	Pusher Large	Misumi	SS2FN-130-110-6	1	10.4	10.4
25	Frame	Misumi	SS2FN-444-130-6	1	31	31
26	Pusher Small	Misumi	SS2FN-130-70-6	1	7.8	7.8
27	Machining	MSL-JHE		1	750	750
					<b>Total</b>	<b>1535.3</b>

**Table 4.2: Mechanical Cost per Actuator**

## 4.3 Electrical Design

The following sections provide the details of the custom external hardware design and the interfaces made with other system components. A description of the design process as well as the final implementation of the electrical hardware components is presented.

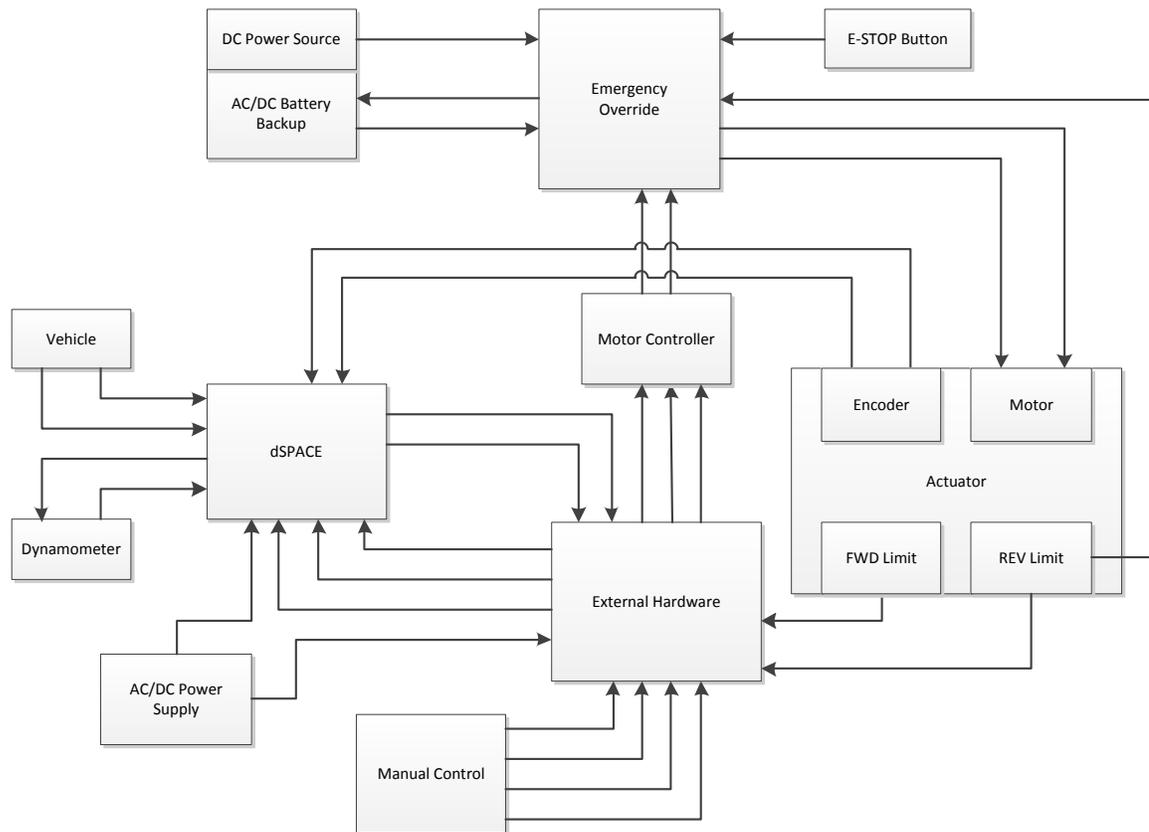
### 4.3.1 Objective

The objective of the electrical design component is to meet all the requirements outlined in Chapter 3.3. The electrical design section goes further to specify all the electrical connections between components and organization of the electrical system. This is designed in a modular way and be portable and robust. Extra care is taken to ensure all the designs are protected from environmental conditions to reduce the chance of a system failure. A modular system allows for controlled testing and verification of each component before integration.

### 4.3.2 System Module Breakdown

Figure 4.9 illustrates the hierarchy of the electrical components and their interface with other hardware. In order to avoid complexity in the diagram, only one actuator is represented. The second actuator will include the same external hardware modules but use the same centralized controller, the dSPACE MicroAutoBox. The arrows in Figure 4.9 provide a direction of connectivity through which control signals and power connections flow. These are used to indicate the number of control variables and direction exchanged between sub-systems. The power sources connected to the modules only indicate the

direction of positive power. A common ground connection is implied for the entire system and not represented on this diagram.



**Figure 4.9 Module Breakdown**

#### 4.3.2.1 Vehicle

The Vehicle module refers to the 2010 Ford Escape Hybrid used for our testing purposes. The connections being made are through the diagnostic port (OBD-II) and allow monitoring of the High Speed and Medium Speed CAN Bus lines.

#### 4.3.2.2 Dynamometer

The dynamometer module refers to the Mustang Dyno MD500, made available to McMaster for the GreenAuto Power Train project. Interfaces include monitoring and control through an RS232 serial connection from the dSPACE MicroAutobox. More information can be found in Section 2.3.

#### 4.3.2.3 Actuator

For the purpose of this section, the actuator is only described in terms of electrical connections. The actuator is equipped with 2 limit switches, an optional encoder for laboratory testing, and a DC motor.

#### 4.3.2.4 dSPACE MicroAutoBox



**Figure 4.10 dSPACE MicroAutoBox**

The dSPACE MicroAutoBox is an embedded microcontroller capable of 16 ADC inputs, 8 DAC outputs, 2 CAN Bus interfaces, RS232 interface, and over 100 digital I/O

pins. The MicroAutoBox is programmed using Simulink models as well as Real-Time Interface block sets from dSPACE. Programming this box involves compiling a Simulink model using the real-time workshop add-on in Matlab. The dSPACE ControlDesk software, is used to create custom user interfaces to the dSPACE box. Instrumentation is added to a layout which is then linked to real-time signals declared in the Simulink models. The link comes from a high speed communication via an external PCI cardExpress port through the laptop. Inputs and outputs on the MicroAutoBox are created using a zero insertion force (ZIF) connector shown in Figure 4.14. Figure 4.15 shows pin-out details for the dSPACE Box being used. Custom cables and connections are easily added through a supplied wiring harness. Figure 4.10 shows a custom parallel port connection (db25 cable) used to interface with an Onboard Diagnostic (OBD-II) cable.



**Figure 4.11 dSPACE Computer Interface**

Figure 4.11 shows the bus connection wire plugged into an external PCIExpress card. Using this link, data can be received and displayed in real-time. The dSPACE software is intended for the Windows platform. Using this hardware, data acquisition can be performed on any vehicle equipped with an OBD-II port using vehicle specific CAN database description files (see section 2.2 and 4.3.3 for more info on vehicle interface).

#### 4.3.2.5 External Hardware

The external hardware module contains connections to each part of the control hardware and integrates the dSPACE, Motor Controller, Manual Control and Actuator modules together. The external hardware module is a custom PCB containing logic chips, voltage regulators and circuit protection fuses.

#### 4.3.2.6 Motor Controller

The motor controller acts as an interpreter to control a DC motor from a 5V pulse width modulation (PWM) control signal coming from the dSPACE MicroAutoBox. There is a wide range of off-the-shelf motor controllers available online which are simple and easy to use. Robotshop.ca offers a variety of inexpensive controllers. Designing a controller circuit from scratch is possible but an inefficient use of time.

#### 4.3.2.7 Manual Control

The manual control module will allow a user to override the motor control signals from the dSPACE controller. Forward and reverse select buttons as well as motor speed will be controllable. Manual control is still bound by limit switches through the external hardware module.

#### 4.3.2.8 Emergency Override

A separate safety control system will be implemented to create an isolated circuit designed to be simple to reduce software or component faults from impacting safety. This system will be connected to a battery backup in case of a power failure. A dedicated motor controller is used to ensure operation of the actuator when the main controller fails. Using a single purpose microcontroller, the safety system will drive the motor backwards until the reverse limit switch is triggered in order to remove control of the accelerator pedal during a triggered emergency state.

#### 4.3.3 Vehicle Interfaces

The electrical and control system will require numerous feedback signals from the vehicle for the control system design. The electrical interface required to the vehicle is outlined in Section 4.3.5.5.

##### 4.3.3.1 High Speed CAN/ Medium Speed CAN

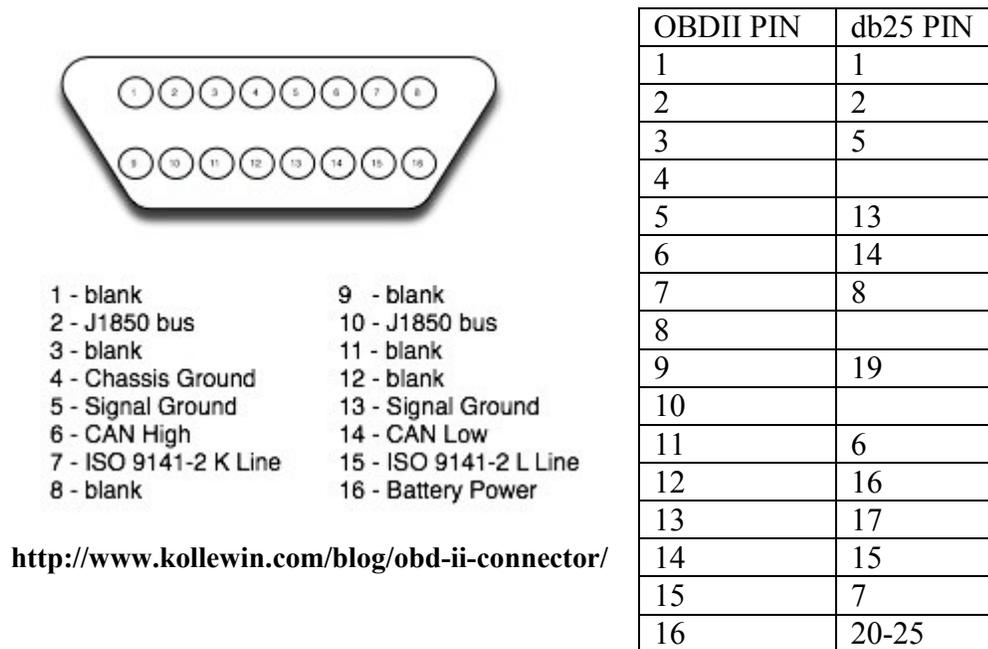
The high speed CAN line on the Ford Escape Hybrid contains high priority information about the vehicle's operation. The Medium Speed CAN also contains important information such as ignition status and dashboard monitored values, which are needed for the software interface design. The CAN bus lines are monitored to receive information on vehicle speed, engine speed, pedal positions and other information that is required for researchers using this equipment. The dSPACE MicroAutoBox is equipped with a CAN controller for interfacing with this vehicle communication system. The RTI CAN Blockset from dSPACE is used to receive and interpret these signals.

### 4.3.3.2 Physical Connection to Vehicle

This project has purchased a NEO FIRE VI CAN interface module, pictured in Figure 4.12, which can be used for monitoring and controlling signals. This connects to a vehicles CAN Bus using the onboard diagnostic (OBDII) port using the supplied cable which breaks out the OBDII connection into a db25 female plug. To interface the car with the dSPACE box, the High Speed CAN lines need to be connected to the CAN controller within the dSPACE box using a mating male db25 connector. The following table in Figure 4.13 breaks out the connections between OBDII and the db25 connector.



Figure 4.12 NEO Fire VI <http://intrepidcs.com>



**Figure 4.13 OBDII to db25 Pin Out**

#### 4.3.4 dSPACE Interface Description

The provided dSPACE control unit operates the mechanical actuators through custom circuitry. The following subsections outline these connections.

##### 4.3.4.1 Pulse Width Modulation

The speed of actuation on the vehicles pedals are controlled using a Pulse Width Modulation signal generated from the dSPACE box. This directly connects to a motor controller that is capable of sourcing the required current to the drive motor.

##### 4.3.4.2 Direction Control

The direction of motion of the actuator is controlled using a single digital output from the dSPACE box. This will ensure there are no mixed signals regarding direction are being sent. However, the motor controller being used requites separate digital signals for

each direction. By using a NOT logic gate chip, separate control signals can be produced that are opposite to one another, disabling one direction and activating the other.

#### 4.3.4.3 Motor Current Feedback

The electrical control circuitry provides a signal to the dSPACE to indicate the current draw of each motor. An analog source is put into an analog to digital converter (ADC) and scaled to the proper current. This is used to ensure the motors and motor controllers are operating within their required current regions. During typical operation, current can range up to 5A draw from the motor. If this limit is exceeded, a system fault should be tripped in the control software. The motor controller has stall detection, so this is not a functional requirement but redundant safety measures are good practice.

#### 4.3.4.4 Limit Switches

Physical limit switches are installed on the forward and reverse directions of each actuator. When a switch is tripped, it prevents motion in the corresponding direction. The system still allows movement in the opposite direction for system recovery. Hardware and software limit switch detection are implemented for redundancy. If there is a hardware or software problem, where they do not detect the limit being triggered, the counterpart detection prevents motion to minimize damage. Software limit switch feedback is achieved through a digital input to the dSPACE box from each switch.

#### 4.3.4.5 Encoder input

In order to do laboratory testing, a digital encoder is used for position feedback for initial controller design and testing. The Quadrature Encoder interface requires two digital input lines as well as a power supply and ground from the dSPACE.

#### 4.3.4.6 Actuator Control

The actuator uses a brushed DC motor that has a negative and positive terminal for control. The speed of the actuator is adjusted by varying the voltage to the motor using a motor controller. The motor controller selected must perform a forward and reverse direction control and source the required amps to the DC motor chosen.

#### 4.3.4.7 Dynamometer Interface

Dynamometers are typically connected and adjusted using a serial connection through an RS232 port. From there, resistances can be adjusted and sensor readings are received. Typically this is done using software provided by the manufacturer. The goal for this project is to interface the dSPACE box and use the RS232 serial controller on pins U6 and U5 (dSPACE GmbH, 2009). For this to be implemented more information is required from the manufacturer on their control protocol.

### 4.3.5 Detailed dSPACE Connections

This section will provide detailed connection information for peripheral connections to the dSPACE MicroAutoBox. Available connections as well as I/O pins used are provided for maintenance purposes.

#### 4.3.5.1 Wire Harness

The dSPACE MicroAutoBox uses a custom wire harness, which connects through a Zero Insertion Force (ZIF) Connector. Wires are added to the ZIF connector using a dSPACE supplied crimping tool that attaches a contact tip to the wires. Figure 4.14 shows a picture of the Male Connector with custom attached wires for this project.



Figure 4.14 Wire Harness

4.3.5.2 1401/1501 Datasheet

Figure 4.15 is from the dSPACE documentation for the 1401/1501 dSPACE MicroAutoBox. This provides pin labels used for configuring Simulink dSPACE blocks.

**Data Sheet MicroAutoBox 1401/1501**

The following table shows the signals of the I/O connector:

1	2	3	4	5	6	
DAC 7 out	DAC 8 out	Group 1 ch 1 out	Group 1 ch 2 out	Group 1 ch 3 out	Group 1 ch 4 out	<b>A</b>
DAC 5 out	DAC 6 out	Group 1 ch 5 out	Group 1 ch 6 out	Group 1 ch 7 out	CTM ch 1 out	<b>B</b>
DAC 3 out	DAC 4 out	CTM ch 2 out	CTM ch 5 out	CTM ch 6 out	CTM ch 7 out	<b>C</b>
DAC 1 out	DAC 2 out	CTM ch 8 out	CTM ch 3 out	CTM ch 4 out	Group 6 ch 1 out	<b>D</b>
VDRIVE in	VSSENS out	Group 6 ch 2 out	Group 6 ch 3 out	Group 6 ch 4 out	Group 6 ch 5 out	<b>E</b>
Group 6 ch 6 out	Group 6 ch 7 out	GND	GND in	GND in	Group 6 ch 8 out	TPU ch 1 out
TPU ch 2 out	TPU ch 3 out	GND	GND in	GND in	TPU ch 4 out	<b>G</b>
TPU ch 5 out	TPU ch 6 out	GND	GND in	TPU ch 7 out	TPU ch 8 out	<b>H</b>
TPU ch 9 out	TPU ch 10 out	TPU ch 11 out	TPU ch 12 out	TPU ch 13 out	TPU ch 14 out	<b>J</b>
TPU ch 15 out	TPU ch 16 out	Group 2 ch 1 out	Group 2 ch 2 out	Group 2 ch 3 out	Group 2 ch 4 out	<b>K</b>
Group 2 ch 5 out	Group 2 ch 6 out	Group 2 ch 7 out	Group 2 ch 8 out	Group 3 ch 1 out	Group 3 ch 2 out	<b>L</b>
Group 3 ch 3 out	CTM ch 1 in	REMOTE in	CTM ch 2 in	CTM ch 3 in	CTM ch 4 in	<b>M</b>
Group 6 ch 1 in	Group 6 ch 2 in	Group 6 ch 3 in	Group 6 ch 4 in	Group 6 ch 5 in	Group 6 ch 6 in	<b>N</b>
						
Group 6 ch 7 in	Group 6 ch 8 in	TPU ch 1 in	TPU ch 2 in	TPU ch 3 in	TPU ch 4 in	<b>P</b>
TPU ch 5 in	TPU ch 6 in	TPU ch 7 in	TPU ch 8 in	TPU ch 9 in	TPU ch 10 in	<b>R</b>
TPU ch 11 in	TPU ch 12 in	TPU ch 13 in	TPU ch 14 in	TPU ch 15 in	TPU ch 16 in	<b>S</b>
Group 2 ch 1 in	Group 2 ch 2 in	Group 2 ch 3 in	Group 2 ch 4 in	Group 2 ch 5 in	Group 2 ch 6 in	<b>T</b>
Group 2 ch 7 in	ADC Type 1 Con 4 Ch 4 in	Serial 2 K / LIN i/o	Serial 2 L in	Serial 1 TXD out	Serial 1 RXD in	<b>U</b>
ADC Type 1 Con 2 Ch 4 in	ADC Type 1 Con 3 Ch 4 in	VBAT in	VBAT in	CAN 1 low i/o	CAN 1 high i/o	<b>V</b>
ADC Type 1 Con 1 Ch 4 in	ADC Type 1 Con 4 Ch 3 in	VBAT in	VBAT in	VBAT in	Group 2 ch 8 in	<b>W</b>
ADC Type 1 Con 2 Ch 3 in	ADC Type 1 Con 3 Ch 3 in	VBAT in	VBAT in	CAN 2 low i/o	CAN 2 high i/o	<b>X</b>
ADC Type 1 Con 1 Ch 3 in	ADC Type 1 Con 4 Ch 2 in	Group 4 ch 1 in	Group 4 ch 2 in	ECU / IF RX+ in	Group 4 ch 3 in	<b>Y</b>
ADC Type 1 Con 2 Ch 2 in	ADC Type 1 Con 3 Ch 2 in	Group 4 ch 4 in	Group 4 ch 5 in	ECU / IF RX- in	Group 4 ch 6 in	<b>Z</b>
ADC Type 1 Con 1 Ch 2 in	ADC Type 1 Con 4 Ch 1 in	Group 4 ch 7 in	Group 4 ch 8 in	ECU / IF TX- out	Group 5 ch 1 in	<b>a</b>
ADC Type 1 Con 2 Ch 1 in	ADC Type 1 Con 3 Ch 1 in	Group 5 ch 2 in	Group 5 ch 3 in	ECU / IF TX+ out	Group 5 ch 4 in	<b>b</b>
ADC Type 1 Con 1 Ch 1 in	VBAT prot out	Group 5 ch 5 in	Group 5 ch 6 in	Group 5 ch 7 in	Group 5 ch 8 in	<b>c</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	

Figure 4.15 1401/1501 Datasheet (dSPACE GmbH, 2009)

#### 4.3.5.3 Actuator 1 Connections – Accelerator

This section is intended to document the electrical connections made to the Accelerator Actuator control circuit from the dSPACE MicroAutoBox. This is a useful reference when designing control software and debugging the hardware.

<b>dSPACE Pin</b>	<b>dSPACE Pin Description</b>	<b>Wire Colour</b>	<b>db9 Pin Number</b>	<b>Signal Function</b>
N/C	N/C	Black	1	N/C
A4	Group 1 Ch2 out	Grey	2	Direction Select
G1	TPU ch2 out	Yellow	3	PWM Out
c1	ADC Type 1 in Con1 Ch 1	Violet	4	Current Sense (for drive motor)
H4	GND	Orange	5	GND
T1	Group 2 Ch1 in	Red	6	Limit Fwd
T2	Group 2 Ch 2 in	Brown	7	Limit Rev
T3	Group 2 Ch 3 in	Green	8	Digital In 1
T4	Group 2 Ch 4 in	Blue	9	Digital In 2

**Table 4.3 dSPACE Connections to External Hardware – Accelerator Control**

#### 4.3.5.4 Actuator 2 Connections- Brake

The following table documents the electrical connections made to the Brake Actuator control circuit from the dSPACE MicroAutoBox.

<b>dSPACE Pin</b>	<b>dSPACE Pin Description</b>	<b>Wire Colour</b>	<b>db9 Pin Number</b>	<b>Signal Function</b>
N/C	N/C	Black	1	N/C
A3	Group 1 Ch1 out	Grey	2	Direction Select
G2	TPU ch3 out	Yellow	3	PWM Out
Z1	ADC Type 1 in Con2 Ch 2	Violet	4	Current Sense (for drive motor)
H4	GND	Orange	5	GND
N1	Group 6 Ch1 in	Red	6	Limit Fwd
N2	Group 6 Ch 2 in	Brown	7	Limit Rev
P1	Group 6 Ch 7 in	Green	8	Digital In 1
P2	Group 6 Ch 8 in	Blue	9	Digital In 2

**Table 4.4 dSPACE Connections to External Hardware – Brake Control**

#### 4.3.5.5 CAN Bus Connections

The CAN Bus connection is made using an OBDII to db25 cable as described in section 4.3.3.2. The following table outlines the db25 cable's connections to the dSPACE MicroAutoBox.

<b>dSPACE Pin</b>	<b>dSPACE Pin Description</b>	<b>Wire Colour</b>	<b>db25 Pin Number</b>	<b>Signal Function</b>
V6	CAN 1 High	Brown+White	14	HS CAN (+)
V5	CAN 1 Low	Red+White	15	HS CAN (-)
W6	CAN 2 High	Green	5	MS CAN (+)
W5	CAN 2 Low	Dark Blue	6	MS CAN (-)
G5	GND	Light Blue	13	Signal Ground
N/C	N/C	Green+White	20-25	12 Volts DC in
U3	Serial 2 K/LIN i/o	Grey	8	K Line
U4	Serial 2 L in	Purple	7	L Line

**Table 4.5 dSPACE Connection to Vehicle Pin Description**

#### 4.3.5.6 Encoder Connections

An encoder can be used for testing and debugging of the system when actuator position feedback information from the CAN Bus is unavailable. The following chart describes the connections made to the dSPACE box from an encoder.

<b>dSPACE Pin</b>	<b>dSPACE Pin Description</b>	<b>Wire Colour (Ethernet Cable)</b>	<b>Signal Function</b>
b4	Group 5 Ch 3 in	Yellow	Channel A
b3	Group 5 Ch 2 in	Yellow-Dash	Channel B
E2	VSense	Blue	5V DC
G5	GND	Blue Dash	GND

**Table 4.6 dSPACE Connection to Encoder**

#### 4.3.5.7 Dynamometer Connections

The connection to the dynamometer is a future consideration for this project. It will communicate to the dSPACE through an RS232 connection. The following is the pin out for the dSPACE MicroAutoBox Serial controller.

<b>dSPACE Pin</b>	<b>dSPACE Pin Description</b>	<b>Cable Pin Number</b>	<b>Signal Function</b>
U5	Serial 1 TXD out	TBD	Transmit
U6	Serial 1 RXD in	TBD	Receive

**Table 4.7 dSPACE to Dynamometer Connections**

#### 4.3.5.8 Computer Connection

The computer connection is made through the dSPACE DS821 Link Board. This is an external adaptor which connects through the ExpressCard/54 slot on a laptop. The communication link is then recognized by dSPACE software through Matlab and dSPACE ControlDesk.

#### 4.3.6 Power Supply

The following sections outline the main power sinks and their requirements to select a suitable system power source. Power is provided externally to the vehicle to minimize effects on vehicle efficiency readings.

##### 4.3.6.1 dSPACE Power Requirement

The dSPACE MicroAutoBox requires a power source from 6 to 40 volts DC which can supply up to 5 amps. A capacitor equal or greater to 1000uF is required at the

power input (dSPACE GmbH, 2009). The typical power input is 12 volts, which can be sourced from the vehicle through the OBDII connection or an external source.

#### 4.3.6.2 Actuator Power Requirement

A sufficient power supply will be needed to power one actuator at a time. Software should limit the gas and brake pedals from being driven together, similar to typical vehicle operation. Continuous current and peak current characteristics of the motors will determine which power supply is selected. Using a motor controller with stall current avoidance will reduce the required current needed. See section 4.3.7.2 for details on the motor controller design decisions. The motor current draw will be limited to a 10A output using a fuse to prevent damaging the power supply. From testing, the average current draw is well below 10A under normal conditions.

#### 4.3.6.3 External Circuitry Power Requirement

External circuitry for safety and signal conditioning as well as manual control requires a limited amount of power. The circuitry runs 5V logic and regulated down from a common 12-volt power supply. Power spikes that come from sudden power draw for the actuators will need to be sufficiently filtered. The current requirements are approximately 200mA and are considered negligible compared to other power sinks in this design.

#### 4.3.6.4 Selected Power Supply

The power supply selected is from TDK-Lambda Americas Inc., Model LS200. It is a 200 watt, 12-volt DC power supply capable of supplying 16.7 Amps. This is a closed frame, chassis mount supply that is 79% efficient with a cooling fan built in. It has screw

terminals on the front for power connections, capable of up to four separate connections. This will power two motor control circuits, one safety circuit and the dSPACE MicroAutoBox. This was selected based on price and flexibility. Having a closed case provides added safety and prevention against damage to the internal circuitry.

### 4.3.7 Schematic Design for Custom Hardware

The following sections will provide details on design decisions for custom circuitry used in this system.

#### 4.3.7.1 External Hardware Module

The external hardware module, seen in Figure 4.9, is used to convey control signals to the actuator and monitored signals from the sensors back to the dSPACE MicroAutoBox. This circuitry also incorporates the manual control module, using an onboard microcontroller. The hardware provides override circuitry to disable power to the motor for one specific direction when the corresponding limit switch is triggered. The schematic design outlines connections needed for this circuit.

#### 4.3.7.2 Motor Controller

The purpose of the motor controller module is to translate low voltage control signals from the dSPACE MicroAutoBox central computer, into a driving voltage to the motor. There is a variety of off-the-shelf motor controllers available from hobbyist electronic sources. The controller was selected based on the motor being used. In section 4.2.3.1, the selection of a Banebots geared DC motor is discussed. Table 4.1 shows the properties of the motor with operating voltage ranges of 4.5 to 12V. The no load current is

1A at 12V and 42A stall current. The stall current exceeds the capabilities of many hobbyist motor controllers, and it will be necessary to find a solution with stall prevention circuitry built in.

Another deciding factor for motor controller selection is the ability to perform “silent” operation. Motor controllers receive a pulse width modulation (PWM) signal sent at constant frequency that is converted to a supply voltage at the same frequency. Motors that are controlled by PWM signals in the audible frequency range produce a high-pitched sound. For silent operation, the frequency for the input signal would typically be at 20kHz, above the human hearing threshold. Applying this criterion to the motor selection narrowed the list down to a hand full of options.

The simplest, most cost effective, off-the-shelf controller was chosen. The Pololu MD01B breakout board with a VNH2SP30 motor controller IC was selected, seen in Figure 4.18. This decision was based on voltage ranges, current output capabilities, control input frequency, current feedback feature, price, and availability. Rather than creating a footprint for the motor controller chip, seen in the schematic in Figure 4.17, a footprint for the breakout board was created. This eliminates the need for surface mount soldering in the final circuit design, and reduces the complexity of the overall design.

#### 4.3.7.3 Hardware Integration

The Motor Controller Board has been designed to accommodate the Pololu MD01B breakout board. A custom schematic representation can be seen in Figure 4.16. The external hardware includes limit switch inputs and two digital connections. External

connections are made using screw terminals for bare wire connections. This design decision is intended to make it easy to swap out a malfunctioning board. The spare digital lines were initially designed for encoder input purposes, but through testing it was discovered these connections were too noisy to have reliable pulse edge detection for a position feedback.

The manual control module in Figure 4.9 has been incorporated into the Motor Controller Board PCB. The manual control consists of a forward and reverse selection buttons (S2 and S3), a potentiometer to vary motor speed (unlabeled) and an AVR ATtiny45 microcontroller to convert the user inputs to the appropriate control signals for the motor controller. For development and testing, programming pins were added to the board (J1). The ATtiny has 6 I/O pins which also act as the programming pins to save space on its footprint. In order to do in-circuit serial programming (ISP), the pins need to be disconnected from external interference. A switch was added to disconnect the potentiometer while programming (S4). Manual or dSPACE control signals are engaged using a double pole, double throw (DPDT) slide switch (S1).

Additional circuitry has been added for redundant safety. In the case of a software failure, where the limit switch feedback does not stop the actuator, logic gates are used to detect and disable the motor controllers. The limit switches are normally closed and will produce a VCC logic signal that will be received by an AND gate. The direction select signal is a binary source, VCC for forward direction and GND for reverse. By using two AND gates and one NOT gate, this binary signal is transformed into two separate control

signals sent to the Channel A and Channel B control pins on the Pololu motor controller. The desired direction will only be enabled provided a limit switch is not triggered for the requested direction. The logic chips are seen in the schematic as 7404 for the NOT gate and 7408 for the AND gate.



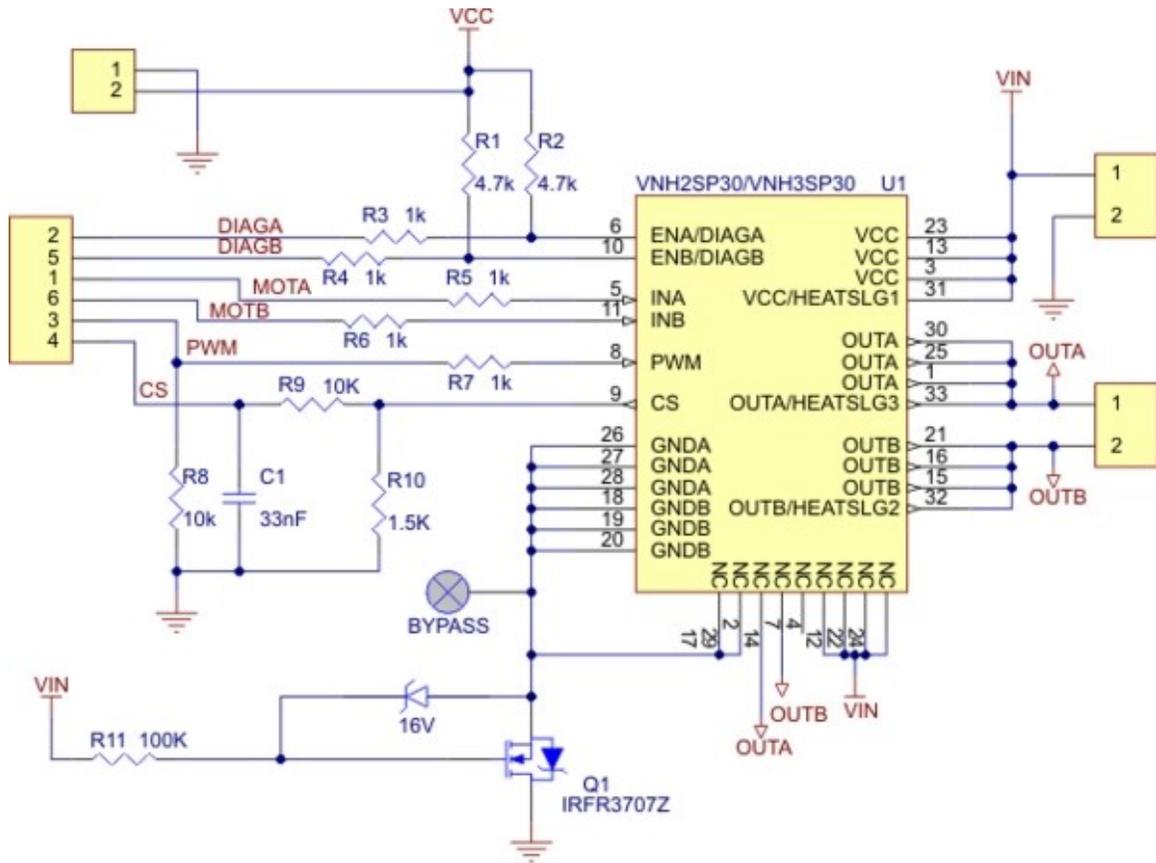


Figure 4.17 VN12SP30 High Current Motor Driver IC Schematic

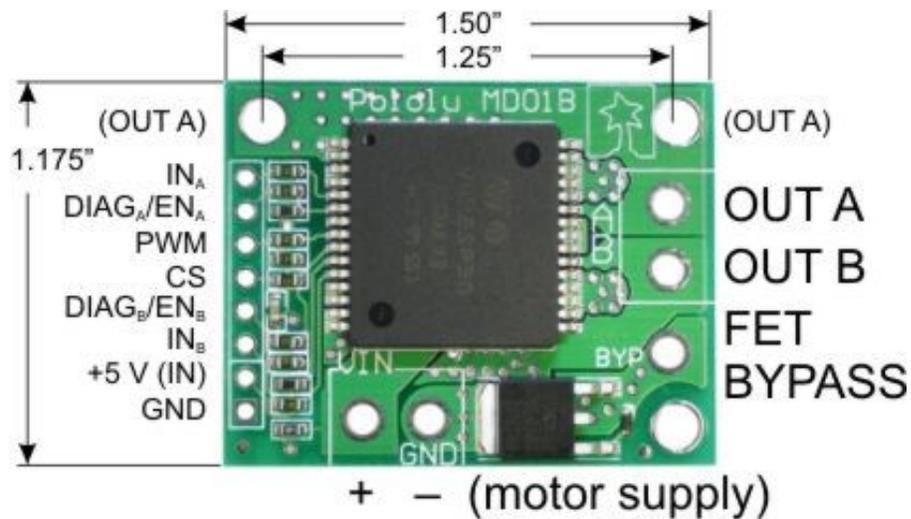


Figure 4.18 Pololu Motor Controller Break-out Board ([www.robotshop.ca](http://www.robotshop.ca))

#### 4.3.7.4 Safety Circuit

This project requires special attention to safety. It is typical for safety critical systems to include separate safety hardware to the main control hardware. The safety circuit design will have the ability to override all control signals that are directly connected to the actuators controlling the vehicle. Scenarios to consider are power failure, software failure, and hardware failure.

In a power failure scenario, the main controller would lose the ability to control the actuators. The dynamometer also requires power to operate and may lead to catastrophic consequences if there is no emergency shutdown procedure in place. The second scenario is a software failure that produces undesired results, creating a dangerous, uncontrollable event. If there is no way to release control quickly of the vehicle from the automated system, the speed limitations of the dynamometer may be exceeded. This would also apply for the third scenario, where a part of the control hardware has failed leaving the system unresponsive, or causing unwanted behaviour.

The safety system is independent of external power, and able to disconnect the actuators from the main control signals and recover the system to a safe state. This is accomplished with a battery source with a separate motor controller. The safety system is able to detect a power failure, using the voltage detect circuitry in section 4.3.7.5, as well as have a manual E-Stop button to initiate a shutdown procedure.

The schematic in Figure 4.19 employs 2 mechanical relays that are triggered by the main wall power source. If power is present, then the relays close and complete the

circuit, connecting the main control hardware to the actuators. When there is a power failure, the relays connect the actuators to the backup control system, which has a single purpose microcontroller to drive the actuator in reverse, until the limit switch is triggered, releasing the pedal. The manual operation will involve an Emergency switch that will simulate a power failure by breaking the trigger voltage to the relays.

The safety hardware is implemented on the accelerator pedal, as it's the power input for the vehicle. If the brake is compressed, and uncontrollable, this will not lead to any problems on its own. The E-stop button must be in reach of an operator continually monitoring the systems operation capable of identify undesirable actions.

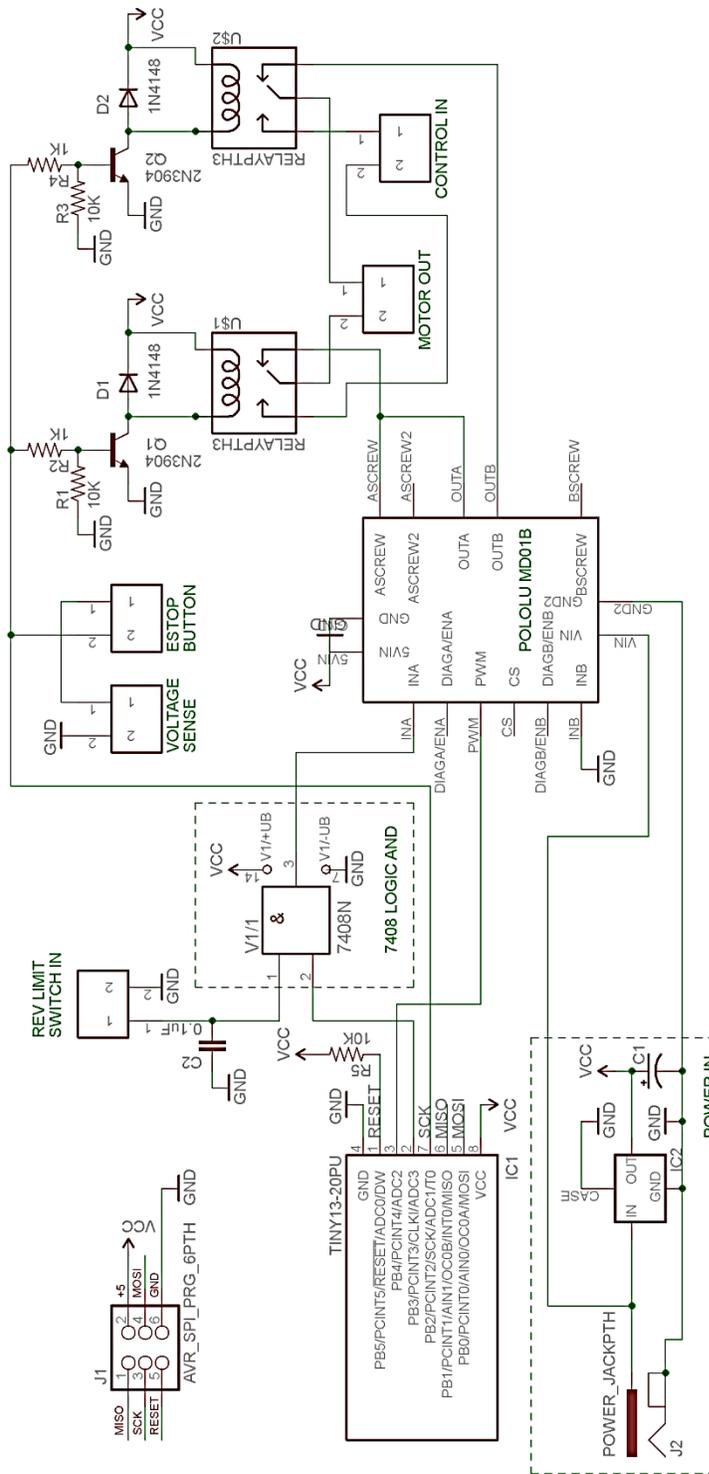


Figure 4.19 External Hardware - Safety Circuit Schematic

#### 4.3.7.5 Voltage Detect

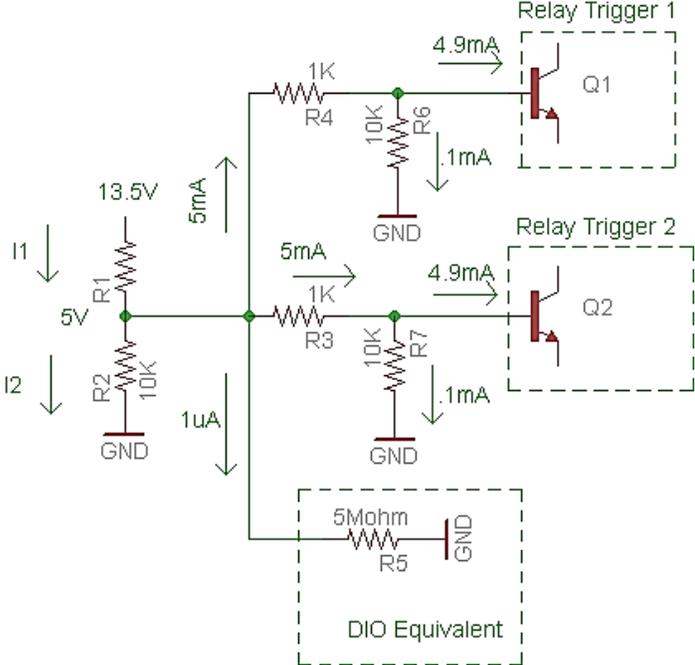
The purpose of the voltage detect circuit is to monitor the system's main power supply and detect whether or not it is powered. The circuit uses a voltage divider circuit to supply 5 volts to the safety circuit with enough current to trigger the transistors controlling the two relays as well as provide a digital enable input to the MCU onboard.

The power supply being monitored is 13.5 V DC which that powers the battery back-up. This was selected because it is already converted to DC and it will ensure the system cannot be started unless the battery backup is plugged in and charging. In order to maintain a 5V logic signal at the voltage divider's bridge, the initial 13.5 driving voltage must be reduced 8.5 V ( $13.5 - 5 = 8.5V$ ). To do this the equation  $V=IR$  is used to calculate the resistors needed. The source current needed is 12.5mA. A 10K resistor is selected as R2 value. To solve for R1, the following equations are used:

$$\frac{13.5 - 5}{I1} = R1$$

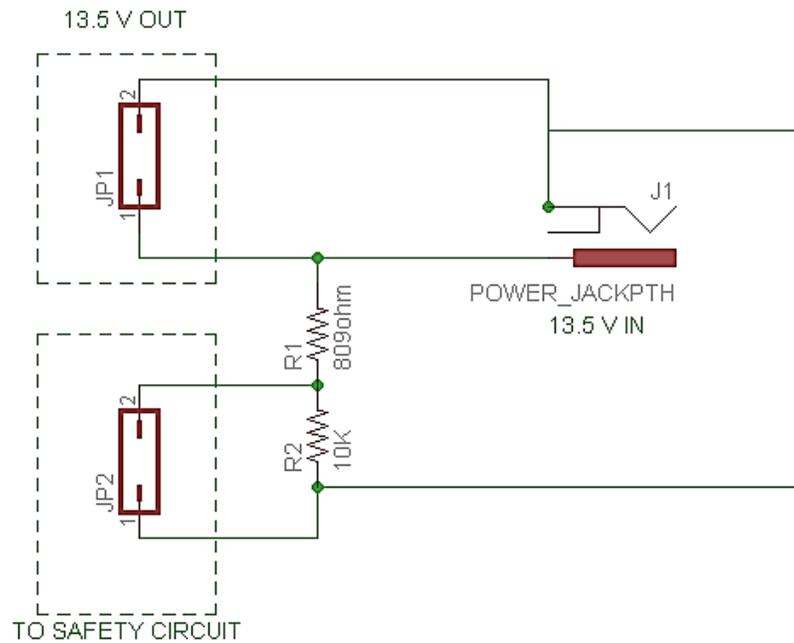
$$I1 = \frac{5}{10000} + 0.010A$$

is an approximate representation of what is happening after the voltage divider circuit. It is important to calculate the proper resistance values to ensure the correct operation of the circuit.



**Figure 4.20 Current Calculations for Voltage Detection**

R1 is calculated to be 809ohms. This will provide 0.0105A over R1, and 5 V trigger voltage in steady state operation.



**Figure 4.21 Voltage Detection Schematic**

#### 4.3.8 Cost

A cost breakdown of the electrical design component is found in Table 4.8. The first section includes parts specific to one actuator assembly. The second section in the table has parts to integrate each system together. For the final implementation, the cost will be  $2 \times \$258.71 + \$265.03 = \mathbf{\$782.45}$ . The initial budget estimate for electrical was \$600. The cost breakdown does not include the purchase of the dSPACE Box or related developmental software costs.

Electrical						
Item	Description	Supplier	Part Number	Quantity	Unit Price	Total
1	Motor	Banebots	M5-R5540-12	1	6.25	6.25
2	Gear Box	Banebots	P60K-444-0004	1	64.5	64.5
3	Motor Controller	Robotshop	RB-POL-21	1	35.66	35.66
4	and gate	Digikey	568-1505-5-ND	1	0.57	0.57
5	Fuse Holder	Digikey	F064-ND	2	3.2	6.4
6	15A Fuse	Digikey	F992-ND	6	4.23	25.38
7	5V Reg	Digikey	497-1443-5-ND	1	0.66	0.66
8	inverter	Digikey	TC74HC04APF-ND	1	0.52	0.52
9	100uF capacitor	Digikey	493-1341-ND	1	0.34	0.34
10	ATTINY	Digikey	ATTINY45-20PU-ND	1	2.54	2.54
11	10K Resistor	Digikey	PPC10.0KXCT-ND	5	0.25	1.25
12	Tactile Switch	Digikey	450-1650-ND	2	0.14	0.28
13	Female db09 cable	Digikey	AE9872-ND	1	7.85	7.85
14	DPDT Switch	Digikey	450-1629-ND	1	5.18	5.18
15	D-SUB	Digikey	A35105-ND	1	1.2	1.2
16	IC Socket	Digikey	3M5473-ND	1	0.2	0.2
17	POT	Digikey	3310Y-001-103L-ND	1	2.21	2.21
18	SPDT Switch	Digikey	EG1903-ND	1	0.8	0.8
19	Screw Terminals	Digikey	ED2600-ND	5	0.22	1.1
20	Female header 8pos	Digikey	609-3559-ND	1	1.6	1.6
21	Male header 8pos	Digikey	609-3257-ND	1	0.61	0.61
22	11mm standoff	Digikey	952-1486-ND	5	0.69	3.45
23	M3 Screws	Digikey	29311K-ND	10	0.38	3.8
24	Spade Connector	Digikey	920024-04-ND	10	0.455	4.55
25	Tab Connector	Digikey	920044-03-ND	10	0.647	6.47
26	Roller Switch	Digikey	SW840-ND	2	3.04	6.08
27	Enclosure	Digikey	HM1093-ND	1	4.93	4.93
28	8 Pin Connector Female	Digikey	708-1069-ND	1	5.5	5.5
29	Female Connector Tins	Digikey	708-1093-ND	1	11.6	11.6
30	8 Pin Plug Male	Digikey	708-1034-ND	1	6.46	6.46
31	Male Connector Tins	Digikey	708-1094-ND	1	7.29	7.29
32	3 Pin Connector Female	Digikey	708-1066-ND	1	5.5	5.5
33	Female Connector Tins	Digikey	708-1091-ND	1	12.6	12.6
34	3 Pin Plug Male	Digikey	708-1031-ND	1	6.34	6.34
35	Male Plug Tins	Digikey	708-1092-ND	1	9.04	9.04
					<b>Subtotal</b>	<b>258.71</b>
Single Parts						
Item	Description	Supplier	Part Number	Quantity	Unit Price	Total
36	Carrying Case	Digikey	SR-R720-FB-ND	1	110.29	110.29
37	8 Pin Connector Female	Digikey	708-1069-ND	1	5.5	5.5
38	Female Connector Tins	Digikey	708-1093-ND	1	11.6	11.6
39	8 Pin Plug Male	Digikey	708-1034-ND	1	6.46	6.46
40	Male Connector Tins	Digikey	708-1094-ND	1	7.29	7.29
41	3 Pin Connector Female	Digikey	708-1066-ND	1	5.5	5.5
42	Female Connector Tins	Digikey	708-1091-ND	1	12.6	12.6
43	3 Pin Plug Male	Digikey	708-1031-ND	1	6.34	6.34
44	Male Plug Tins	Digikey	708-1092-ND	1	9.04	9.04
45	db25 Panel Mount	Digikey	AE10108-ND	1	14.28	14.28
46	AC In	Digikey	486-1073-ND	1	11.8	11.8
47	Power Supply	Digikey	285-1984-ND	1	61.17	61.17
48	Zip Ties 4"	Digikey	3M101C-ND	1	3.16	3.16
					<b>Subtotal</b>	<b>265.03</b>
					<b>Total</b>	<b>523.74</b>

Table 4.8 Electrical Cost Breakdown

## 4.4 Software Design

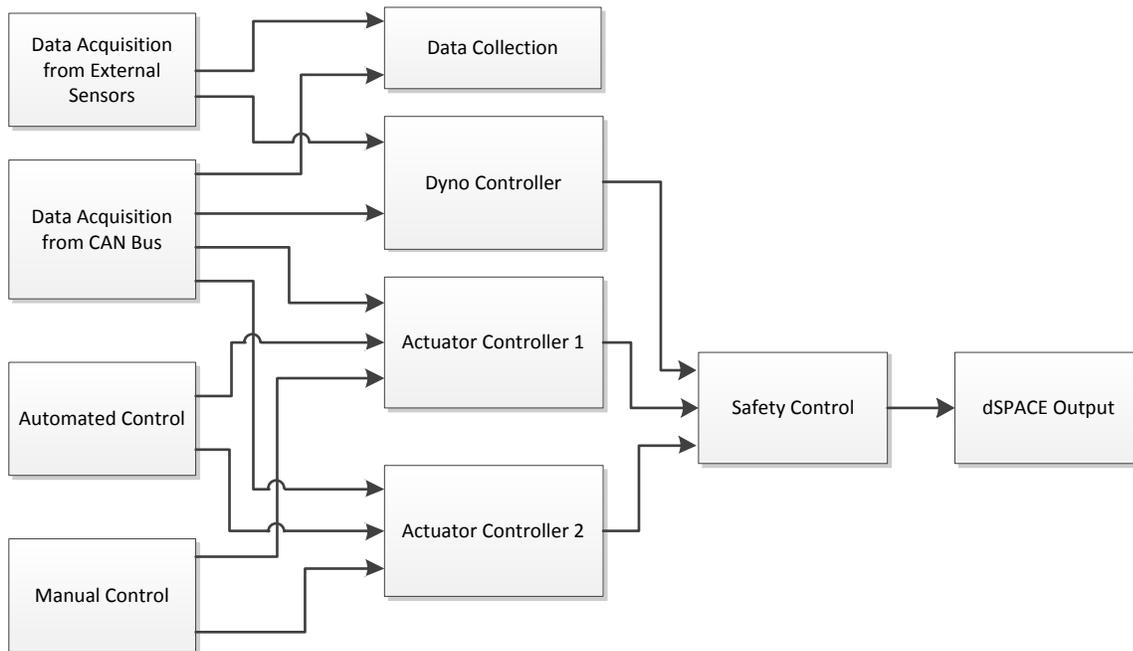
This section explains the design process behind the software implemented on the dSPACE device. The intention is to provide the required information to modify and improve current software implementations. This will however not explain the functionality of Simulink and related block sets. It is assumed knowledge for the reader and can be researched through the Matlab help menus.

### 4.4.1 Objective

The objective of the software design is to provide an organized, maintainable software layout that is easy to debug and modify. Simulink models provide a block style interface with a series of subsystems intended to reduce complexity through levels of software implementation. Sections 4.4.5 through 4.4.7 break down the implementation levels used in the Simulink model layout.

### 4.4.2 Data Flow Breakdown

Figure 4.22 shows a general overview of the software design. This provides a starting point for the software design.



**Figure 4.22 Software Data Flow Breakdown**

#### 4.4.2.1 Data Acquisition from External Sensors

The data acquisition subsystem provides an interface to any external sensors used that provide a control variable to the software. This does not include any information received from the CAN Bus interface. Examples are wind speed, angle of grade, and air density, used for equivalent resistive force calculations (2.4.1).

#### 4.4.2.2 Data Acquisition from CAN Bus

This subsystem provides a software interface to the CAN Bus messages received from the car. It provides only the needed outputs for data collection and system control. This module only monitors the CAN Bus interface and does not directly write anything that may interfere with the Vehicle's operation.

#### 4.4.2.3 Automated Control

The automated control subsystem includes preloaded or preset conditions that determine how the vehicle will run. The simplest implementation has velocity vs. time values that the vehicle follows to complete a set of tests. This module includes functionality for selecting or loading driving schedules, a user input to enable and disable the automated control and be designed for future modifications.

#### 4.4.2.4 Manual Control

This subsystem is used to transform user interface commands to actuator motion. Signals from this module are the same as the automated control module and it provides a controlled way to test other modules.

#### 4.4.2.5 Data Collection

The data collection subsystem receives monitored and controlled variable states and displays them in the user interface. This subsystem also records selected variable values for data analysis, and replaying custom automated driving cycles.

#### 4.4.2.6 Dyno Controller

The dyno controller subsystem gives commands to the dynamometer to adjust the resistance for driving simulation. This is controlled by the automated driving mode and is governed by feedback from the CAN Bus module and external sensor. Due to time and availability of the Dynamometer, this subsystem is a future consideration for this overall system.

#### 4.4.2.7 Actuator Controller 1 & 2

This subsystem is intended to translate requested position and speed values into speed and direction control outputs to send to a DC motor controller circuit. Two cascading PI closed-loop controllers are implemented in this module. This controller layout is based on the proven example presented in “Development of a Robotic Driver for Vehicle Dynamometer Testing” (Namik, et al.). Safety limits specific to the actuator and hardware being used are incorporated to prevent undesired operation. This subsystem also allows the PI parameters to be externally adjusted from the user interface. The accelerator and brake controllers are similar in principle but incorporate different scaling factors specific to each feedback signal. The two controllers need to communicate with one another to prevent opposing behavior in controlling the vehicle.

#### 4.4.2.8 Safety Control

The safety control subsystem incorporates an emergency stop feature and also inspects values before they are passed to the digital output blocksets. This prevents erratic behavior caused by unforeseen errors during testing of the controllers. The safety control also provides added software features, such as limit switches and E-Stop button.

#### 4.4.2.9 dSPACE Output

The dSPACE output subsystem contains the output blocks to produce electrical signals to the motor controller circuitry. Using block sets provided by dSPACE, the correct output pins are mapped to the hardware control logic. This also contains the software limit switch detection and prevents control signals from being passed when a

limit switch is triggered. This provides a backup to the logic gates implemented in hardware (see section 4.3.7.3).

### 4.4.3 User Interface Design

The user interface will display basic information about the vehicles operation to a remote terminal to monitor variables in real-time. This gives the user additional information about the vehicles operation during testing. The interface can be customized depending on the testing being performed. The information in Table 4.9 and user control in Table 4.10 should be in each interface to meet software requirements.

#### 4.4.3.1 Required Display Information

The following display information is required in the user interface to ensure a safe and complete implementation.

<b>Variable</b>	<b>Suggested Display type</b>	<b>Units</b>
Vehicle Speed	Analog & Digital	Km/h
Engine Speed	Analog & Digital	RPM
Ignition Status	LED	On/Off
Fuel Level	Analog	%
Battery SOC	Digital	Magnitude
Coolant Temp	Digital	Degrees Celsius
Gas Pedal Position	Analog	%
Brake Torque Requested	Analog	N-m
Estop	LED	On/Off
FWD Limit	LED	On/Off
REV Limit	LED	On/Off
Testing	LED	On/Off

**Table 4.9 Required Display Information**

#### 4.4.3.2 Required User Control

The following user controls are required in the user interface to be functionally complete.

<b>Controlled Variable</b>	<b>Suggested Interface Type</b>
FWD Select	Momentary Button Press
REV Select	Momentary Button Press
Duty % Select	Knob Select
Pedal Control Select	Radio Button
E-Stop	On Button
E-Stop Reset	Off Button
Gas Pedal Position	Slider
Brake Force	Slider
Rolling Resistance	Slider
Controller 1 P Value	Knob Select
Controller 1 I Value	Knob Select
Controller 2 P Value	Knob Select
Controller 2 I Value	Knob Select
Mode Select	Radio Button
Driving Test Select	Drop Down Menu
Test Start	On Button
Test Stop	Off Button

**Table 4.10 User Control Required**

#### 4.4.4 Implementation Level 1 – System\_Testing.mdl

The software created in Simulink has different levels of implementation. This is done to make the software more maintainable. The first level contains the highest-level blocks, which contain all the lower level code. It can be looked at as a more general layout and its design is based on the Module breakdown in section 4.4.2. The following module guides provide descriptions of signals and show how the different modules communicate with one another.

##### Module Guide

Description	High-level model space containing user controlled variables and subsystems, making up the control software.
Secret	N/A
Requirements	3.4.1, 3.4.2

##### Module Design

Interface	Name	Description
Incoming	None	
Outgoing	None	
Uses	CAN Bus Signals	Decodes CAN messages from vehicle
	Encoder Feedback	Returns encoder position value
	Mode Selection	Used to change operating modes
	Source Selection	Select automated input test data
	Source Control	Switch between auto and manual requests
	Actuator Control Gas	Create control signal for gas actuator
	Actuator Control Brake	Create control signal for brake actuator
	Output Control Accel	Output mode switch for gas actuator
	Output Control Brake	Output mode switch for brake actuator
	Manual Control	Manual control output

	Error Output		Outputs controller errors when in auto mode
	E-Stop		Logical operators for E-stop conditioning
	dSPACE to Accelerator		Controls output ports for dSPACE to hardware
	dSPACE to Brake		Controls output ports for dSPACE to hardware
<b>Objects</b>	<b>Name</b>	<b>Type</b>	<b>Description</b>
Constants	None		
Monitored	Speed Error Gas	Real	actuator speed error (actual-desired)
	Position Error Gas	Real	actuator position error (actual-desired)
	Speed Error Brake	Real	actuator speed error (actual-desired)
	Position Error Brake	Real	actuator position error (actual-desired)
	Vehicle_Speed	Real	Vehicle speed in km/h from CAN Bus
	Temp_Eng_Coolant	Real	Engine temp. reading from CAN Bus
	Ignition_Status	Int	Ignition position from CAN Bus
	Meas_SOC	Real	Battery state of charge from CAN Bus
	Yaw_Angle_Rad	Real	Car YAW angle in rads from CAN Bus
	Fuel_Econ	Real	Fuel economy in L/100km from CAN
	Outside_Temp	Real	Outside temp from CAN Bus
	Vehicle_Yaw	Real	Car YAW angle in rad/s from CAN Bus
	Eng_Speed	Real	RPM of gas engine from CAN Bus
	Fuel_Level	Percent	Percentage of fuel remaining from CAN
	Distance_Total	Real	Odometer reading from CAN Bus
	Gas_Pos	Percent	Gas pedal position from CAN Bus
	Brake_Pos	Real	Brake requested torque from CAN Bus
	RX_Time	Real	dSPACE internal time from power on
	Accel\Limit Fwd	Bool	Accelerator actuator fwd limit status
	Accel\Limit Rev	Bool	Accelerator actuator rev limit status
	Accel\Dig in1	Bool	Digital pin status (optional encoder)
	Accel\Dig in2	Bool	Digital pin status (optional encoder)
	Accel\Current Sense	Real	Accelerator actuator current reading

---

	Brake\Limit Fwd	Bool	Brake actuator fwd limit status
	Brake\Limit Rev	Bool	Brake actuator rev limit status
	Brake\Dig in1	Bool	Digital pin status (optional encoder)
	Brake\Dig in2	Bool	Digital pin status (optional encoder)
	Brake\Current Sense	Real	Brake actuator current reading
Controlled	Encoder Reset	Bool	Command to reset encoder counter
	Mode Select	Int	User input for testing mode selection
	Source Select	Int	User select for driving schedule
	Source Man En	Bool	Enable manual control of speed or pos
	Software_Estop	Bool	Turn off external commands to motors

**Table 4.11 System Testing.mdl Description**



## 4.4.5 Implementation Level 2

This section gives a description of modules used by Implementation Level 1. This is intended to be a reference for software maintenance.

### 4.4.5.1 System\_Testing\CAN Bus Signals

#### Module Guide

Description	Contains blocks to receive and interpret messages received by the dSPACE CAN controllers. Medium and High speed CAN messages are received from the vehicles diagnostic port.
Secret	Hide message configurations required to interpret Ford Escape Hybrid CAN Bus communication lines
Requirements	Interface to Vehicle, 3.4.1

#### Module Design

Interface	Name	Description
Incoming	None	
Outgoing	Gas_Pos	Gas pedal position from CAN Bus
	Brake_Pos	Brake requested torque from CAN Bus
	Vehicle_Speed	Vehicle speed in km/h from CAN Bus
	Temp_Eng_Coolant	Engine temp. reading from CAN Bus
	Ignition_Status	Ignition position from CAN Bus
	Meas_SOC	Battery state of charge from CAN Bus
	Yaw_Angle_Rad	Car YAW angle in rads from CAN Bus
	Fuel_Econ	Fuel economy in L/100km from CAN
	Outside_Temp	Outside temp from CAN Bus
	Vehicle_Yaw	Car YAW angle in rad/s from CAN Bus
	Eng_Speed	RPM of gas engine from CAN Bus
	Fuel_Level	Percentage of fuel remaining from CAN
	Distance_Total	Odometer reading from CAN Bus

	RX_Time	dSPACE internal time from power on
Uses	HS_CAN_Setup	Used to setup HS_CAN input from OBDII port
	CAN_Type1_Setup	Used to setup MS_CAN input from OBDII port
	Ignition_Switch Position	Message block for ignition switch information
	Engine_Data	Message block for engine information
	HS_BSC_Control_1	Message block for brake controller
	HS_PCM_Eng_OBDII	Message block for Throttle Position information
	YAW	Message block for vehicle accelerometer outputs
	HS_TBCM_State_2	Message block for hybrid battery information
	Fuel_Economy	Message block for vehicle calculated fuel econ
	Cluster_Information	Message block for dashboard information values
	Cluster_Information_4	Message block for dashboard information values
	Integrator_Limited	Integrator block used to convert yaw accel to pos
Objects	Name	Description
Constants	Constant	Offset for calculating accelerator position
	Gain	Constant gain for calculating accelerator position
	Gain1	Constant gain for calculating accelerator position
Monitored	None	
Controlled	None	

**Table 4.12 CAN Bus Signal Module Guide/Design**

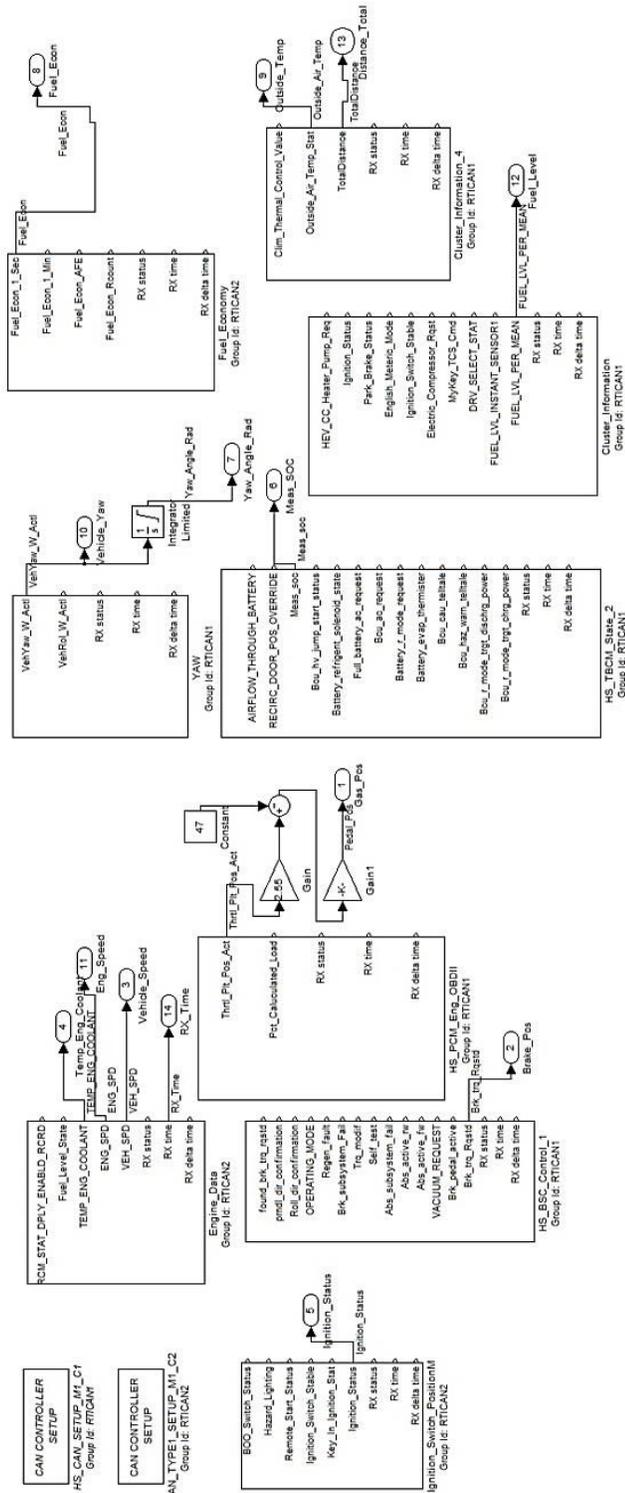


Figure 4.24 CAN Bus Signals

#### 4.4.5.2 System\_Testing\Encoder Feedback

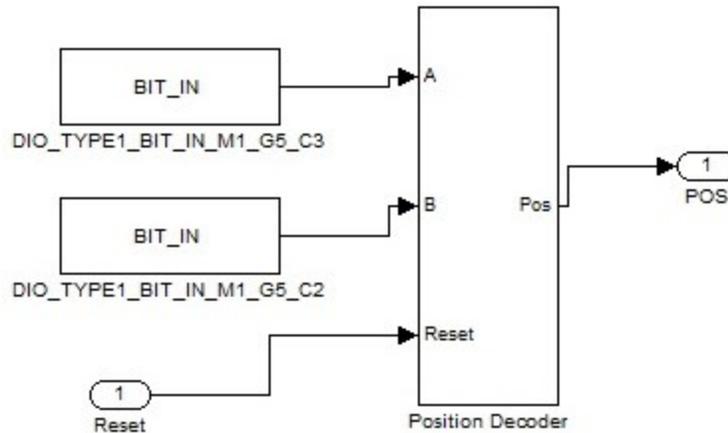
##### Module Guide

Description	This module takes in 2 encoder channels and outputs a position. The accumulated position is able to be reset by the user.
Secret	Hides the math behind turning a sequence of pulses into a relative position output.
Requirements	none

##### Module Design

Interface	Name	Description
Incoming	Bit_In A	Digital signal from encoder channel A
	Bit_In B	Digital signal from encoder channel B
	Reset	User Input from System_Testing
Outgoing	POS	Distance of actuator travel (%) 100% = 2.5"
Uses	Position Decoder	4.4.6.4
Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	N/A	

**Table 4.13 Encoder Feedback Module Guide/Design**



**Figure 4.25 Encoder Feedback**

#### 4.4.5.3 System\_Testing\Mode Selection

##### Module Guide

**Description** This module is used to read an integer input corresponding to a mode and set the corresponding state variables to activate signal outputs in the program.

**Secret** Hide the decision statement to activate and deactivate each state variable.

**Requirements** 4.4.3.2

##### Module Design

Interface	Name	Description
Incoming	Mode Select	Mode 0 to Mode 9, input from user
	Position Feedback Car Gas	Position of accelerator pedal from CAN Bus
	Encoder Feedback	Position of encoder
	Position Feedback Car Brake	Position of brake pedal from CAN Bus
	Manual Mode Enable	Bit output to turn on manual mode
Outgoing	Gas Actuator Enable	Bit output to turn on gas actuator motion
	Brake Actuator Enable	Bit output to turn on brake actuator motion

	Position Source	Bit output to turn on feedback for controller
	Activate	Throughput for position feedback selection
	Position Feedback Gas	Throughput for position feedback selection
	Position Feedback	
	Brake	
Uses	N/A	
Objects	Name	Description
Constants	Gain	Scalar to convert encoder value for brake
Monitored	N/A	
Controlled	GasEnable	Enable operation to accelerator actuator
	BrakeEnable	Enable operation to brake actuator
	EncoderEnable	Enable Encoder as position feedback
	PositionSourceActivate	Select between external input or controller
	ManualModeEnable	Turn on manual input for actuators
Mode 0: Idle		No movement or control of the actuators
Mode 1: Manual Accel		Manual control for Accelerator Actuator
Mode 2: Manual Brake		Manual control for Brake Actuator
Mode 3: Position w/Encoder Gas		Position controller test with encoder feedback on accelerator actuator
Mode 4: Position w/Encoder Brake		Position controller test with encoder feedback on brake actuator
Mode 5: Position w/Car Gas		Position test with car feedback - Accelerator
Mode 6: Position w/Car Brake		Position test with car feedback – Brake
Mode 7: Speed Controller Test Gas		Speed controller test Accelerator
Mode 8: Speed Controller Test Brake		Speed controller test Brake
Mode 9: Full System Test		Full System test – Both actuators active

**Table 4.14: Mode Selection Module Guide/Design**

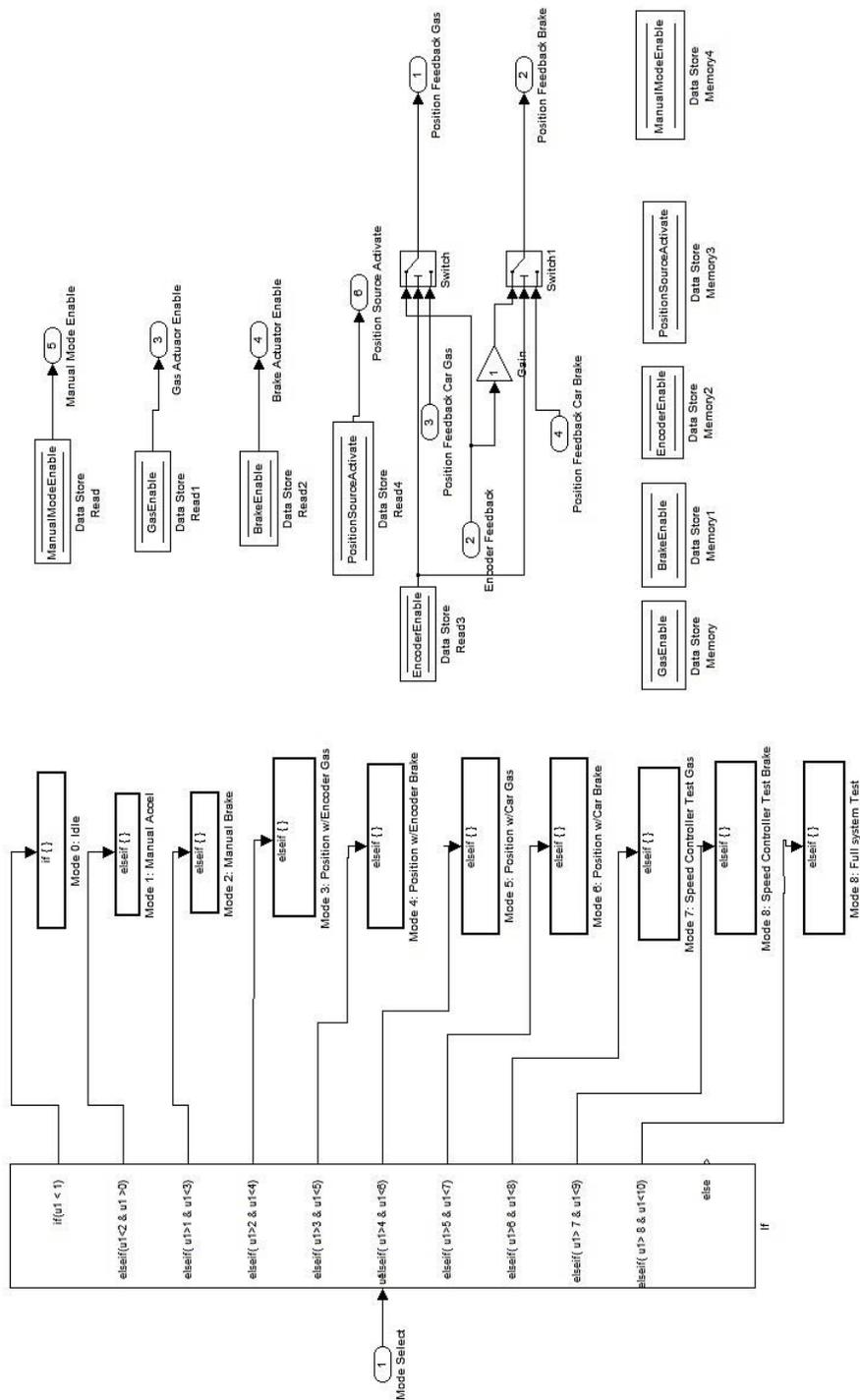


Figure 4.26 Mode Selection

4.4.5.4 System\_Testing\Source Selection

**Module Guide**

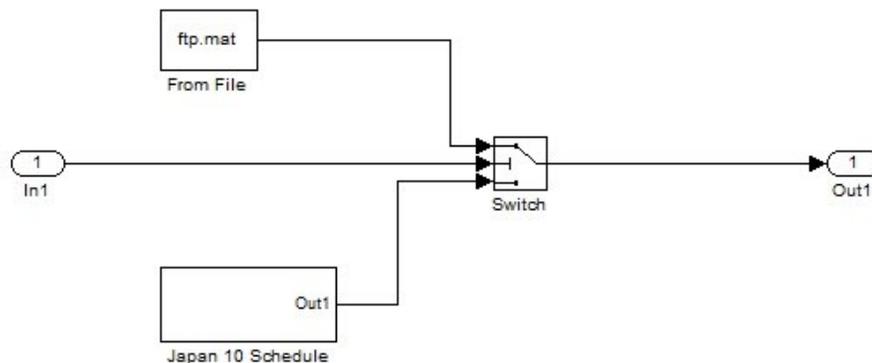
Description	Select between driving schedules pre-loaded into the software.
Secret	Hide the selection and source of the loaded driving schedules to add and modify in the future.
Requirements	4.4.3.2

**Module Design**

Interface	Name	Description
Incoming	In1	Bit input to select between 2 sources
Outgoing	Out1	Time series vector of driving schedule
Uses	From File	Load a .mat file from dSPACE memory
	Japan 10 Schedule	Combination of Matlab signal generations to Produce the driving schedule

Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	Switch	On/Off to change source selection

**Table 4.15 Source Selection Module Guide/Design**



**Figure 4.27 Source Selection**

## 4.4.5.5 System\_Testing\Source Control

**Module Guide**

**Description** This module switches between manual speed/position request and external source. The external source would be the preprogrammed driving schedule selected using source selection module 4.4.5.4.

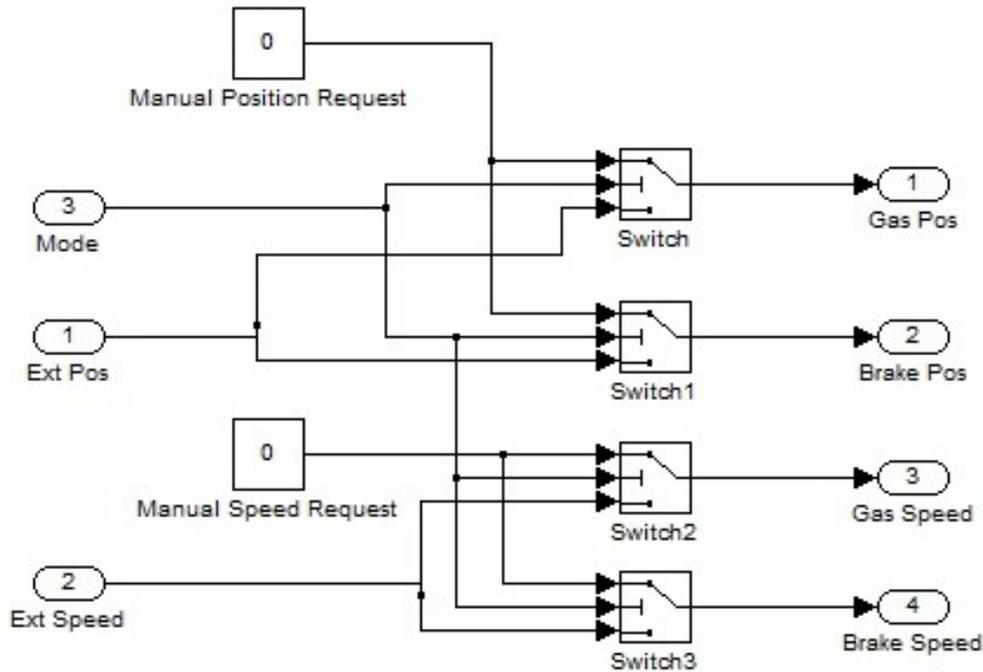
**Secret** Hides the switching control between manual and external inputs for the actuator controllers. Also routes sources to position and speed request outputs for the corresponding mode selected in Mode Select 4.4.5.3.

**Requirements** 4.4.3.2

**Module Design**

Interface	Name	Description
Incoming	Mode	User input to select between manual and ext
	Ext Pos	Time series vector from Source Select
	Ext Speed	Time series vector from Source Select
Outgoing	Gas Pos	Result for selected gas position request
	Brake Pos	Result for selected brake position request
	Gas Speed	Result for selected gas speed request
	Brake Speed	Result for selected brake speed request
Uses	N/A	
Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	Manual Speed Request	User input from interface to set position
	Manual Position Request	User input from interface to set position

**Table 4.16 Source Control Module Guide/Design**



**Figure 4.28 Source Control**

#### 4.4.5.6 System\_Testing\Actuator Control Gas

##### Module Guide

**Description** This module will provide control signals for the accelerator motor controller based on feedback information received. It is comprised of 2 PI controllers. One for speed control and one for position.

**Secret** Hide the control feedback loop for operating the accelerator actuator. Provides safety elements to prevent erratic behavior.

**Requirements** 4.4.2.7

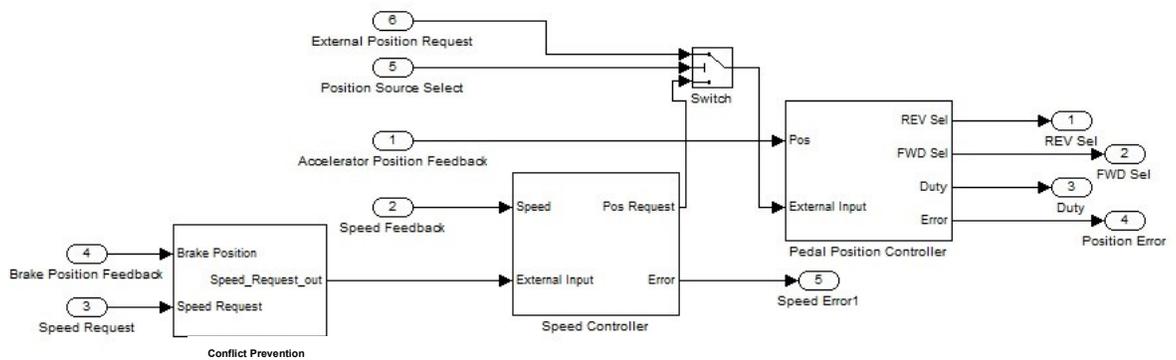
##### Module Design

Interface	Name	Description
Incoming	Accelerator Position Feedback	Measured pedal position (%)
	Speed Feedback	Measured Vehicle Speed (km/h)
	Speed Request	Requested vehicle speed to follow

	Brake Position Feedback	Measured brake position
	Position Source Select	Bit Trigger to change to external position req
	External Position Request	Requested position input to follow (%)
Outgoing	REV Sel	Bit output to drive actuator in reverse
	FWD Sel	Bit output to drive actuator forward
	Duty	Duty % of PWM sent to motor
	Position Error	Actual position compared to requested
	Speed Error	Actual speed compared to requested
Uses	Conflict Prevention	4.4.6.3
	Speed Controller	4.4.6.2
	Pedal Position Controller	4.4.6.1

Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	N/A	

**Table 4.17 Actuator Control Gas Module Guide/Design**



**Figure 4.29 Actuator Control Gas**

## 4.4.5.7 System\_Testing\Actuator Control Brake

**Module Guide**

**Description** This module will provide control signals for the brake motor controller based on feedback information received. It is comprised of 2 PI controllers. One for speed control and one for position.

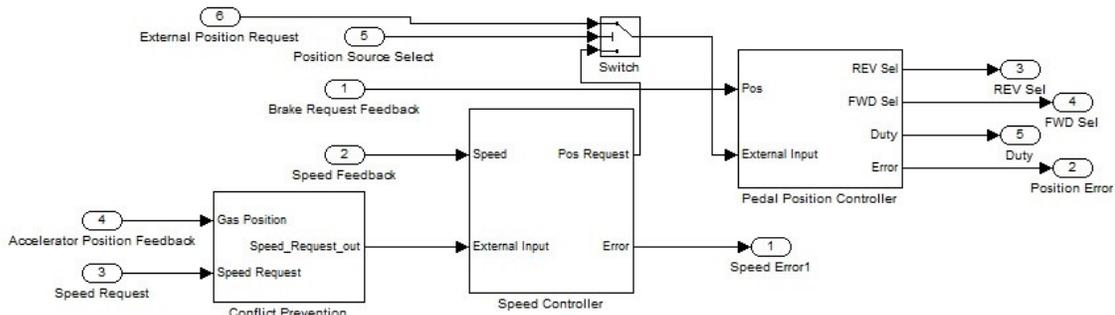
**Secret** Hide the control feedback loop for operating the brake actuator. Provides safety elements to prevent erratic behavior.

**Requirements** 4.4.2.7

**Module Design**

Interface	Name	Description
Incoming	Accelerator Position Feedback	Measured pedal position (%)
	Speed Feedback	Measured Vehicle Speed (km/h)
	Speed Request	Requested vehicle speed to follow
	Brake Position Feedback	Measured brake position
	Position Source Select	Bit Trigger to change to external position req
	External Position Request	Requested position input to follow (%)
Outgoing	REV Sel	Bit output to drive actuator in reverse
	FWD Sel	Bit output to drive actuator forward
	Duty	Duty % of PWM sent to motor
	Position Error	Actual position compared to requested
	Speed Error	Actual speed compared to requested
Uses	Conflict Prevention	Similar to gas actuator controller 4.4.6.3
	Speed Controller	Similar to gas actuator controller 4.4.6.2
	Pedal Position Controller	Similar to gas actuator controller 4.4.6.1
Objects	Name	Description
Constants	N/A	
Monitored	N/A	

**Table 4.18 Actuator Control Brake Module Guide/Design**



**Figure 4.30 Actuator Control Brake**

#### 4.4.5.8 System\_Testing\Output Control Accel

##### Module Guide

**Description** This module connects signals from the gas motor controller 4.4.5.6 and manual control 4.4.5.10 modules and will simultaneously switch motor signal output sources depending on the selected mode. Also provides a software override to stop the actuator.

**Secret** This hides the software behavior of switching between controller and manual mode, and also contains default actuator output stop settings for software emergency stop function.

**Requirements** None

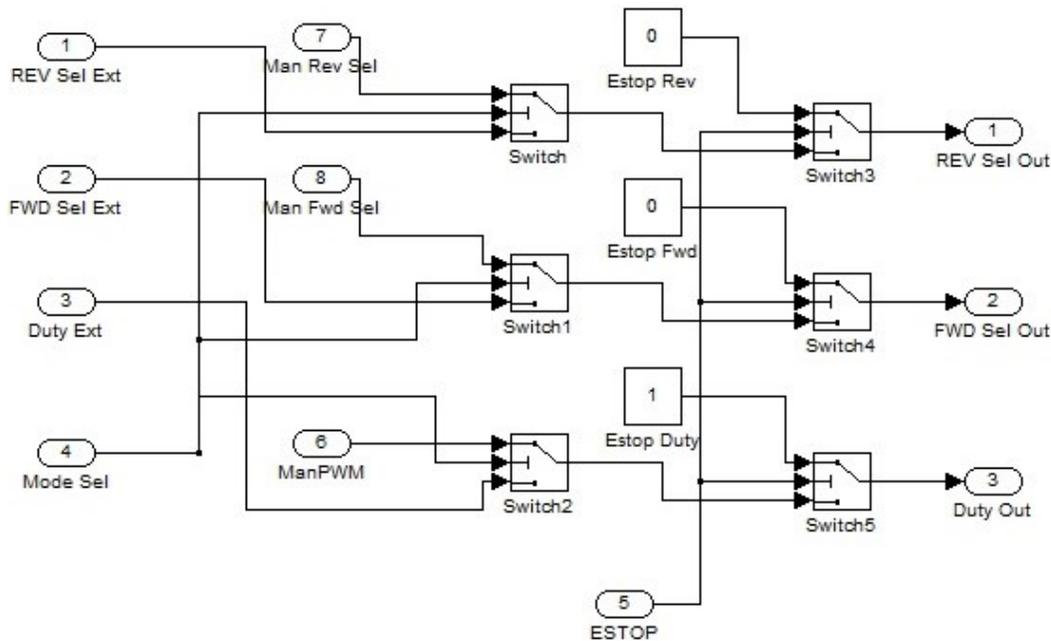
##### Module Design

Interface	Name	Description
Incoming	REV Sel Ext	Enable reverse direction
	FWD Sel Ext	Enable forward direction
	Duty Ext	Duty value for PWM controller in percentage
	Mode Sel	User controlled to switch to manual mode
	ESTOP	Switches outputs to default motor stopped values
	Man Rev Sel	Input from manual controller to select reverse
	Man Fwd Sel	Input from manual controller to select forward
	ManPWM	Input from manual controller to set Duty percent

Outgoing	REV Sel Out	Signal to motor controller to set reverse direction
	FWD Sel Out	Signal to motor controller to set forward direction
	Duty Out	Signal to motor controller to set duty percentage
Uses	Switch	Matlab function to switch between signal inputs using a binary input signal

Objects	Name	Description
Constants	Estop Rev	Value to disable reverse mode
	Estop Fwd	Value to disable forward mode
	Estop Duty	Value to select 0% duty output
Monitored	N/A	
Controlled	N/A	

**Table 4.19 Output Control Accel Module Guide/Design**



**Figure 4.31 Output Control Accel**

#### 4.4.5.9 System\_Testing\Output Control Brake

##### Module Guide

**Description** This module connects signals from the brake motor controller 4.4.5.7 and manual control 4.4.5.10 modules and will simultaneously switch motor signal output sources depending on the selected mode. Also provides a software override to stop the actuator.

**Secret** This hides the software behavior of switching between controller and manual mode, and also contains default actuator output stop settings for software emergency stop function.

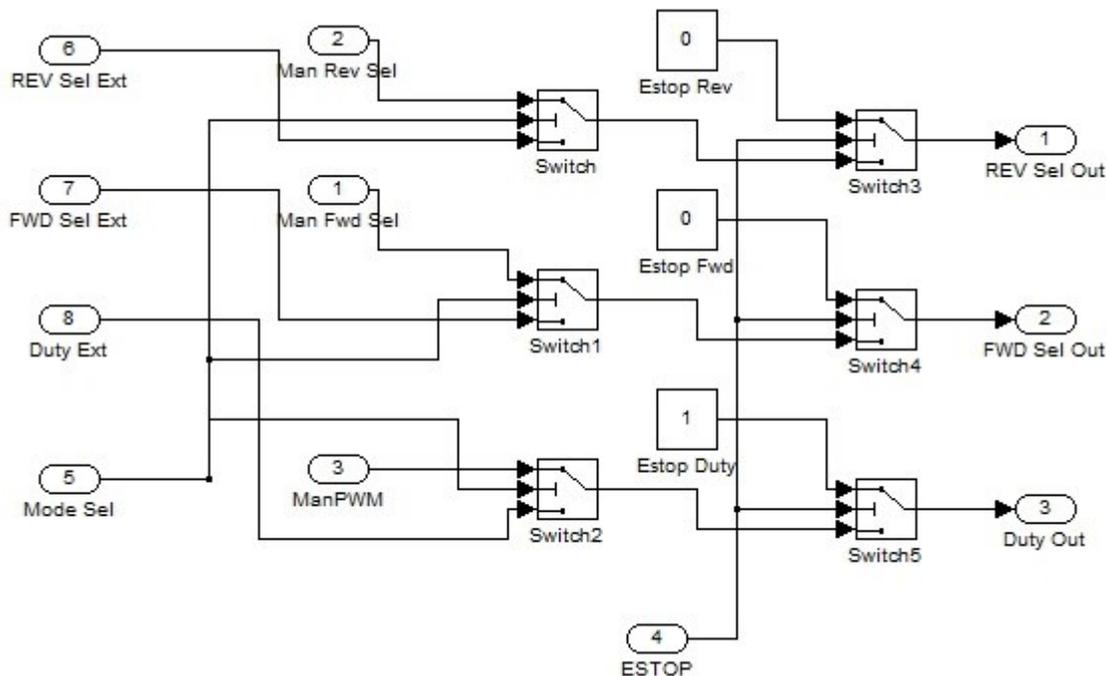
**Requirements** None

##### Module Design

Interface	Name	Description
Incoming	REV Sel Ext	Enable reverse direction
	FWD Sel Ext	Enable forward direction
	Duty Ext	Duty value for PWM controller in percentage
	Mode Sel	User controlled to switch to manual mode
	ESTOP	Switches outputs to default motor stopped values
	Man Rev Sel	Input from manual controller to select reverse
	Man Fwd Sel	Input from manual controller to select forward
	ManPWM	Input from manual controller to set Duty percent
Outgoing	REV Sel Out	Signal to motor controller to set reverse direction
	FWD Sel Out	Signal to motor controller to set forward direction
	Duty Out	Signal to motor controller to set duty percentage
Uses	Switch	Matlab function to switch between signal inputs using a binary input signal
<b>Objects</b>	<b>Name</b>	<b>Description</b>
Constants	Estop Rev	Value to disable reverse mode

	Estop Fwd	Value to disable forward mode
	Estop Duty	Value to select 0% duty output
Monitored	N/A	
Controlled	N/A	

**Table 4.20 Output Control Brake Module Guide/Design**



**Figure 4.32 Output Control Brake**

4.4.5.10 System\_Testing\Manual Control

**Module Guide**

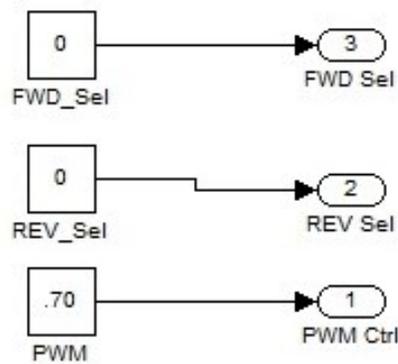
Description This module provides 3 user inputs to control a single actuator manually. The actuator being controlled is selected in the Mode Selection Module (4.4.5.3).

Secret Contains default values for motor speed and direction selection that is controllable through the user interface (4.4.8).

Requirements 3.3.2, 4.4.3.2

**Module Design**

Interface	Name	Description
Incoming	N/A	
Outgoing	FWD Sel	Enables motor controller for forward motion
	REV Sel	Enables motor controller for reverse motion
	PWM Ctrl	Sets PWM duty percentage for speed
Uses	N/A	
Objects	Name	Description
Constants	FWD_Sel	Default value to disable forward motion
	REV_Sel	Default value to disable reverse motion
	PWM	Default PWM value at .70
Monitored	N/A	
Controlled	FWD_Sel	User defined value in user interface
	Rev_Sel	User defined value in user interface
	PWM	User defined value in user interface

**Table 4.21 Manual Control Module Guide/Design****Figure 4.33 Manual Control**

## 4.4.5.11 System\_Testing\Error Output

**Module Guide**

Description Takes in all the different error values associated with each actuator controller and outputs them when the mode isn't in manual control mode.

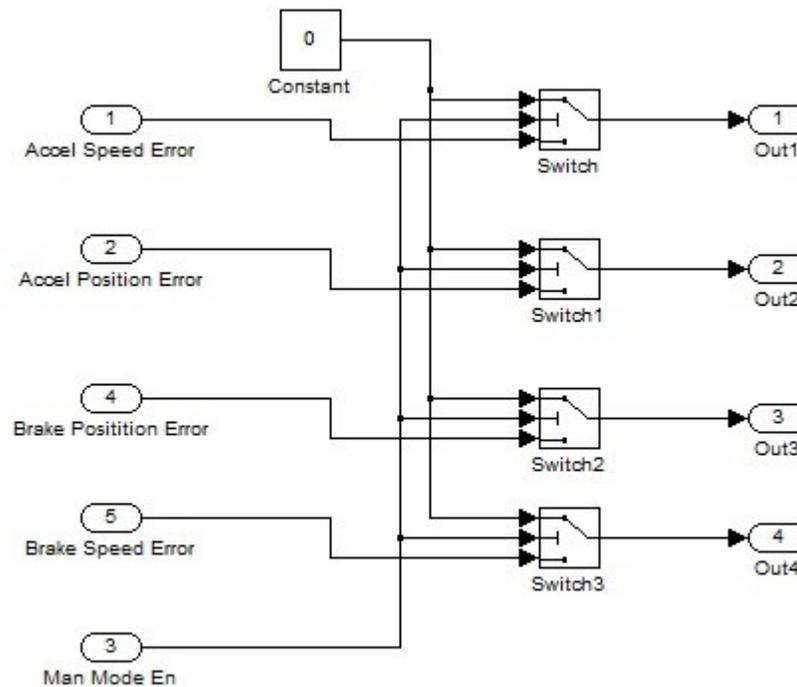
Secret Hides the enabling and disabling of the monitored error variables which get displayed in the user interface.

Requirements None

**Module Design**

Interface	Name	Description
Incoming	Accel Speed Error	Speed error from Accelerator controller 4.4.5.6
	Accel Position Error	Position error from Accelerator controller 4.4.5.6
	Brake Speed Error	Speed error from Brake Controller 4.4.5.7
	Brake Position Error	Position error from Brake Controller 4.4.5.7
	Man Mode En	Manual mode enable from Mode Select 4.4.5.3
Outgoing	Out1	Throughput of Accel Speed Error
	Out2	Throughput of Accel Position Error
	Out3	Throughput of Brake Position Error
	Out4	Throughput of Brake Speed Error
Uses	Switch	Matlab function to switch between signal inputs using a binary input signal
Objects	Name	Description
Constants	Constant	Displayed value for all errors when in manual mode
Monitored	N/A	
Controlled	N/A	

**Table 4.22 Error Output Module Guide/Design**



**Figure 4.34 Error Output**

#### 4.4.5.12 System\_Testing\E-Stop

##### Module Guide

**Description** This module provides the Estop input to the Output Control Accel (4.4.5.8) and Output Control Brake (4.4.5.9) modules. The mode selection module will trigger an Estop command to the specific actuator if it has been disabled.

**Secret** Hides the logic associated with controlling the Estop commands.

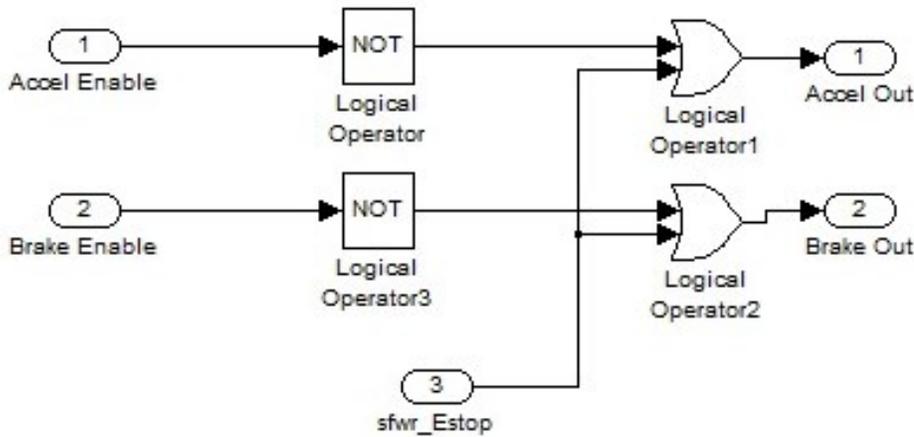
**Requirements** 4.4.3.2, 3.3.1

##### Module Design

Interface	Name	Description
Incoming	Accel Enable	Logic high turns on accelerator actuator
	Brake Enable	Logic high turns on brake actuator
	sfwr_Estop	User controlled; Logic high causes an Estop

Outgoing	Accel Out	Estop command to Output control Accel
	Brake Out	Estop command to Output Control Brake
Uses	Logic NOT	Matlab logic function
	Logic OR	Matlab logic function
Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	N/A	

**Table 4.23 E-Stop Module Guide/Design**



**Figure 4.35 E-Stop**

4.4.5.13 System\_Testing\dSPACE to Accelerator

**Module Guide**

Description This module takes in commands to control the accelerator actuator. This module also provides a software limit switch detection that turns off the actuators when detected.

Secret Hides the blockset configurations needed to connect software control signals to pin outs on the dSPACE MicroAutobox.

Requirements 3.4.2

**Module Design**

Interface	Name	Description	
Incoming	FWD Sel	Signal to enable forward motion	
	REV Sel	Signal to enable reverse motion	
	Duty Cycle	PWM duty percentage	
Outgoing	Limit Fwd	Forward limit switch value from dSPACE	
	Limit Rev	Reverse limit switch value from dSPACE	
	Dig in 1	Digital in value from dSPACE (opt. encoder)	
	Dig in 2	Digital in value from dSPACE (opt. encoder)	
	Current Sense	Scaled current sense value from dSPACE	
	Uses	Switch	Matlab function to select between 2 sources
		Logic AND	Matlab function for logic AND
Logic OR		Matlab function for logic OR	
dSPACE PWM		dSPACE block to generate PWM signal	
dSPACE Bit_Out		dSPACE block to connect to digital out pin	
dSPACE Bit_IN		dSPACE block to connect to digital in pin	
dSPACE ADC		dSPACE block to connect to analog input	
Objects	Name	Description	
Constants	FWD	Value to enable forward direction	
	REV	Value to enable reverse direction	
	Gain	Scalar to convert analog input to amperage	
	20kHz Frequency	Value to create a 20kHz frequency output	
	Off State	Value to turn off PWM output	
Monitored	N/A		
Controlled	N/A		

**Table 4.24 dSPACE to Accelerator Module Guide/Design**

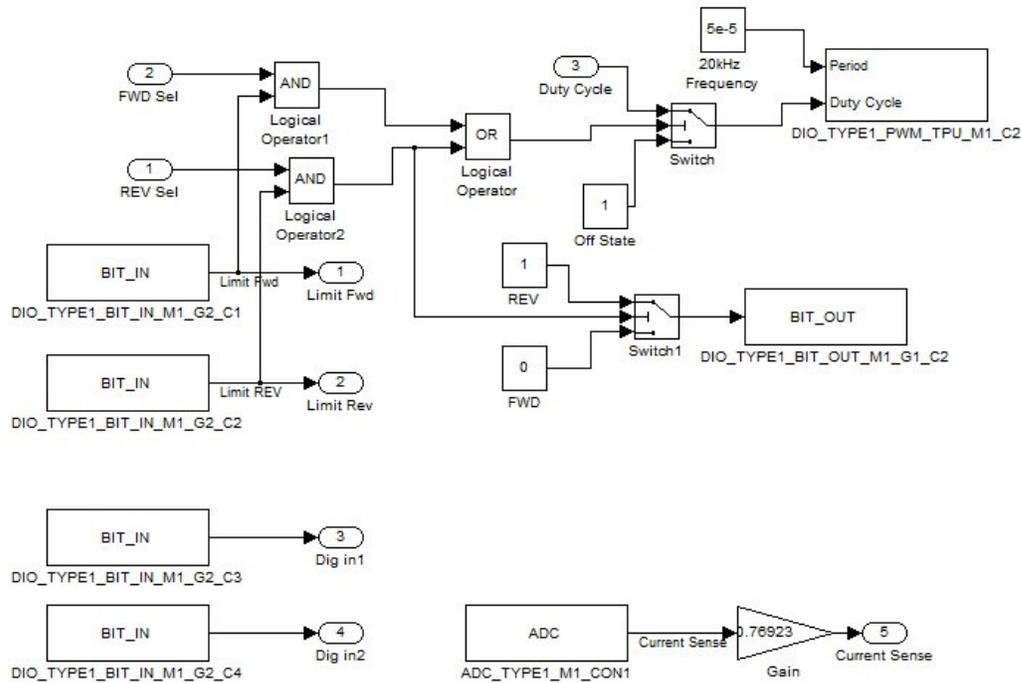


Figure 4.36 dSPACE to Accelerator

#### 4.4.5.14 System\_Testing\dSPACE to Brake

##### Module Guide

**Description** This module takes in commands to control the brake actuator. This module also provides a software limit switch detection that turns off the actuators when detected.

**Secret** Hides the blockset configurations needed to connect software control signals to pin outs on the dSPACE MicroAutobox.

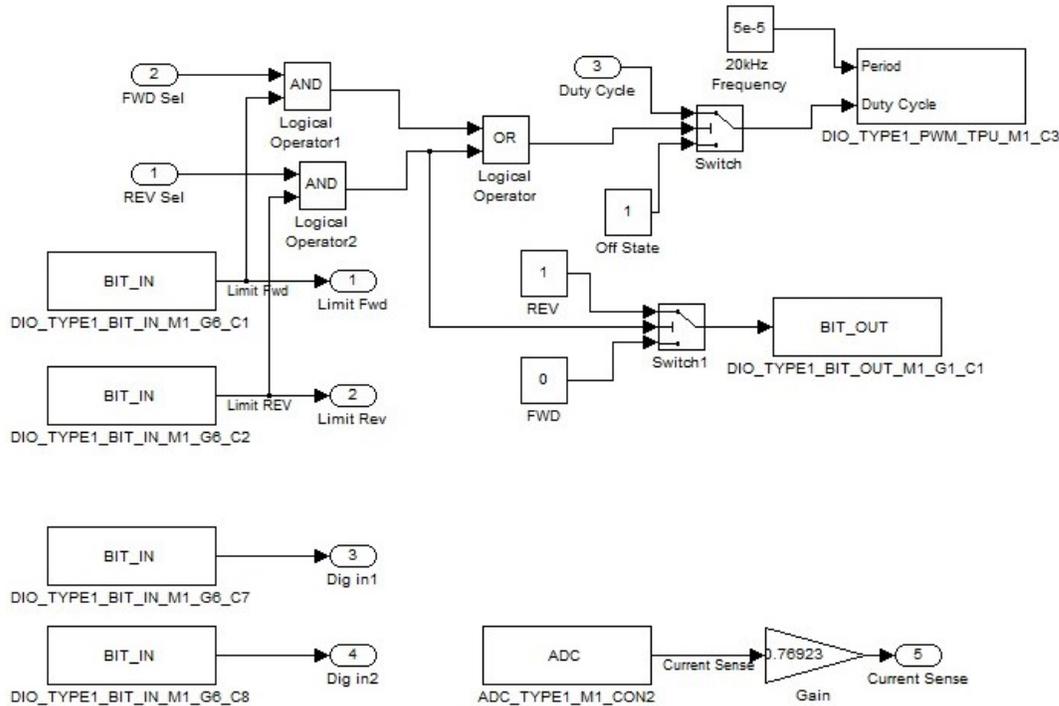
**Requirements** 3.4.2

##### Module Design

Interface	Name	Description
Incoming	FWD Sel	Signal to enable forward motion
	REV Sel	Signal to enable reverse motion
	Duty Cycle	PWM duty percentage

Outgoing	Limit Fwd	Forward limit switch value from dSPACE	
	Limit Rev	Reverse limit switch value from dSPACE	
	Dig in 1	Digital in value from dSPACE (opt. encoder)	
	Dig in 2	Digital in value from dSPACE (opt. encoder)	
	Current Sense	Scaled current sense value from dSPACE	
	Uses	Switch	Matlab function to select between 2 sources
		Logic AND	Matlab function for logic AND
		Logic OR	Matlab function for logic OR
		dSPACE PWM	dSPACE block to generate PWM signal
		dSPACE Bit_Out	dSPACE block to connect to digital out pin
	dSPACE Bit_IN	dSPACE block to connect to digital in pin	
	dSPACE ADC	dSPACE block to connect to analog input	
Objects	Name	Description	
Constants	FWD	Value to enable forward direction	
	REV	Value to enable reverse direction	
	Gain	Scalar to convert analog input to amperage	
	20kHz Frequency	Value to create a 20kHz frequency output	
	Off State	Value to turn off PWM output	
Monitored	N/A		
Controlled	N/A		

**Table 4.25 dSPACE to Brake Module Guide/Design**



**Figure 4.37 dSPACE to Brake**

#### 4.4.6 Implementation Level 3

This section gives a description of modules used by Implementation Level 2. This is intended to be a reference for software maintenance.

##### 4.4.6.1 Actuator Control Gas\Pedal Position Controller

###### Module Guide

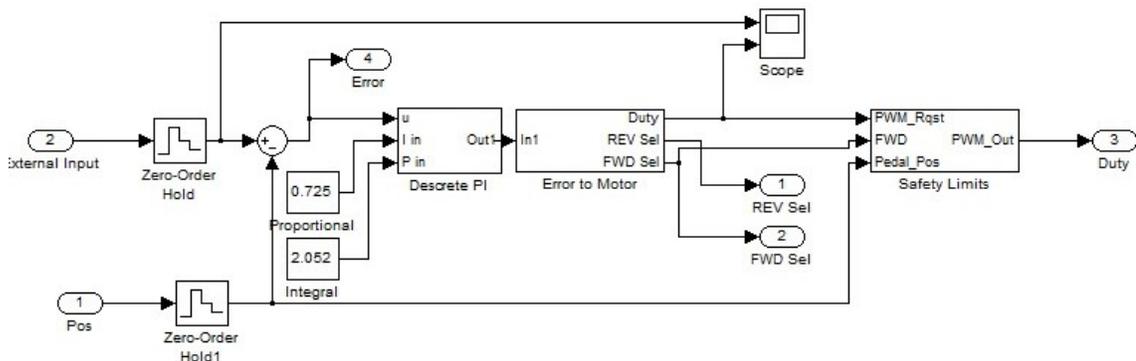
Description	This module contains the functions for the feedback controller for actuator position relative to the vehicles pedal position.
Secret	This module hides PI values, conversion blocks and functionality specific to creating the controlled motor speed value in the form of pulse width modulation duty percentage and direction.
Requirements	None

**Module Design**

Interface	Name	Description
Incoming	External Input	Desired position
	Pos	Position feedback
Outgoing	Error	Difference in actual and desired position
	Duty	Value between 0 and 1 for motor PWM
	Rev Sel	Binary value to enable reverse direction
	Fwd Sel	Binary value to enable forward direction
Uses	Zero-Order Hold	Converts continuous value to discrete
	Scope	Visual display function used for debugging
	Discrete PI	PI controller (4.4.7.2)
	Error to Motor	Converts PI output to motor signals (4.4.7.1)
	Safety Limits	Confirms values are in range (4.4.7.3)

Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	Proportional	Proportional value for PI controller
	Integral	Integral value for PI controller

**Table 4.26 Pedal Position Controller Module Guide/Design**



**Figure 4.38 Pedal Position Controller**

#### 4.4.6.2 Actuator Control Gas\Speed Controller

##### Module Guide

Description	The speed controller module contains the necessary software to drive the desired vehicle speed to the actual vehicle speed by requesting an accelerator or brake pedal position. There is one speed controller for each pedal.
Secret	This module hides the functionality of setting values for a PI controller that drives the speed error to zero. Safety limits are checked for the output values.
Requirements	None

##### Module Design

Interface	Name	Description
Incoming	External Input	Requested Speed
	Speed	Vehicle speed feedback
Outgoing	Error	Difference between desired and actual speed
	Pos Request	Value for pedal position between 0 and 1
Uses	Rate Transition	Changes the sampling rate of a discrete source
	Zero-Order Hold	Converts continuous value to discrete
	Scope	Visual display function used for debugging
	Discrete PI	PI controller (4.4.7.2)
	Safety Limits	Confirms values are in range (4.4.7.3)
Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	Proportional	Proportional value for PI controller
	Integral	Integral value for PI controller

**Table 4.27 Speed Controller Module Guide/Design**

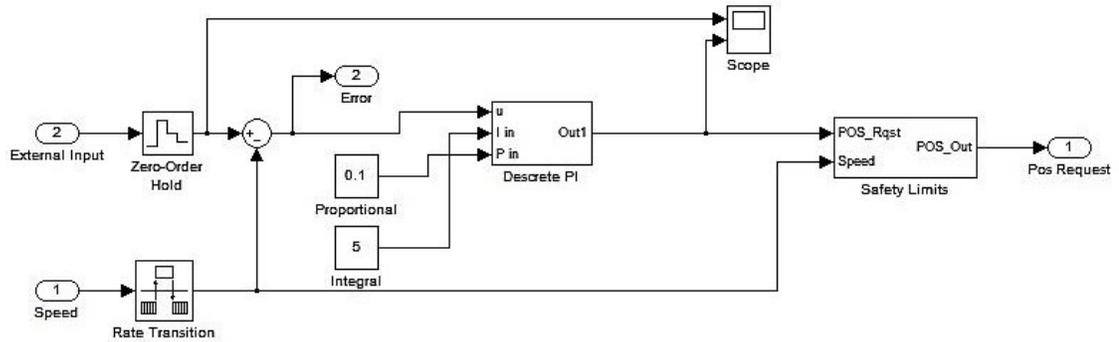


Figure 4.39 Speed Controller

#### 4.4.6.3 Actuator Control Gas\Conflict Prevention

##### Module Guide

**Description** This module is used to prevent the accelerator and brake actuators from moving at the same time and fighting to control the speed of the vehicle.

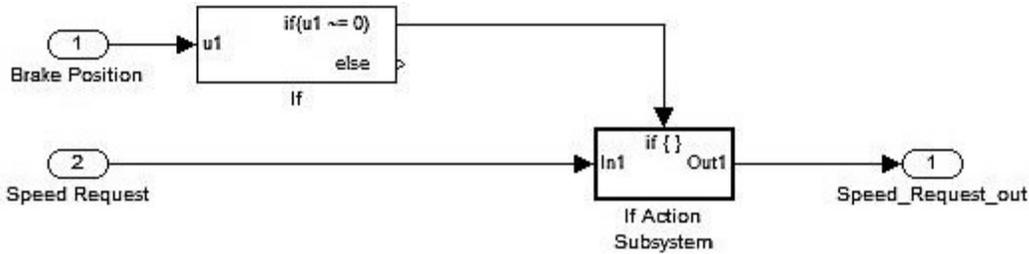
**Secret** In order for the gas pedal to be compressed the brake must be released. In order for the Brake pedal to be compressed, the gas pedal must be released.

**Requirements** None

##### Module Design

Interface	Name	Description
Incoming	Brake Position	
	Speed Request	
Outgoing	Speed_Request_Out	
Uses	If	Condition evaluation to trigger If Action
	If Action	Lets Speed Request through if true, else 0
Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	N/A	

**Table 4.28 Conflict Prevention Module Guide/Design**



**Figure 4.40 Conflict Prevention**

#### 4.4.6.4 Encoder Feedback\Position Decoder

##### Module Guide

**Description** Module converts 2 pulse inputs from encoder into a percentage of travel.

**Secret** Converts rotational pulse readings into translational distance measurement based on 1 inch per rev of lead screw. Then converts into a percentage based on total travel distance being 2.5 inches.

**Requirements** none

##### Module Design

Interface	Name	Description
Incoming	A	Encoder channel A
	B	Encoder channel B
	Reset	User input; count reset
Outgoing	Pos	
Uses	Edge Detector	

Objects	Name	Description
---------	------	-------------

---

Constants	Gain	Scalar to convert from pulse/rev to % travel
Monitored	N/A	
Controlled	Up	Forward pulse counter
	Down	Reverse pulse counter
	Up_Reset	Forward counter reset
	Down_Reset	Forward counter reset

**Table 4.29 Position Decoder Module Guide/Design**

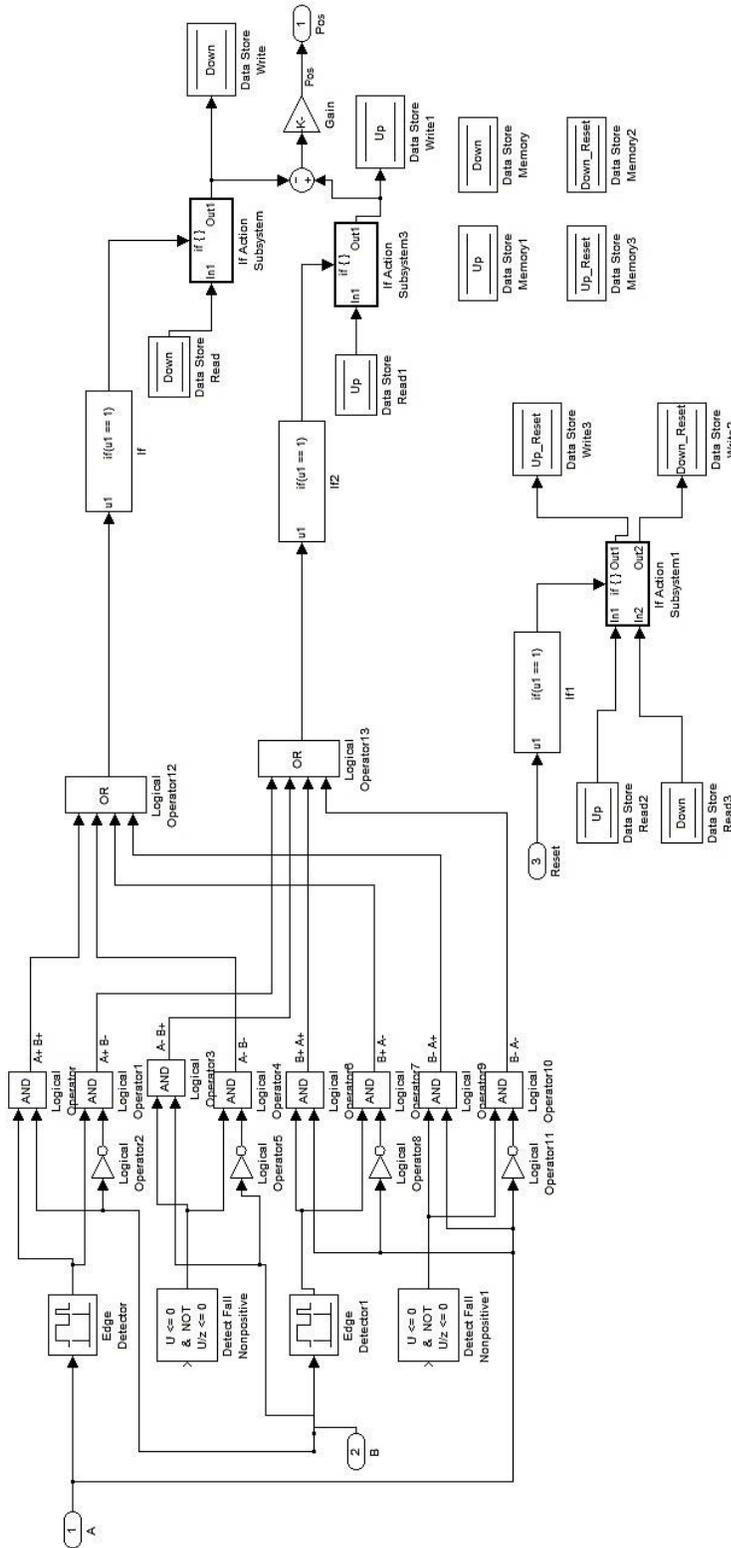


Figure 4.41 Encoder Position Decoder

#### 4.4.7 Implementation Level 4

This section gives a description of modules used by Implementation Level 3.

##### 4.4.7.1 Pedal Position Controller\Error to Motor

###### Module Guide

**Description** This module holds the mathematical equation to convert a position error driven PI controller to usable DC motor inputs for the motor controller.

**Secret** Hides the motor specific parameters for controlling the actuator position using a DC motor that drives a lead screw. Conversion works backwards from pedal contact to input voltage at the motor.

**Requirements** None

###### Module Design

Interface	Name	Description
Incoming	In1	Error value from PI controller
Outgoing	Duty	Duty % for PWM; real value between 0 and 1
	REV Sel	Binary value to enable reverse direction
	FWD Sel	Binary value to enable forward direction
Uses	POS to rad	Actuator position to radians based on lead screw
	Discrete Derivative	Converts input to discrete radians per second
	Rad/s to Volts1	Scalar for geared motor speed to volts
	Volts to Duty	Scalar converts voltage to Duty % for 12V motor
	Abs	Output of absolute value
	Direction Select	Detects value sign to determine motor direction
Objects	Name	Description
Constants	Constant2	Used to invert value through subtraction
Monitored	N/A	

**Table 4.30 Error to Motor Module Guide/Design**

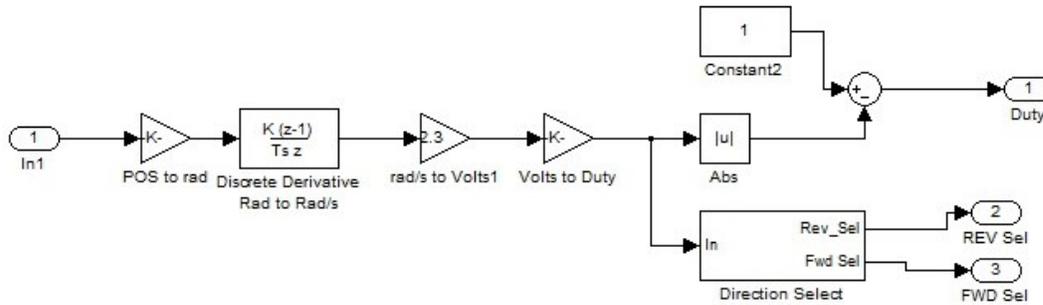


Figure 4.42 Error to Motor

#### 4.4.7.2 Pedal Position Controller \Discrete PI

##### Module Guide

**Description** This module holds the math to execute a discrete PI controller.

**Secret** The PI controller converts an error value into a driving value to create a controllable convergence on the desired state. Used to decrease settling time and make systems stable.

**Requirements** None

##### Module Design

Interface	Name	Description
Incoming	u	Actual and desired error input
	I in	User defined Integral value
	P in	User defined Proportional value
Outgoing	Out1	PI result output
Uses	Product	Multiplies 2 inputs together
	Integrator	Discrete integrator
	Sum	Adds to inputs together
Objects	Name	Description
Constants	N/A	
Monitored	N/A	
Controlled	N/A	

Table 4.31 Discrete PI Module Guide/Deisgn

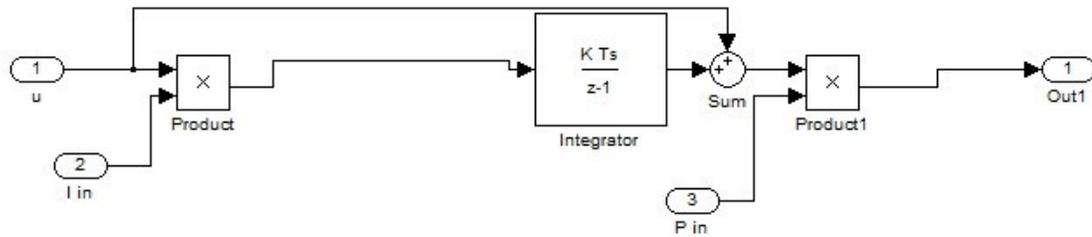


Figure 4.43 Discrete PI

#### 4.4.7.3 Pedal Position Controller \Safety Limits

##### Module Guide

**Description** The safety limits module is used to check to make sure input values are in a safe range. If they are out of range, the values are set to the max or min limit depending on which value is being exceeded.

**Secret** This module contains hard coded limit values that should not be changed to avoid unwanted or unsafe system behavior. This will change out of range values to an upper or lower limit or stop the actuator completely.

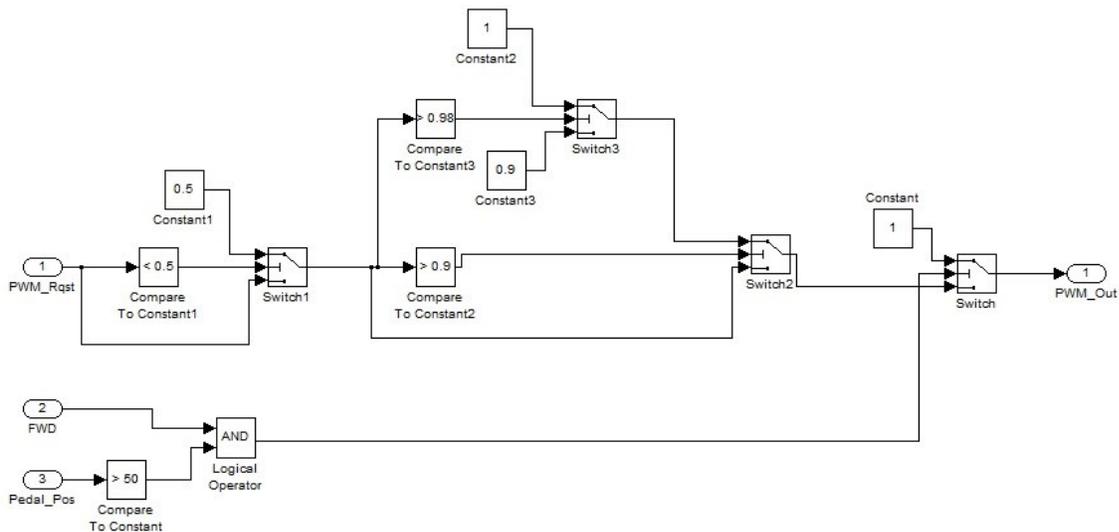
**Requirements** None

##### Module Design

Interface	Name	Description
Incoming	PWM_Rqst	Value that motor controller wants to pass
	FWD	Direction of the motor
	Pedal_Pos	Position feedback
Outgoing	PWM_Out	Verified PWM output for motor
Uses	Compare to Constant	True or False output for logic value compare
	Switch	Switches 2 inputs from logic trigger input
Objects	Name	Description
Constants	Constant	Stops the motor when large external Pos rqst
	Constant1	0.5 is the set max limit for motor speed
	Constant2	Stops motor if very small PWM request

	Constant3	Min speed of motor to prevent damage
	Compare to Constant	POS>50= out of range. Connection problem
	Compare to Constant1	Compare PWM rqst to 0.5. limits max speed
	Compare to Constant2	Compare PWM rqst to 0.9. limits min speed
	Compare to Constant3	Compare PWM to 0.98. Prevents shuttering
Monitored	N/A	
Controlled	N/A	

**Table 4.32 Safety Limits Module Guide/Design**



**Figure 4.44 Safety Limits**

### 4.4.8 Interface Implementation

The software interface has been created using dSPACE ControlDesk. This software provides a drag and drop interface for adding visual displays and user controls. Variables from the Simulink file are linked to features by dragging and dropping variables from a directory tree that lists each block and signal used in Simulink. An interface can be quickly customized for different applications without modifying core software.

## 4.4.8.1 Testing Layout

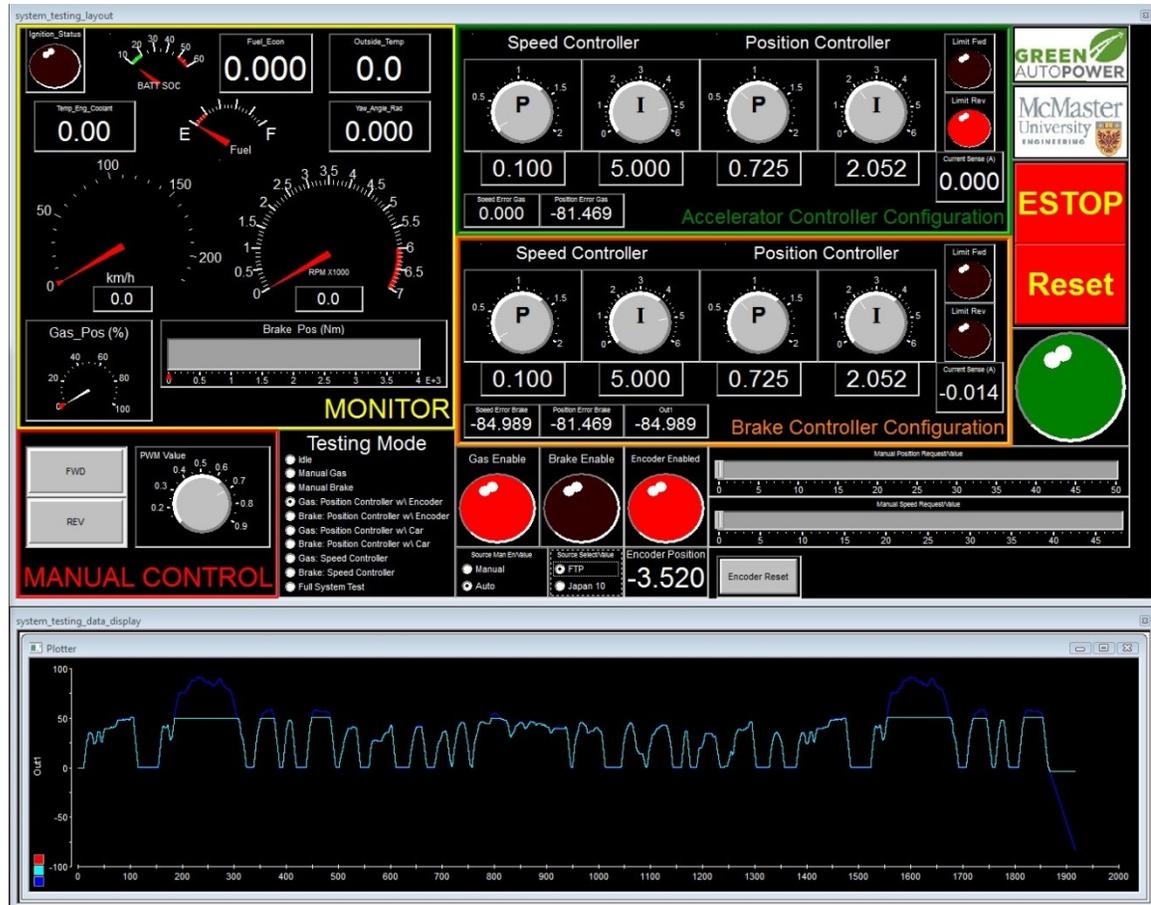


Figure 4.45 User Interface

Figure 4.45 shows the testing interface used to configure and control the actuators. The “Monitor” area, outlined in yellow, shows real-time information received from the test vehicle. Accelerator and Brake Controller Configuration sections are used to adjust the PI values used in the speed and position controllers within Simulink. 4.4.6.1, 4.4.6.2

Manual Control is a user input to control the direction and speed (duty percentage) of the motor controllers. The desired actuator is selected using the radio buttons in “Testing Mode”. 4.4.5.10

The sliders are for testing the PI controllers in the manual source mode. Rather than providing a driving schedule, users can assign the actuator a position or select a speed for the car to reach. 4.4.5.5

The auto mode will feed a time-series data source to the controllers. There are two driving schedules pre-loaded into the software, and will provide an input when auto mode is enabled (4.4.5.4). Playback begins when the software starts running, and will output a zero value when the file is complete.

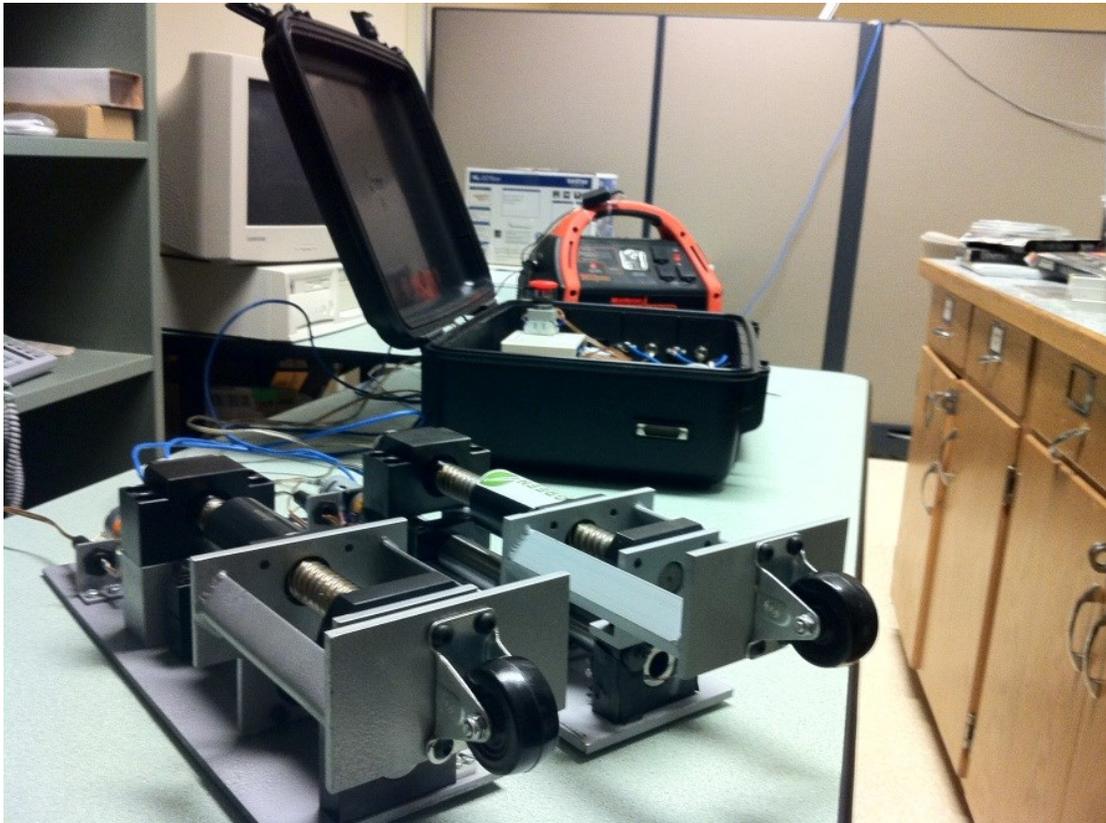
The plotter at the bottom shows a comparison of the driving schedule to speed of the vehicle. When there is a software safety limit imposed, a clipping effect occurs when the maximum point is reached, as seen in Figure 4.45.

The ESTOP button controls the software estop function, disabling both motor outputs. This does not function the same way the hardware e-stop would, if there is a power failure. It will just prevent signals from being passed to the actuators from the software. 4.4.5.12.

LEDs are used as indicators for binary information the user would like to see. Examples are limit switches, ignition status and when an encoder is enabled for feedback rather than the vehicle.

## 5 System Validation

The system validation section is intended to review the final design for functionality and performance using in-lab and field testing. System components are broken up into Mechanical, Electrical, and Software sections. For a complete validation, the system must meet the requirements laid out in chapter 3 and also perform the desired tasks presented in the thesis proposal. Requirements not met are deemed future work and are outlined in section 6.2 for another researcher to continue. Figure 5.1 shows the two actuators and control hardware as it was when this thesis was completed.



**Figure 5.1 Robotic Driving System**

## 5.1 In-Lab Tests

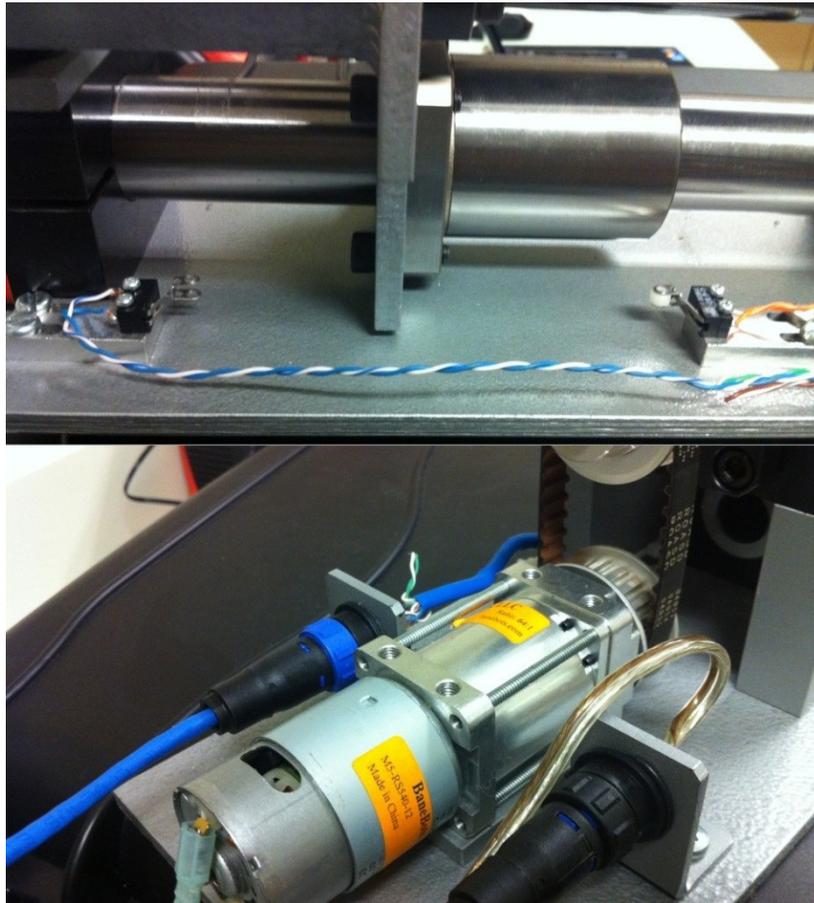
The in-lab testing verifies system components in a controlled environment where modifications are easily performed. Tests are completed systematically to easily identify and fix faults and ensure a smooth integration process.

### 5.1.1 Mechanical

The first actuator was designed and assembled, before parts for the second were ordered, so necessary changes could carry over into the second iteration. The parts came together very well and there were no major mechanical issues for functionality. The DC motors functioned well with the ball screw drive. Tests were done using manual control from a DC power supply to drive the actuator back and forth.

Additional mechanical parts were needed, when the electrical design began. Limit switches needed to be mounted securely and required small aluminum blocks, that made for a more robust and rigid design. In addition to that, the need to add a disconnect for the motor power and sensors was added. This gave a benefit of portability and simplicity to the design. L-brackets are used to hold a panel mount weather resistant connector. These modifications can be seen in Figure 5.2.

There are 7 Mechanical requirements from Chapter 3 that need to be met to verify this design. Some of these were talked about in Chapter 4.2.7, in the preliminary design verification section. Table 5.3 outlines the completion of these requirements.



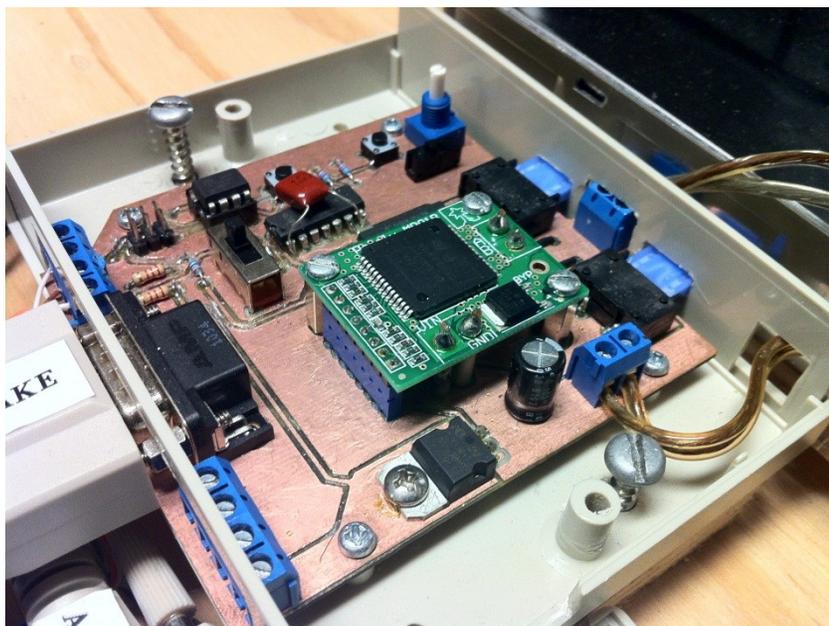
**Figure 5.2 Mechanical Additions**

### 5.1.2 Electrical

Circuit testing was done to verify the functionality of each design. The motor controller circuit was made as simple as possible, so that there are fewer things to go wrong. The electrical design does not include an integrated closed-loop solution, but rather just a means to transform control signals into a driving voltage. Manual controls are included for initial testing, but are disengaged during normal operation. The voltage-detect and safety circuits add safety in case of a power failure or the need for an emergency stop.

### 5.1.2.1 Motor Controllers

The motor controller circuit had some issues initially with a floating voltage for one of the logic controller inputs. When the limit switches are triggered, this creates an open circuit going to the logic controller that can become noisy. The actuator would continue moving because of a floating logic level, despite having a pull down resistor. To fix this, a .1uF capacitor was added between the logic input pin and ground. This acts as a filter to reduce noise, which creates stable operation on the sensitive logic chips.



**Figure 5.3 Motor Controller Circuit V1**

An encoder was used for position feedback during lab testing, as seen in Figure 5.6. The motor controller circuit was designed to have an encoder attached by including pins for A and B channels, Ground and 5V lines broken out on screw terminals. When the encoder was tested, it was noticed that pulses were not always being detected when the

actuator travelled at high speeds. It was determined that running the pulses through the same wire as PWM control created noise on the encoder signal lines. When the encoder is connected directly into the dSPACE box using a CAT 5e cable, with twisted pairs, the encoder operates fine at all speeds.

In-lab testing proved reliable functionality from the motor controller circuit. Revisions were made to the second board to incorporate the .1uF capacitor. The manual control as well as control from the dSPACE box was tested extensively to ensure the limit switches were functioning correctly.

#### 5.1.2.2 Manual Control Testing

The manual control of each motor controller circuit was verified during in-lab testing. Manual mode is selected by a DPDT slide switch on the physical circuit board. The potentiometer works to control speed and two push buttons to enable each direction. Direction selection goes through logic controllers, which check the status of each limit switch before sending a command to the motor controller. If both switches are pressed, disconnected, or fault in a wire than no movement can occur. This function works reliably and meets the requirements specified in Section 3.3.2.

#### 5.1.2.3 Safety Circuit and Voltage-Detect

The voltage-detect circuit produces enough current to hold the relays on the safety circuit, providing a path for the actuators motors to be powered. This has been tested by doing bench tests with the battery backup system pictured in Figure 5.4. When power is cut from the circuit, it takes approximately 3 seconds for the residual voltage being

---

produced to drop low enough to turn off the relays. When the E-stop switch is pressed, this immediately disconnects the voltage detect circuit and turns off the relays.

The next step of the safety circuit is to drive the motor backwards until the reverse limit switch is hit. There is an issue with the motor controller board, in that it fails to produce a voltage out to the motor. While the circuit is powered, the motor controller should be continually trying to drive the motor in reverse. When the relays lose power, the safety circuit motor controller connects to the actuator. While disconnected, but powered, the motor controller module appeared to get very warm. This may indicate misuse of the motor controller or an undetected short in the circuit. Over extended periods of producing an open voltage (or short), the motor controller has become damaged and produces no voltage out. To fix this, a new motor controller board needs to be installed and additional control added to only trigger its operation when it is needed. Due to time restrictions, testing continued without using the safety circuit, but using the battery back-up with the E-stop button to cut power.



**Figure 5.4 Safety Circuit Bench Testing**

#### 5.1.2.4 Connecting to dSPACE

The circuits were connected to the dSPACE for testing with the feedback signals from an encoder and the limit switches, while providing pulse width outputs to each controller. The dSPACE is powered from a 12V wall jack for lab tests. The dSPACE box was able to run the motors correctly and receive limit switch information, that's used to trigger software emergency stops and provide on-screen user feedback. The encoder did not work correctly when run alongside the oscillating PWM motor control wires. As discussed in section 5.1.2.1, this was overcome using a separate connection to the dSPACE.

### 5.1.2.5 Full System Integration and Testing

After all the issues with the circuits were fixed, all the electronics were securely installed into a pelican case. Inner connections are made using appropriately sized wires and exterior connections with off the shelf connectors to easily assemble and disassemble. This can be seen in Figure 5.5.



**Figure 5.5 Equipment Case Assembly**

An electrical system test was performed using all the newly assembled wiring and dual actuators. The dSPACE box was able to control both motor controller circuits simultaneously, and provide encoder feedback.

### 5.1.3 Software

Software in-lab testing allowed some simulations to be performed on the controllers. Limited PI controller tests were performed to ensure they are implemented correctly. The user interface was tested to verify all the features are connected and functioning.

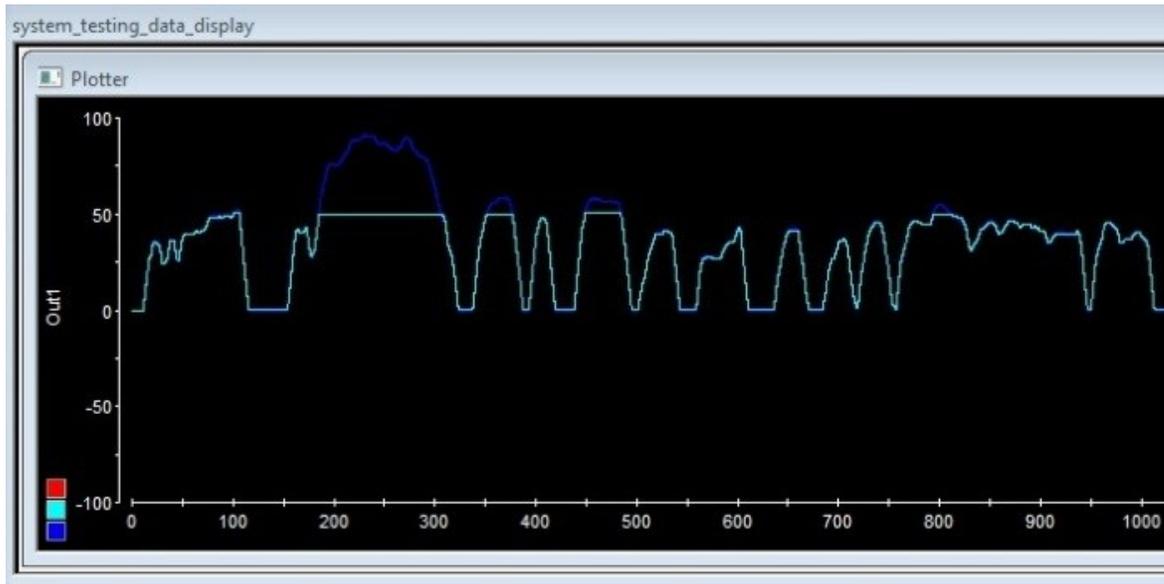
#### 5.1.3.1 Controller Testing

The position controller underwent in-lab testing using a digital encoder for feedback, as seen in Figure 5.6. The control software converts the incoming A and B channels to a linear position, then an overall percentage, using 2.5” travel as the max. (ie. 2.5” extended will output a value of 100). The actuator is then driven to different positions using the user interface. Initial testing revealed that a jittering occurred around the desired position. This happens because the PI controller is trying to reduce the error to zero, and does not have a tolerance built in. There are safe operating limits for the motor, which are hard coded into software so as not to try to drive the motor too slowly. The PI controller outputs a PWM duty value until the error is reduced to zero. If the software continues to give the minimum duty, there will be overshoot since the inertia of all the spinning parts cannot stop in time at higher speeds. As the controller drives the motor backwards and forwards, trying to reach the desired position, it continually overshoots its mark. To fix this issue, a tolerance was put in so that the controller stopped when it reached its max resolution.



**Figure 5.6 Motor with Encoder**

One key bench test that was performed was to see if the position controller could follow a programmed schedule. This was done using a standard driving schedule that is intended as a speed vs time output and fed the values into the position controller. There are hard-coded safety values that the controller cannot exceed, resulting in a peak value of 50%. When the driving schedule requests more, the actuator remains at this peak. Figure 5.7 shows the results after 1000 seconds of testing.



**Figure 5.7 FTP Schedule with Position Controller**

### 5.1.3.2 User Interface

Many iterations of the user interface were made as testing progressed. The requirements section leaves this open to whatever a user would require for the tests being performed. The user interface was designed to perform iterative testing of the software to ensure all the subsystems are functioning properly. Figure 5.8 shows the overall view of the final user interface used in testing. The testing modes section, shown in Figure 5.9, has a variety of test scenarios to verify the position and speed controllers of each actuator and allows the user to use either the car or encoder as feedback. The in-lab tests were unable to incorporate the speed test, as there are no feedback signals available. The tuning of this controller is directly related to the characteristics of the vehicle, and requires the car to be on the Dyno for tuning. Due to the availability of equipment and time restrictions, the speed controllers and the full system tests weren't able to be completed.

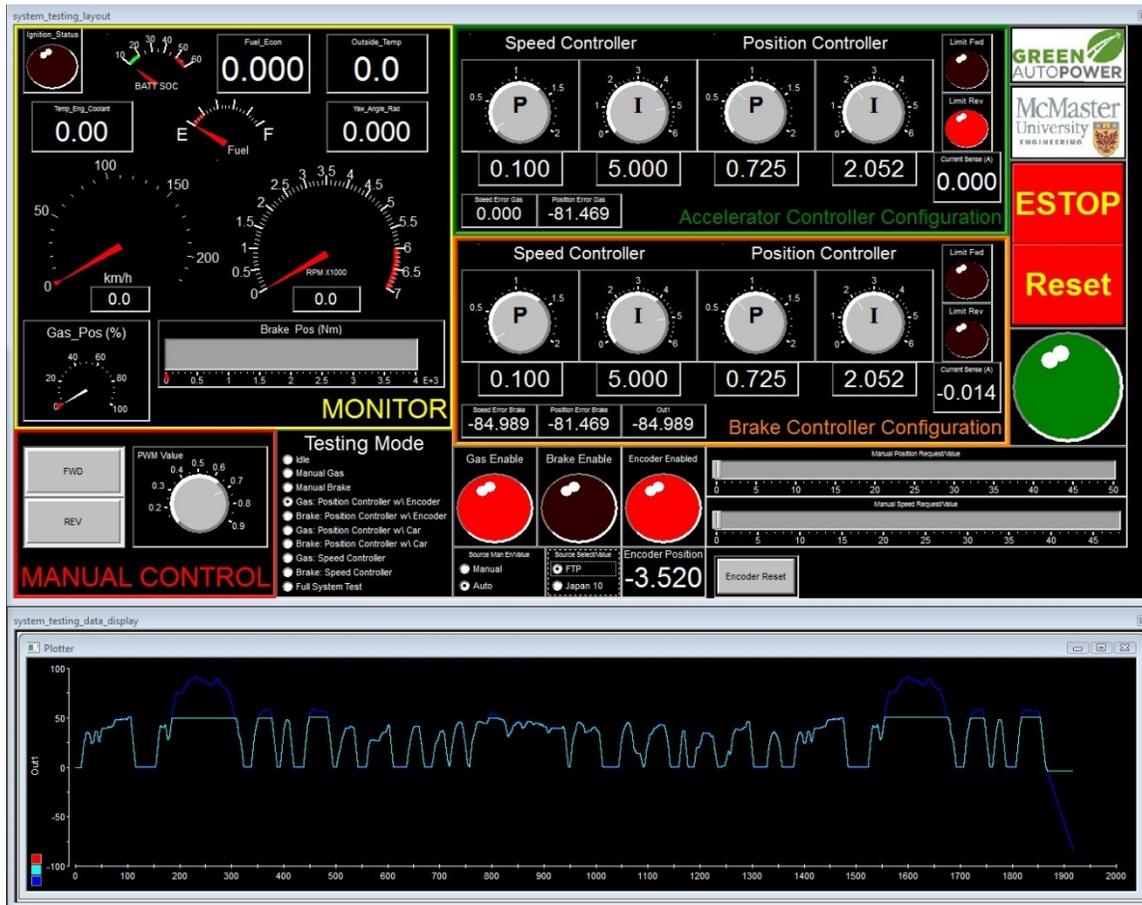


Figure 5.8 User Interface



Figure 5.9 Testing Modes

## 5.2 Field Testing

The goal of the overall design was to drive the gas and brake of a Ford Escape Hybrid vehicle for in-lab testing. The mechanical design, carried out in chapter 4, describes design decisions that lead to the final implementation. Using a ball screw driven by a geared DC motor, and a rolling contact, the final design is capable of operating the arc motion of the vehicles pedals.

### 5.2.1 Mechanical

With in-vehicle testing, it was found that each actuator could successfully drive each pedal, at the maximum speed required, while having enough static friction to prevent the actuator from sliding. Each pedal can be compressed the required distance determined in sections 3.2.4 and 3.2.5. Limit switches prevent damage to each unit, and do not restrict recovery when a direction limit has been reached. Having the limit switches as a feedback signal creates a flag in software, alerting the operator that something is wrong.



**Figure 5.10 Final Implementation Testing**

## 5.2.2 Electrical

The field testing consisted of connecting the dSPACE to the vehicle. Tests were conducted to ensure the right signals were being received. There were no issues with the wiring, and the dSPACE box was able to receive and display values. Initial tests were done by driving around and displaying values using the ControlDesk software. A virtual dashboard was created to display messages to verify functionality.

## 5.2.3 Software

The next tests were done using vehicle feedback. The setup can be seen in Figure 5.11. There is an error output on the user interface measuring the difference between feedback value and driven values. The following table illustrates the results while driving the gas pedal.

	Driven Value	Actual Value	Error
Position 1	10	8.333	1.667
Position 2	20	17.361	2.639
Position 3	10	13.194	-3.194
Position 4	0	2.778	-2.778

**Table 5.1 Test 1 - Gas Position Controller****Figure 5.11 Field Testing**

From Table 5.1, it's observed that there is undershoot in the controller. This is because of the tolerance built in to the controller to prevent jittering and also the tuning of the controller. When the controller starts requesting a speed slower than the motor's capabilities, it stops. For Test 1,  $P = 0.566$ , and  $I = 2.750$ . For the next test, the  $P$  and  $I$  values are adjusted using the Ziegler-Nicols Method (Nise, 2009). The values  $P=0.725$  and  $I= 2.152$  were found to give the least amount of error during these tests.

	Driven Value	Actual Value	Error
Position 1	10	7.639	2.361
Position 2	20	19.44	0.556
Position 3	10	12.5	-2.5
Position 4	0	2.083	-2.083

**Table 5.2 Test 2 - Gas Position Controller**

The PI controller values are editable in real-time from the user interface which makes tuning easy to do. With further testing and development of the controller software, it is possible to reduce the errors more.

#### 5.2.4 Driving Cycle Recording

Using the dSPACE ControlDesk interface, it is possible to record any incoming values, including CAN messages, to a .mat file, saved on the interfacing computer hard drive. Values are stored in a matrix where a sensor is indexed by a row value, and the time of the recording is saved as its column value. Using matlab script, plots can be generated of the discrete recorded values for analysis. An example of this can be seen in Figure 3.1.

To replay a recorded driving cycle, the only way I've found is to load the .mat file on to the dSPACE box, using Simulink. Once the file is stored on the dSPACE memory, you can access its values in real-time using Simulink blocksets. The driving cycle can be replayed to command the actuators. This feature has not been fully implemented in the user software, because it would be better to develop a script for the user to select a file to replay from the interface, rather than have hard coded driving cycles. There are 2 standard

driving cycles available in the user interface. One was downloaded from the internet and loaded from a .mat file and the other generated using Simulink signals. These are used for automated control, such as the test performed in Section 5.1.3.1.

### 5.2.5 Dynamometer Setup

During this project, there have been facility improvements taking place, causing limited access to the Dyno equipment. This has delayed development in control and testing this equipment. There has also been limited support on interfacing with the Dyno and what would be necessary to control the resistance. Interfacing with the dyno was removed from the scope of this thesis, but is still a necessary requirement for the completion of an in-lab testing suite.

### 5.2.6 Ease of use

Both actuators are designed in a way that they are duplicates of each other. By doing this, it creates an easy installation that only relies on connecting the right cables from the control board. The electronics are placed in an enclosure to prevent any damage in transportation, and also simplify the external connections, as seen in Figure 5.5. This can now be used by researchers in the lab, with little explanation of the setup required.

## 5.3 Validation Matrix

The validation matrix serves as a summary of what requirements from Chapter 3 have been met.

<b>Description</b>	<b>Type</b>	<b>Requirement</b>	<b>Validation</b>
Force Output for Gas	Mechanical	3.2.1	5.2.1
Force Output for Brake	Mechanical	3.2.2	5.2.1
Size Constraints	Mechanical	3.2.3	4.2.7.4
Actuation distance gas pedal	Mechanical	3.2.4	4.2.7.3
Actuation distance brake pedal	Mechanical	3.2.5	4.2.7.3
Minimum speed	Mechanical	3.2.6	4.2.7.2
E-Stop Implementation	Electrical	3.3.1	5.1.2.3
Manual Control	Electrical	3.3.2	5.1.2.2
Monitored Values	Control/Software	3.4.1	4.4.4
Controlled Values	Control/Software	3.4.2	4.4.4
Modular Design	Safety	3.5.1	4.3.2
Safety Mode	Safety	3.5.2	Incomplete; 5.1.2.3
Battery Back-up	Safety	3.5.3	5.1.2.3

**Table 5.3 Requirements Validation Matrix**

Table 5.4 lists overall system requirements that are described in Section 1.2 of the Introduction chapter.

<b>Description</b>	<b>Type</b>	<b>Validation</b>
Dynamometer Setup	Electrical/Software	Incomplete; 5.2.5
Gas and Brake Control	Mechanical/Electrical	5.2.1
Closed-Loop Control System	Control/Software	Incomplete; 5.1.3
Vehicle Interfacing	Electrical	5.2.2
Driving Cycle Collection	Software	5.2.4

**Table 5.4 System Overview Validation Matrix**

## 6 Conclusion

The conclusion chapter reviews the accomplishments of this thesis and discusses future work to complete unfinished tasks. The reader will have an understanding of how this project benefits researchers at McMaster and give an explanation of where to go next.

### 6.1 Accomplishments

The initial goal of this thesis is to present an alternative driving simulation method to off the shelf solutions. The projects intention is to end up with a prototype of a working system to preform driving cycle playback using a dynamometer.

The mechanical design of a linear actuator was discussed in Chapter 4. The requirements presented in Chapter 3 outline the functional needs of this system, which include physical dimensions, motion ranges, force, speed and precision. An optimal design that balances functionality with cost was chosen and developed. Testing in Chapter 5 verified the Mechanical design to be adequate for the purposes of this project.

Electrical system design was looked at in Chapter 4, Section 3. It was determined that the actuator would need custom circuitry developed for control and safety. Consideration of power systems, sources, and potential safety risks are looked at to ensure safe operation of the entire system. Scenarios including power failure and unwanted behavior are some of the design criteria considered. Verification of the electrical systems performed in Chapter 5 found some overall hardware flaws, but ultimately proves the design is feasible. Testing involved debugging some issues, but the result was two

---

separate working systems. Further electrical hardware development is included in the Future Work, Section 6.2.

The software development requirements included a modular component to make editing the control software easy for future researchers. The benefit of having a custom developed system is having the ability to change the actuator control-loop algorithms to optimize system response times. Further work is needed on the control loop implementation for the two actuators to work together. In sections 5.1.3, and 5.2.3, verification was performed to test the PI controller. The tests prove positive using positional feedback. Due to time and resource constraints, testing with the speed feedback control could not be completed. The modular requirement was met, and there is a great benefit programming using Simulink for visualizing signal propagation through the software. Having set out the groundwork in Simulink for the control software, it should be less of a challenge for future researchers to pick up where this thesis left off.

The final system cost for Mechanical and Electrical development totaled \$3853.05. Additional equipment needed includes a laptop computer, the dSPACE Micro Auto Box, Matlab license and Simulink. The majority of the cost comes from the dSPACE box, and future research may involve replacing this with a cheaper embedded solution that meets the needs of the project.

## 6.2 Future Work

This section is intended to outline outstanding tasks that need to be completed to verify the final implementation. The scope of the project was very ambitious for a Master's degree, and should be continued by another researcher.

### 6.2.1 Mechanical

There needs to be a retrofit to the original actuator for the additional items added during testing. Limit switch blocks need to be fabricated and mounted. The connector brackets need to be securely mounted.

### 6.2.2 Electrical

Further testing is needed to implement the Safety circuit. The actuator is not being released from the pedal (driven in reverse) when an E-stop is triggered or power is lost. In order to fulfill the requirements in Section 3.3.1, this circuit will have to be implemented for each actuator.

Additional electrical work would involve interfacing the Dynamometer with the dSPACE box in order to control resistances on the rotating mass. It is not clear that it's possible to do with the Dyno we have, and would require a substantial amount of work to interface with the un-documented operation of that equipment.

### 6.2.3 Software

The motor controller algorithm has not been fully tested with 2 actuators. There has also been very little field testing done with the PID control. Further development is needed to have both actuators work together to adjust the speed in an efficient manner.

A controller will be required to adjust the resistance of the dynamometer to desired values to simulate road frictions. The scope of this project calls for in-lab driving simulations for various driving scenarios, to test the power train controller. These scenarios would include driving up hill, in high winds, icy conditions etc. There is a potential for future work in developing a method of capturing values though driving in different scenarios and accurately replaying them in the lab using the dyno.

Further work should also be done to develop an easy way to select user recorded driving cycles to be loaded and replayed for testing with the Dyno. Currently this has to be done in Simulink and code be modified and compiled to call the required driving cycle. Reducing the need for a user to alter code to perform their daily tasks will create a more usable system.

## 7 Appendix

### 7.1 Mechanical Drawings

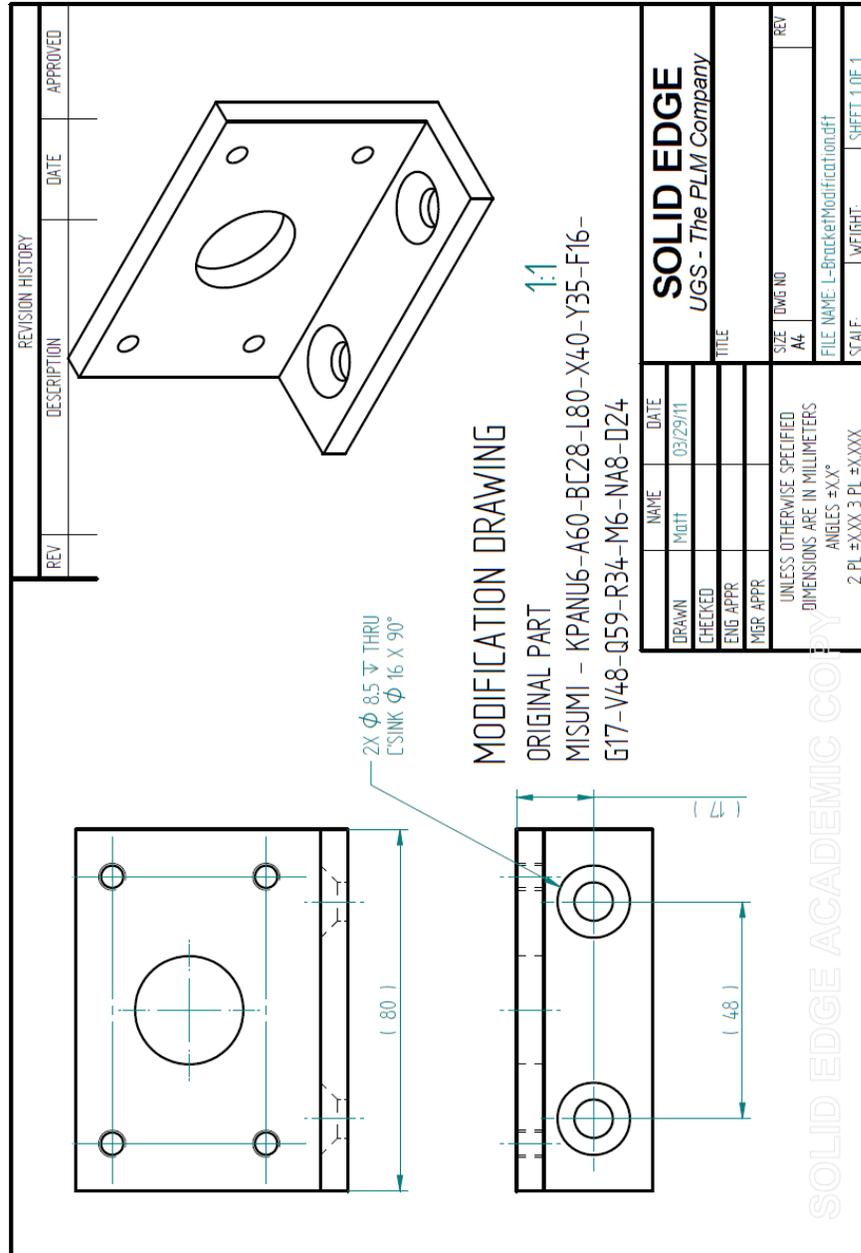


Figure 7.1 L-Bracket Modification Drawing

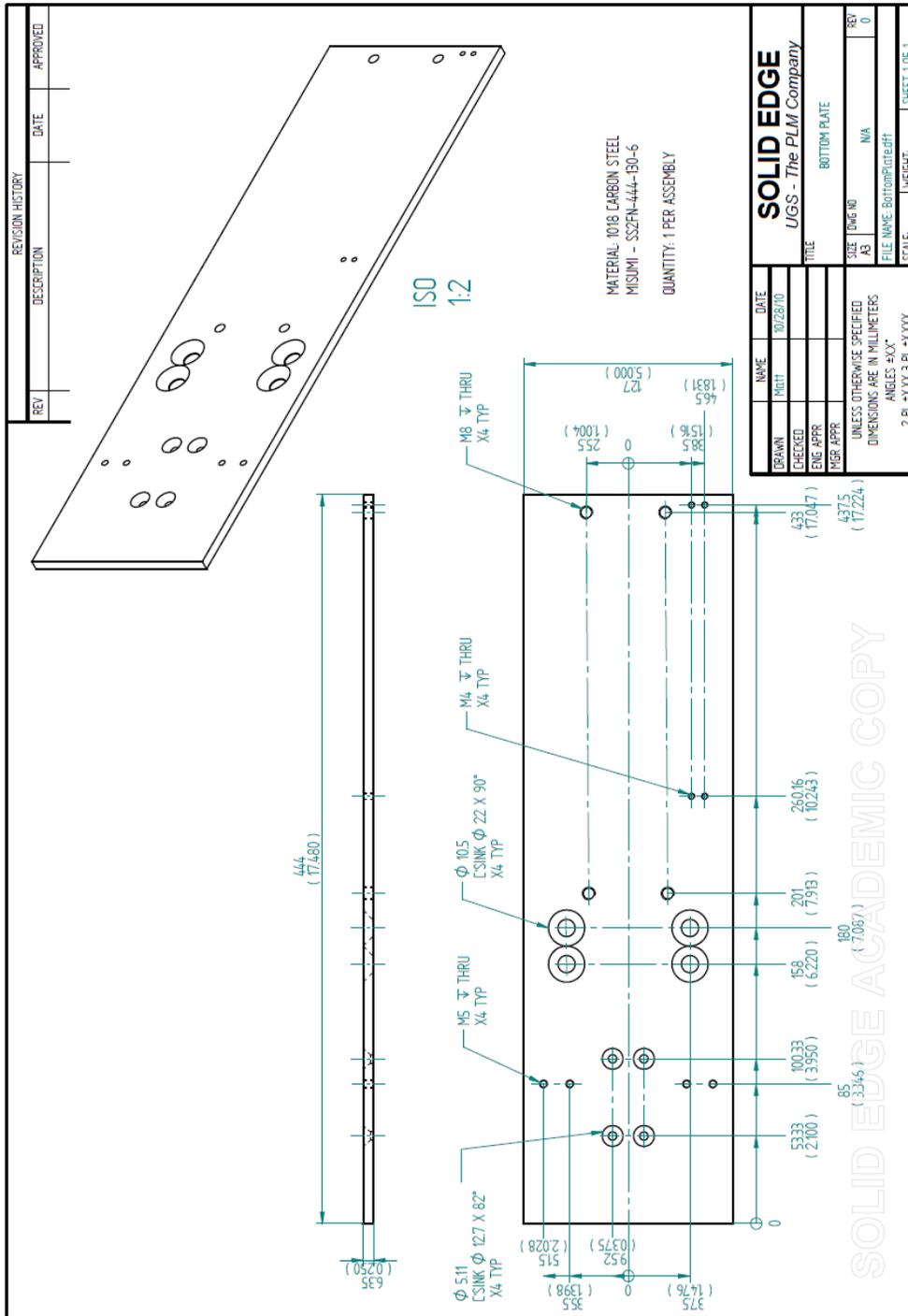


Figure 7.2 Bottom Plate Drawing



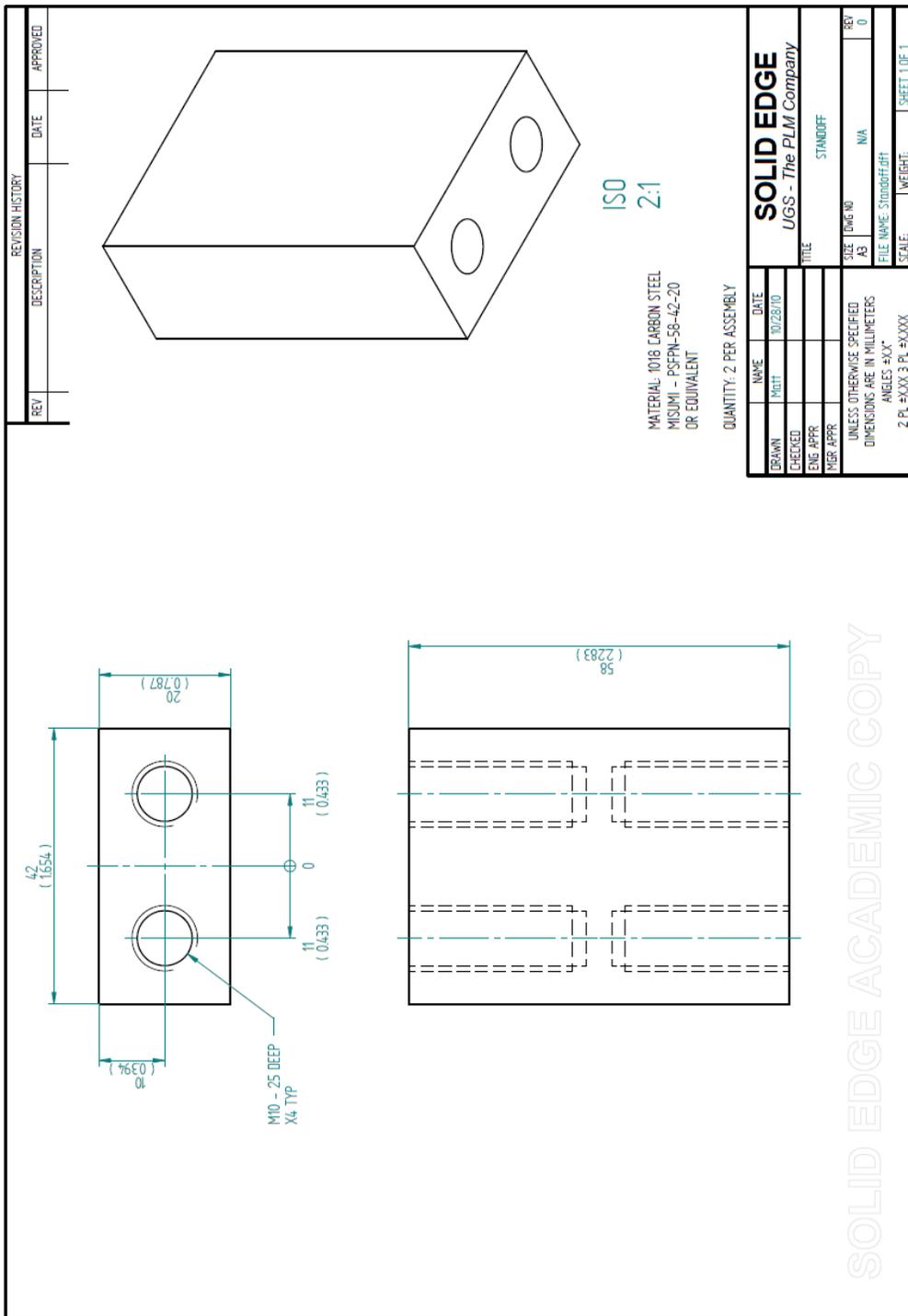


Figure 7.4 Standoff Drawing

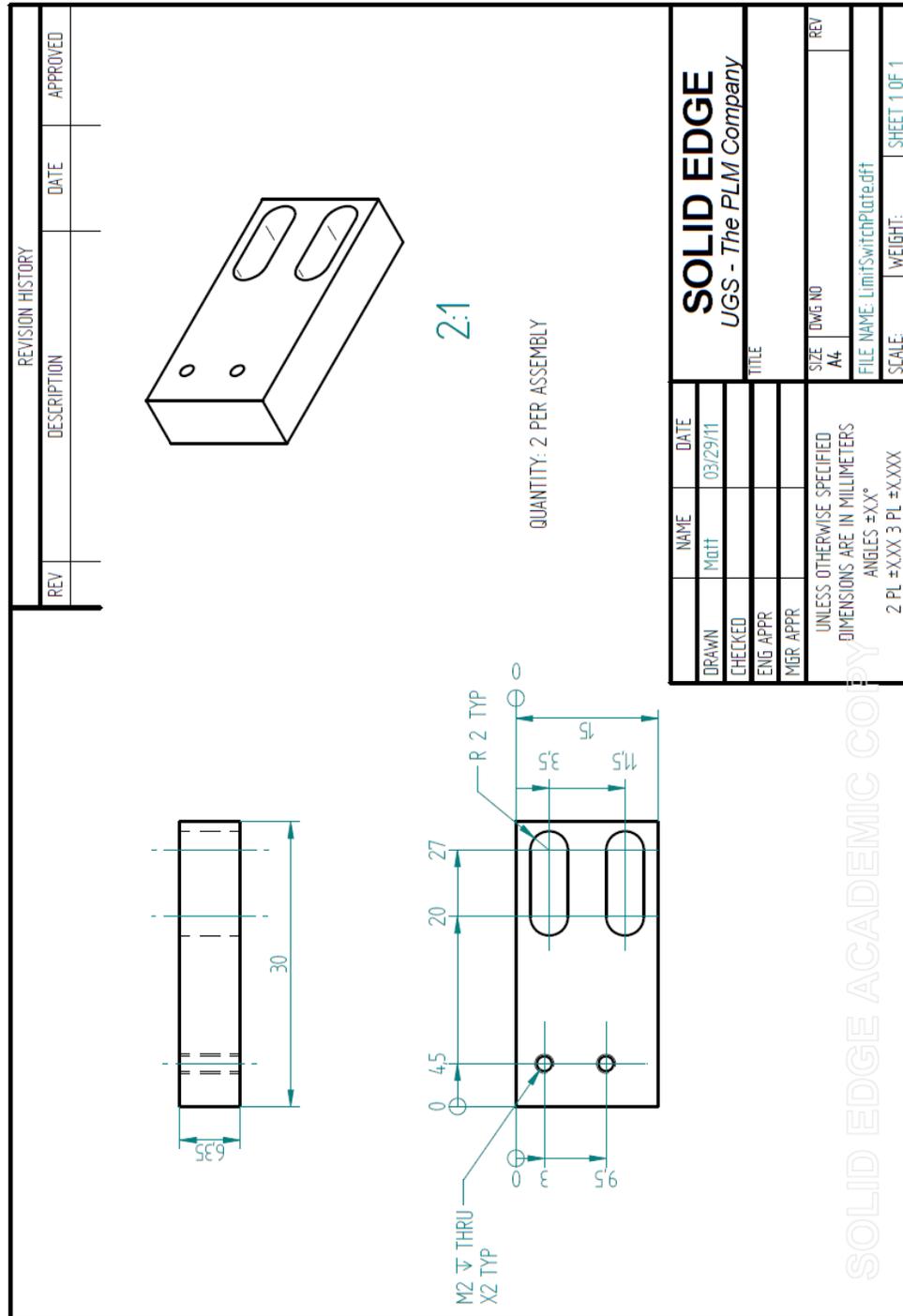


Figure 7.5 Limit Switch Plate Drawing

### 7.2 Printed Circuit Board Design

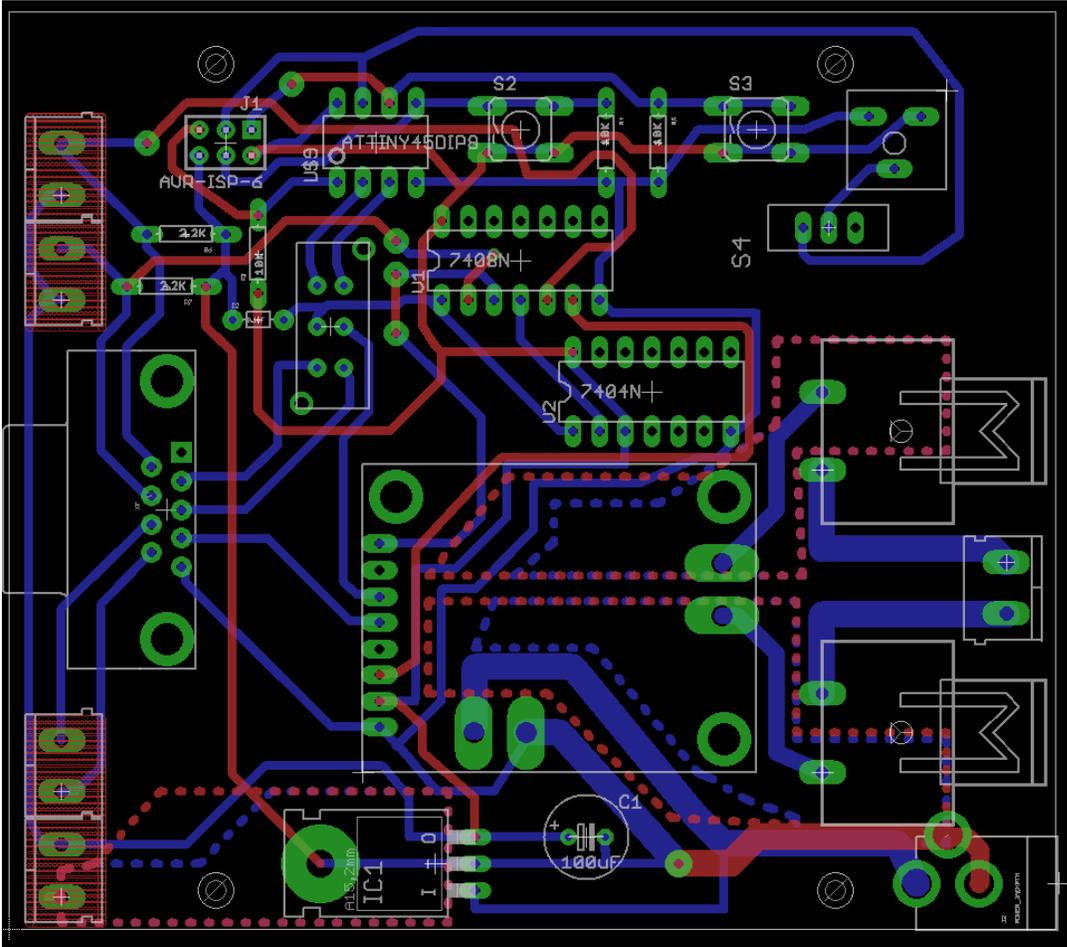


Figure 7.6 Controller Board PCB Layout

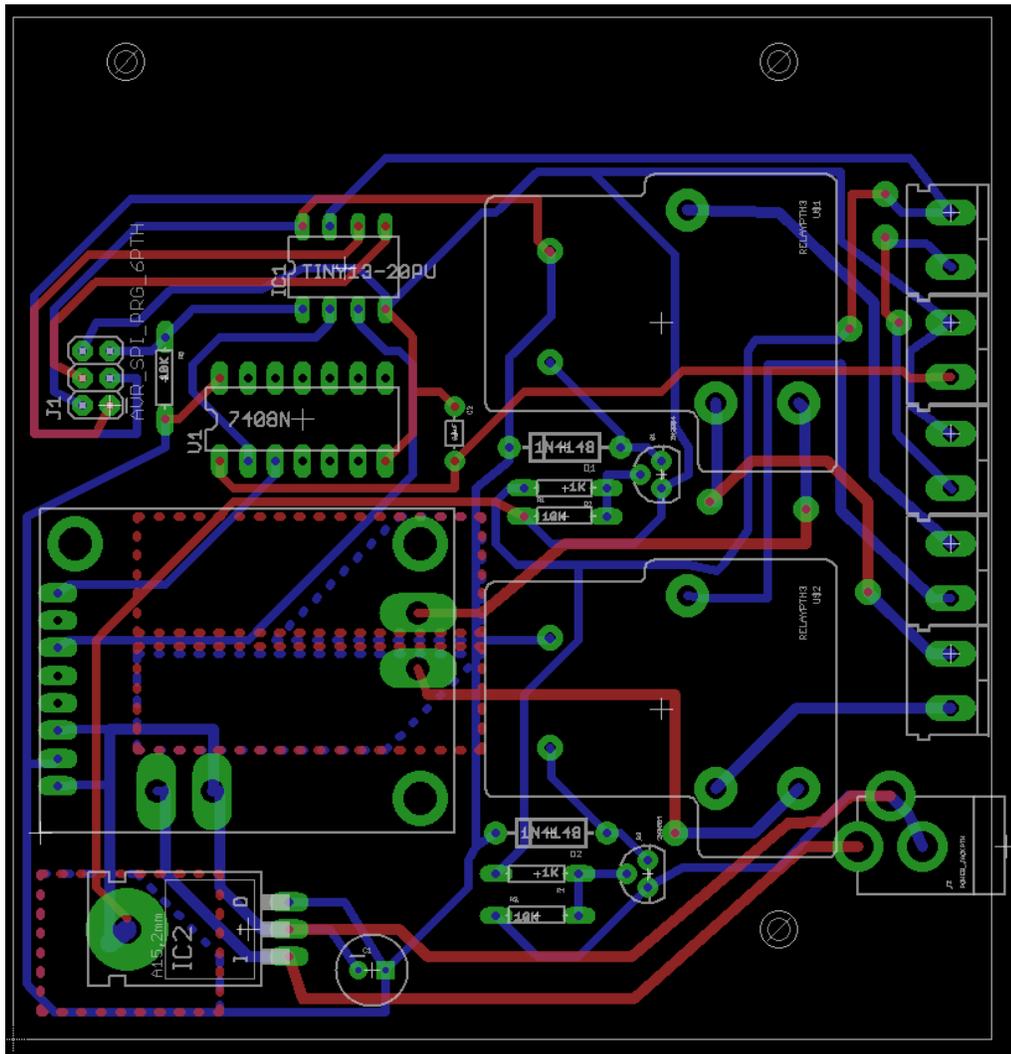


Figure 7.7 Safety Circuit PCB Layout

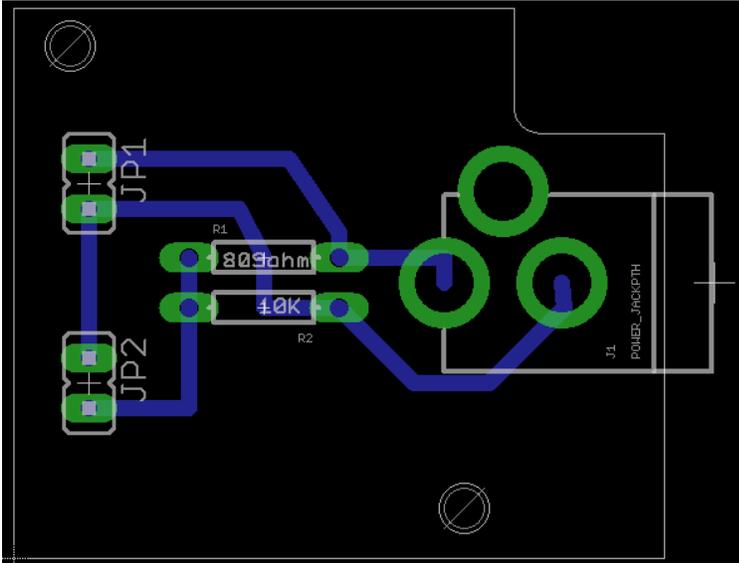


Figure 7.8 Voltage Detect PCB Layout

## 8 Bibliography

**CiA CAN Made Easy** [Online] // CAN in Automation. - 2011. - January 5, 2011. - [http://www.can-cia.com/pg/can/additional/can\\_basics\\_print.pdf](http://www.can-cia.com/pg/can/additional/can_basics_print.pdf).

**CiA CAN Specification 2.0, Part A** [Online] // CAN in Automation. - 2011. - January 5, 2011. - <http://www.can-cia.org/fileadmin/cia/specifications/CAN20A.pdf>.

**dSPACE GmbH Hardware Installation and Configuration** [Book]. - Paderborn, Germany : dSPACE GmbH, 2009.

**Ehsani Mehrdad** Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design [Book]. - [s.l.] : CRC Press LLC, 2005. - 0-8493-3154-4.

**Environmental Systems Products Holding Inc.** Ontario's Drive Clean Operator's Manual SystemOne 4000 [Book]. - Oakville : [s.n.], 2002.

**Ford Motor Company** 2010 Ford Escape Hybrid Owners Guide [Book]. - 2009.

**Horiba** ADS-7000 [Online] // Horiba Automotive Test Systems. - Horiba, 2010. - September 2010. - <http://www.horiba.com/us/en/automotive-test-systems/products/mechatronic-systems/vehicle-test-systems/details/ads-7000-973/>.

**Koot Michiel [et al.]** Energy Management Strategies for Vehicular Electric Power Systems [Journal]. - [s.l.] : IEEE Transactions on Vehicular Technology, 2005. - 3 : Vol. 54.

**Manzie C. [et al.]** A Comparison of Fuel Consumption Between Hybrid and Intelligent Vehicles During Urban Driving [Journal]. - Victoria, Australia : IMechE, 2005. - Vol. 220.

**Miller John M. and Everett Michael** An Assessment of Ultra-capacitors as the Power Cashe in Toyota THS-II, GM-Allision AHS-2 and Ford FHS Hybrid Propulsion Systems [Conference]. - Austin, TX : Applied Power Electronics Conference and Exposition, 2005. APEC 2005. Twentieth Annual IEEE, 2005. - Vol. 1. - 0-7803-8975-1.

**Miller John M.** Hybrid Electric Vehicle Propulsion System Achitectures of the e-CVT Type [Journal]. - [s.l.] : IEEE Transactions on Power Electronics, 2006. - 3 : Vol. 21.

**Mitsubishi** Continuously Variable Transaxle (CVT) [Online] // Mitsubiti Lancer X Club. - 2010. - September 15, 2010. - [http://lancerox.ru/tech/lx\\_cvt.pdf](http://lancerox.ru/tech/lx_cvt.pdf).

**Namik Hazim, Inamura Tokushu and Stol Karl** Development of a Robotic Driver for Vehicle Dynamometer Testing [Report]. - University of Auckland, New Zealand : [s.n.].

**Nise Norman S.** Control Systems Engineering, Fifth Edition [Book]. - Pomona, CA : John Wiley & Sons, Inc, 2009. - 978-81-265-2153-1.

**NZ Government** Where do the fuel consumption figures used on this site come from? [Online]// fuel\$aver. - NZ Transport Agency, 2009. - February 3, 2011. - <http://www.fuelsaver.govt.nz/explain.html>.

**Romanchik Dan** Robotic Drivers Take the Drudgery out of Testing [Online]. - 2004. - September 1, 2013. - [www.edn.com/Pdf/ViewPdf?contentItemId=4382246](http://www.edn.com/Pdf/ViewPdf?contentItemId=4382246).

**Staackmann Milton, Liaw Bor Yann and Yun D.Y.Y** Dynamic Driving Cycle Analysis Using Electric Vehicle Time-Series Data [Conference]. - Honolulu : Energy Conversion Engineering Conference, 1997. IECEC-97., Proceedings of the 32nd Intersociety, 1997. - Vol. 3.

**Stahle** SAP-2000 [Online]. - September 2010. - September 2010. - <http://www.stahle.com/en/produkte.php?gr=1&prod=1>.

**Stalman M., Schouten Niels J. and Kheir Naim A.** Control Strategies for Parallel Hybrid Vehicles [Conference]. - Chicago : Proceedings of the American Control Conference, 2000.

**Vahid and Givargis** Embedded System Design - A Unified Hardware/Sftware Introduction [Book]. - [s.l.] : John Wiley & Sons, Inc., 2002. - ISBN 0-471-38678-2.