Nonlinear Optimal Control of Automated Vehicles in a Connected Environment

by

Derrick Tan

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2019

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis is based around the University of Waterloo EcoCAR Team (UWAFT) and the EcoCAR Mobility Challenge. The overall objective of the competition is to design and build a hybrid electric vehicle with SAE Level 2 Autonomous capability. The vehicle platform used in this thesis was based on the 2019 Chevrolet Blazer – the vehicle that General Motors has donated to UWAFT as part of the EcoCAR Mobility Challenge.

The scope and objective of this thesis is comprised of three parts:

First, various vehicle models were considered and developed using MATLAB and Simulink, as well as ADAMS Car. These models were developed and used for the simulation of the vehicle as well as for the development of vehicle dynamics controllers.

Second, various control architectures and strategies were developed and evaluated to understand the benefits and limitations of each controller design under varying situations. Controllers for generating viable and optimal paths, as well as controllers for controlling the vehicle to track a path were developed.

Third, a visualization framework was developed for streamlining the development of connected and automated vehicle (CAV) systems. Simulation environments for these models were also developed in Simulink (visualized using the Unreal Engine) as well as using ADAMS Car.

Acknowledgement

There are many individuals who I must thank as they have been instrumental to my success as a graduate student and in my development as an engineer. I would like to thank Dr. Roydon Fraser, Dr. Michael Fowler, Dr. William Melek, and Jonathan Spike who have all provided excellent guidance, advice, and support throughout my graduate studies. I would also like to thank AVTC organizers, mentors, and sponsors for their tireless efforts in organizing and sustaining an incredible program that has had a tremendous impact on the development of students. Finally, my sincerest gratitude goes out to all teammates, colleagues, and coops I've had the pleasure of working with throughout my journey.

Aidan O'Gorman
Asad Bhatti
Bade Akinsanya
Ben Davies
Chris Shum
Cole Powers
Cole Tofflemire
David Hernandez
Dinil Karunaratne
Haocheng Zhang
Jack Mo
Jake McGrory
Jaskaran Basson
Jerica Rattana
Julian Lowrey
Jusroop Sangha
Kelly Zheng
Kien Tran
Majdi Khalil
Michael Wu
Patrick DiGioacchinno
Paul Boctor
Paul McInnis
Priyanka Verma
Radhika Kartha
Ramin Shaikhi
Sid Kakodkar
Sumeet Kler
Teo Crnobrnja
Tim Mui
Timothy Er
Urban Pistek
Vaibhav Patel
Vatsalya Saini
Winnie Wu
Yiwei Wang

Sept 15, 2018, photo of the University of Waterloo EcoCAR team in front of the newly built E7 engineering building. (Author was the photographer)

Dedication

To my family

Table of Contents

List of Figures

List of Tables

List of Abbreviations

VD	Vehicle Dynamics
VDC	Vehicle Dynamics Control
ECU	Engine Control Unit
V2X	Vehicle-to-Everything
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2P	Vehicle-to-Pedestrian
OTA	Over the air
NHTSA	National Highway Traffic Safety Association
AEB	Automatic Emergency Braking
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
UNECE	United Nations Economics Commission for Europe
LKA	Lane Keep Assist
CAV	Connected and Automated Vehicles
TV	Torque Vectoring
DOF	Degree of Freedom
OG	Original (equation)
SMA	Small Angle Approximation
CGH	Center of Gravity Height
CG	Center of Gravity
PID	Proportional Integral Derivative
MPC	Model Predictive Control
RRT	Rapidly Exploring Random Tree
A*	A-Star
NRT	Near Real-Time
RT	Real-Time
QP	Quadratic Programming
CA	Control Allocator
IMU	Inertia Measurement Unit

Chapter 1 - Introduction

1.1 Motivation

The proliferation of automated vehicle technologies has opened new possibilities in the way vehicle dynamics is conceived and with it, an opportunity to improve ride, handling, and efficiency presents itself [1]. Simultaneously, the advancement in compute technology has enabled high performance onboard processing to be possible on today's vehicles allowing for sophisticated sensing and optimization algorithms to be deployed on vehicles and updated routinely over-the-air (OTA) [2]. Vehicle-to-Everything (V2X) communication is another up-and-coming technology that is being researched and developed. These technologies and will have a major impact on road safety, traffic efficiency, and energy consumption [2]. Limit handling situations arise organically in traffic for reasons such as slippage due to adverse weather conditions, animals jumping onto the road, and children running into traffic. Most human drivers are not well trained to handle situations like these and the result is often injury or property damage when these situations arise [3].

According to the National Highway Traffic Safety Administration (NHTSA), nearly all car crashes are due to human errors. 94 percent of serious accidents are due to humans making dangerous choices on the road, and nearly 37,000 people lost their lives in crashes on the U.S. highways in 2016 [4].

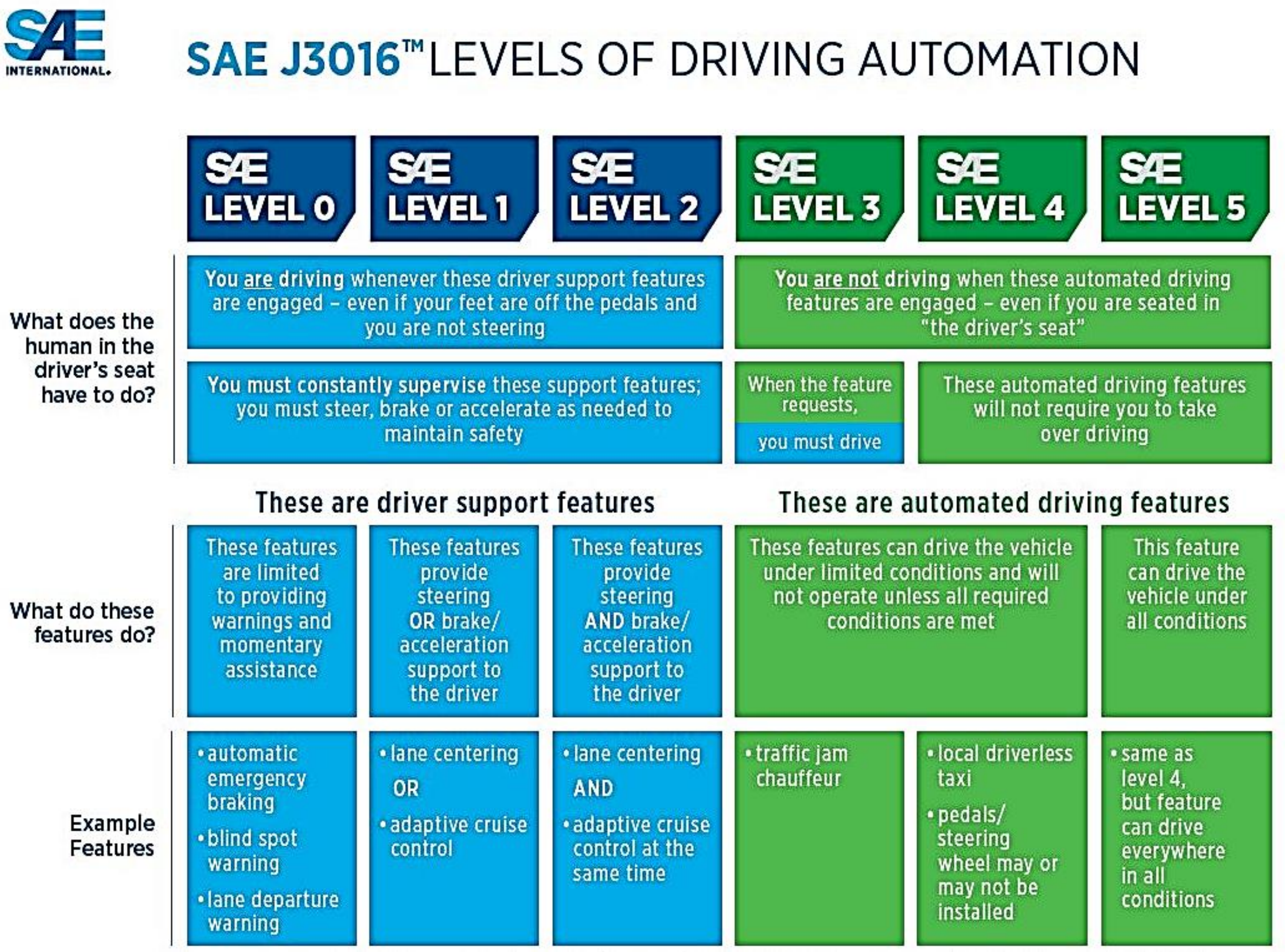
By reducing or eliminating the human factor from the equation, there is a potential to significantly reduce the number of fatalities that occur as a result of human errors [4]. For the reasons mentioned, many universities and automotive companies have made large investments to rapidly develop technologies in this arena.

Automotive companies around the world are racing to develop this technology and gain market share. Partnerships and recruitment from traditionally software and hardware-oriented companies such as Google, Intel and Nvidia are becoming ever more prevalent as autonomous technologies rely on advanced software and hardware systems which serve as the backbone of the technology for partial or full autonomy [5] [6] [7].

In 2014, the Society of Automotive Engineers (SAE) published a system for classifying varying levels of driving automation [8]. The most up-to-date chart at the writing of this paper is shown in Figure 1.

Six different levels of automation were proposed, ranging from Level 0 to Level 5. Each of these levels indicate the degree of vehicle automation with Level 0 vehicles supporting primarily warnings and momentary intervention and Level 5 vehicles supporting full autonomy and can perform as well or better than a human driver in every situation. The United Nations Economics Commission for Europe (UNECE) has adopted rules that would require automatic emergency braking (AEB) by 2020 [9]. The United States National Highway Traffic Safety Administration (NHTSA) published guidelines in 2016 calling on automakers to adopt this technology in their vehicles. Over 20 automakers agreed to implement the technology in their vehicles by 2020 [9]. Other advanced automated features such as Adaptive Cruise Control (ACC) and Automated Lane Keep Assist (LKA) are also continuing to be adopted in greater numbers by automakers.

In December of 2018, Waymo was the first to commercialize a fully autonomous taxi service in the United States [10]. General Motors is another leader in the automated vehicle space, embracing new activities and businesses that are emerging such as car-sharing as part of the new “gig economy”. The vehicles will feature Advanced Driver Assistance Systems (ADAS) for improving driver experience and vehicle safety by offering features such as automated lighting, adaptive cruise control, lane keep assist, lane centering, and lane departure warning systems which can help to reduce human error. Vehicle-to-Everything (V2X) communication is also being rapidly developed and is gaining traction due to its potential to enable vehicles to communicate with infrastructure, other vehicles, and pedestrians. Enveloping communication on this scale will enable cooperative driving, enabling safer and more efficient transport of people and goods. In August 2014, the NHTSA published a document in the U.S. Federal Register arguing the safety benefits of V2X systems and proposed a mandatory introduction, recognizing that V2X systems can only work if it was widely adopted [11].



SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering OR • adaptive cruise control	• lane centering AND • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 1: Society of Automotive Engineers Classification of Automated Vehicles [11].

This thesis is based around the University of Waterloo EcoCAR Team (UWAFT) and the EcoCAR Mobility Challenge. The overall objective of the competition is to design and build a hybrid electric vehicle with SAE Level 2 Autonomous capability. The vehicle platform used in this thesis was based on the 2019 Chevrolet Blazer – the vehicle that General Motors has donated to UWAFT as part of the EcoCAR Mobility Challenge. The overall objective of this thesis is to support the development of automated vehicle systems for the UWAFT Blazer.

1.2 Project Definition

In particular, the objective of this thesis is comprised of three parts:

First, various vehicle models are considered and developed. These vehicle models vary in the degree of fidelity and assumptions made. The utility of the models range in accuracy and complexity, making some models suitable for simulation and others more suitable for designing vehicle dynamics controllers as will be discussed in subsequent sections.

Second, various control architectures and strategies for automated vehicles with active steering and torque vectoring capability in a connected environment will be developed and evaluated.

Third, a visualization framework is explored in detail and highlights a means of creating virtual environments for developing CAV systems.

This paper does not catalog all existing work in vehicle modelling and control but presents a selection that can be useful for the development of the UWAFT Blazer vehicle with advanced driver assistance systems, utilizing information passed down from V2X systems.

Chapter 2 – Model Development

1.3 Vehicle Modelling

In terms of vehicle dynamics, there are various aspects of the vehicle that can be modelled ranging from system level models of the vehicle to component level models that capture the intricacies of components such as the suspension links and tires. There are also a variety of modelling approaches that can taken for the various aspects, each with their own unique set of benefits, assumptions and limitations. For automated vehicles controllers, a fast and accurate vehicle dynamics model is required to ensure that vehicle predictions can be made in a timely manner. For lab-based studies, a slower, but higher fidelity vehicle dynamics is used for simulation of the vehicle; this is required to accurately test and vet vehicle dynamics controllers prior to field testing. Controllers are only as good as the models they are trained on; hence it is imperative that models and simulations accurately reflect reality.

Vehicle dynamics models are required for three main purposes in this thesis:

1. Simulating the vehicle with a high degree of accuracy
2. Tuning of classical controllers
3. Use in predictive control for optimal control-based controllers

While the kinematic state estimation is useful for estimating current vehicle states, it is not able to reliably predict the future behavior of the vehicle as there are no dynamic considerations (road friction coefficients, lateral and longitudinal load transfers, etc). However, the vehicle state estimation information can be used to validate the accuracy of a dynamic model of the vehicle running simultaneously, in real-time. The vehicle dynamics model is used to optimize the commands of the TV controller by minimizing a cost-function, whose metrics include power consumption, vehicle response (lag, jerkiness, and oscillation) and ability to achieve commanded desired vehicle state.

This chapter will explore various vehicle dynamics models and their role in the development of an automated vehicle.

The vehicle models developed were based on the best estimate of the new mass properties of the modified Blazer. The controls discussed in this thesis were also based on the modified Blazer, however some parts have been abstracted and extended to a version of the vehicle with features that extend beyond the scope of the competition but would be useful when generalized for modern production vehicle applications.

1.3.1 Point Mass Model (2 DOF)

A point mass model is simply a representation of the vehicle in the form of a lumped mass with translational inertia but effectively no moment of inertia. It can be visualized as a single point with a tire governing the grip limits of the vehicle model, as shown in Figure 2. The point mass model was developed for use with the high-level trajectory planning controller and captures the effects of planar dynamics. The simplicity of the model allows a controller utilizing a point mass model as a vehicle prediction model to operate over a long time horizon and be updated in near-real time. The particular use of a point mass model is to generate the appropriate vehicle trajectory for the vehicle and pass it down to lower level controllers. This allows the vehicles trajectory to be optimized such that the vehicle would operate within the limits of handling, while also minimizing energy consumption due to the long look over the prediction horizon. The information from a the V2X system allows the controller to take into account the motion of adjacent vehicles as well as incorporate the timing of traffic infrastructure to minimize energy usage in the trajectory planning.

Figure 2 - Point mass representation; a single lumped mass with a tire acting as the limiting factor in providing grip.

The general dynamics equations for a point mass object in 3-dimensional space is as follows:

$$F = ma \tag{2.1}$$

$$\frac{d}{dt}(mv) = m\frac{dv}{dt} = F \tag{2.2}$$

$$F = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{2.3}$$

$$\frac{dv}{dt} = \dot{v} = \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \frac{1}{m} I_3 F = \begin{bmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{2.4}$$

$$F = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \left[F_{gravity} + F_{aerodynamics} + F_{wheel}\right] \tag{2.5}$$

$$\frac{dr}{dt} = \dot{r} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.6}$$

1.3.2 Bicycle Model (3 DOF)

A bicycle model, also known as a single-track model, is a rigid body representation of the vehicle that includes two wheels, one located at the front and one located at the back. The front wheel is able to steer and uses a tire model to capture cornering dynamics. The overall model has 3 degrees of freedom: x-translation, y-translation, and yaw-rotation. The bicycle model is a reduced-order model and was developed as a step up from the point mass model to be used as a benchmark for validating higher order models; it is not used directly in any controllers. A schematic of a bicycle model is shown in Figure 3.

There are many variations of the bicycle model, each with their own set of assumptions and take on various levels of complexity. Two major approaches can be taken to describe the vehicle motion for a bicycle model: a kinematic approach and a dynamic approach. The kinematic approach describes vehicle motion strictly by the geometry of the system, whereas a dynamic approach considers the forces generated by the tires and applies the appropriate moments and forces to the vehicle body to generate motion. The following equations describe the handling dynamics of a rigid vehicle [12] [13] [14].

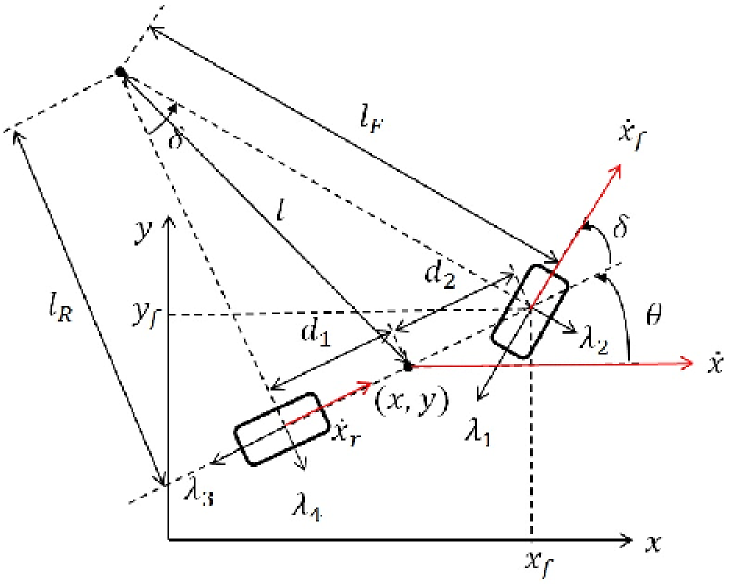


Figure 3 - Bicycle model with instantaneous center of rotation depicted [15]

Starting with the general equations of motion of a body in an inertial frame, the following equations apply:

$$m\ddot{X} = \sum F_X \qquad 2.7$$

$$m\ddot{Y} = \sum F_Y \qquad 2.8$$

$$J_z\ddot{\psi} = \sum M_Z \tag{2.9}$$

However, most forces and velocities are given or known in vehicle body coordinates; a rotation matrix is used to convert between the inertial frame and the body reference frame:

$$\begin{bmatrix} \sum F_X \\ \sum F_Y \end{bmatrix} = \begin{bmatrix} cos\psi & -sin\psi \\ sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} \sum F_x \\ \sum F_y \end{bmatrix} \tag{2.10}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} cos\psi & -sin\psi \\ sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.11}$$

Using these two equations, the equations of motion in the X-direction of the inertial frame reduces to:

$$m\ddot{X} = m\frac{d}{dt}(ucos\psi - vsin\psi) \ = \sum F_x cos\psi - \sum F_y sin\psi \tag{2.12}$$

$$= m\left(\dot{u}cos\psi - u\dot{\psi}sin\psi - \dot{v}sin\psi - v\dot{\psi}cos\psi\right) = \sum F_x cos\psi - \sum F_y sin\psi \tag{2.13}$$

Similarly, the equations of motion in the Y-direction of the inertial frame reduces to:

$$m\ddot{Y} = m\frac{d}{dt}(usin\psi + vcos\psi) \ = \sum F_x sin\psi + \sum F_y cos\psi \tag{2.14}$$

$$= m\left(\dot{u}sin\psi + u\dot{\psi}cos\psi + \dot{v}cos\psi - v\dot{\psi}sin\psi\right) = \sum F_x sin\psi + \sum F_y cos\psi$$

Then, solving the forces in the body frame:

$$m\left(\dot{u} - \dot{\psi}v\right) = \sum F_x \tag{2.15}$$

$$m\left(\dot{v} + \dot{\psi}u\right) = \sum F_y \tag{2.16}$$

$$J_z\ddot{\psi} = \sum M_z \tag{2.17}$$

1.3.3 Rigid Two Track Model (3 DOF)

A rigid two track model is very similar to the bicycle model in that it has the same 3 degrees of freedom, but with the addition of an additional set of wheels bringing the total wheel count to 4, located in the same positions as a regular 4-wheeled vehicle. This is shown in Figure 4. The advantage of this model is that it includes dynamics for steering and delivering torque to all 4 tires independently. The persistence of a track width allows the modelling of torque vectoring capabilities. This vehicle model is used in the vehicle regulation controller, which will be discussed in further detail in Chapter 3 – Controller Design.

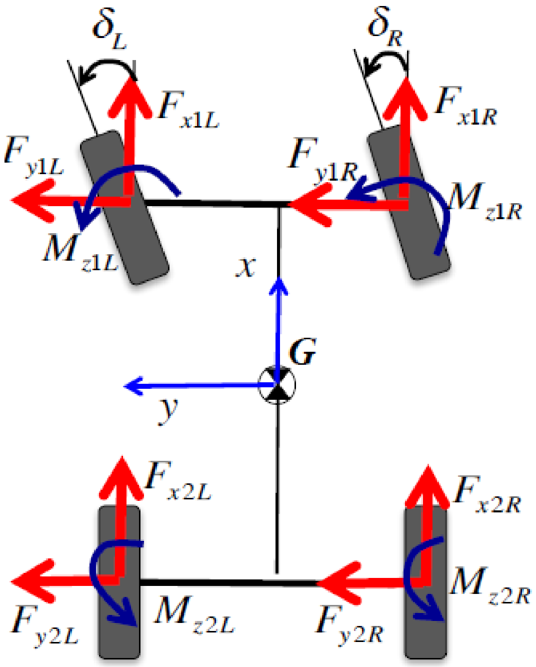


Figure 4 – Rigid two-track model representation of a vehicle with four wheels [16]

The equations of motion for this model is the same as the equations for the bicycle model. The difference is the presence of additional terms in the summation formulation that allow for additional yawing capability due to the nature of having track width in the model, as shown here:

$$\sum M_z = \sum_i F_{xi_t} \sin\delta_i x_i + \sum_i F_{yi_t} cos\delta_i x_i - \sum_i \cos\delta_i y_i + \sum_i F_{yi_t} sin\delta_i y_i \qquad 2.18$$

Where (x_i, y_i) are the coordinates for the tire center in the body reference frame.

The tire forces are generally nonlinear functions of the tire states: slip, normal load, road adhesion, as shown in the following:

$$F_{xi_t} = f\left(\alpha_i, \sigma_i, F_{zi_t}, \dots\right) \quad 2.19$$

$$F_{yi_t} = f\left(\alpha_i, \sigma_i, F_{zi_t}, \dots\right) \quad 2.20$$

Additional details about tire modelling can be found in section 1.3.7.

Due to the nonlinearity, dependence of the tire forces on the slip angles, and heavy product coupling between terms, a closed form solution is generally not possible. Numerical methods are generally the only solution, which are solved easily with MATLAB/Simulink or similar tools. It is important to note the results are only valid for the vehicle whose parameters are used.

1.3.4 9 DOF Vehicle Model

A 9 DOF model is a vehicle model that is another step above a rigid two track model. It uses the same equations and includes the same degrees of freedom but incorporates 6 additional degrees of freedom: pitch, roll, and 4 rotational degrees of freedom to account for each of the 4 wheels. The reason why the z translational degree of freedom is neglected is because the road is assumed to be flat with road elevation and road imperfections neglected. This model was used for simulation of the vehicle in a lab setting, not for use in any of the controllers to perform predictive control as it is too complicated to be run in real-time. A diagram showing the full vehicle model is shown in Figure 5.

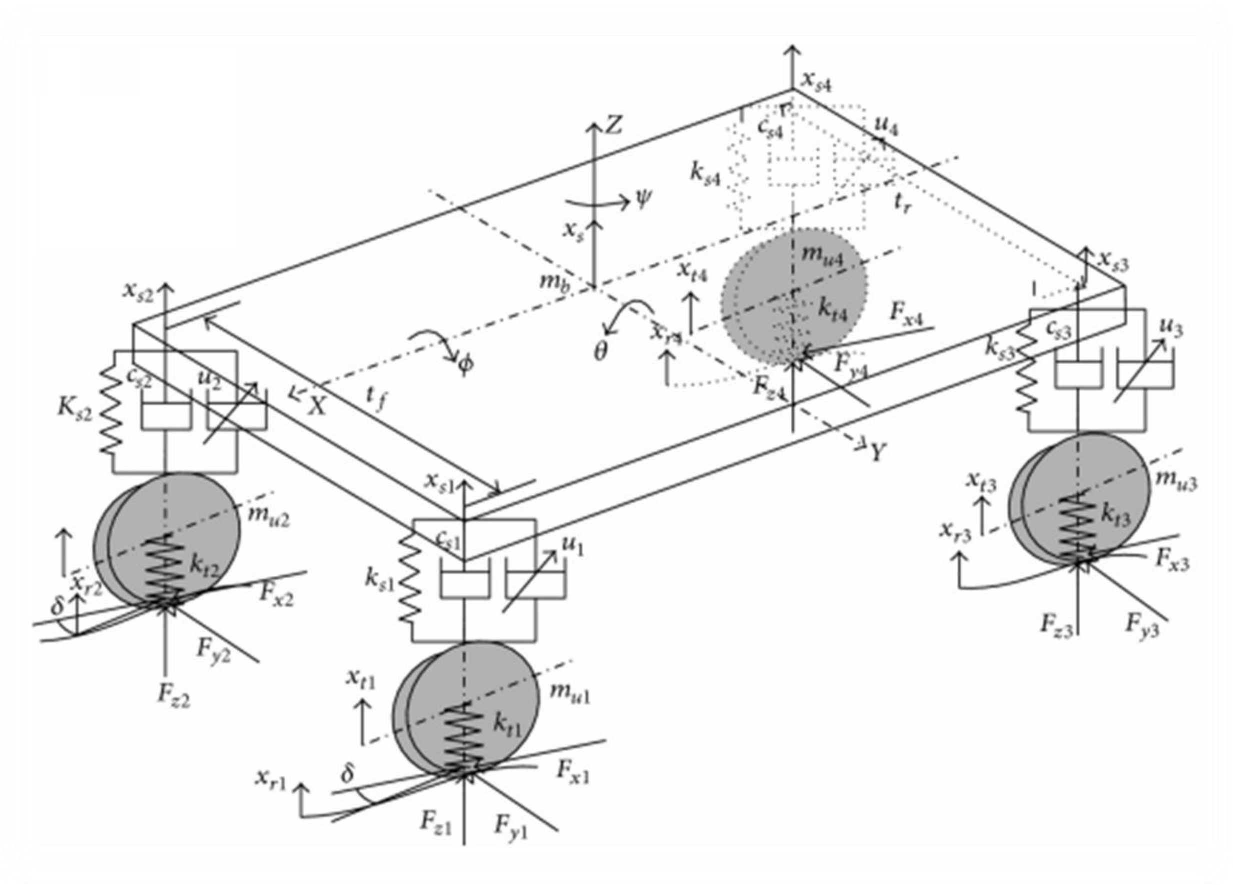


Figure 5 - 9 Degree of Freedom Vehicle Model [17]

The full set of equations in expanded form, including load transfer, with and without small angle approximation, are as follows:

<u>X axis Dynamics</u> –

The net force in X direction in the inertial reference frame is calculated based on two methods:

Using the full method (OG) and using small angle approximation (SMA). The fWx and fWy notation shows that these forces are in wheel coordinate frame for the x and y directions, respectively. The equations of motion in X direction are as follows -

a) OG

$$Fx_1 = cos\,\varphi * (fWx1 * cos\delta_1 - fWy1 * sin\delta_1) + sin\,\varphi * (-fWx1 * sin\delta_1 - \ fWy1 * cos\delta_1) \quad 2.21$$

$$Fx_2 = cos\,\varphi * (fWx2 * cos\delta_2 - fWy2 * sin\delta_2) + sin\,\varphi * (-fWx2 * sin\delta_2 - \ fWy2 * cos\delta_2) \quad 2.22$$

$$Fx_3 = cos\,\varphi * (fWx3 * cos\delta_3 - fWy3 * sin\delta_3) + sin\,\varphi * (-fWx3 * sin\delta_3 - \ fWy3 * cos\delta_3) \quad 2.23$$

$$Fx_4 = cos\,\varphi * (fWx4 * cos\delta_4 - fWy4 * sin\delta_4) + sin\,\varphi * (-fWx4 * sin\delta_4 - \ fWy4 * cos\delta_4) \quad 2.24$$

$$Fx_{Net} = Fx_1 + Fx_2 + Fx_3 + Fx_4 \quad 2.25$$

b) SMA

$$Fx_1 = sin\,\varphi * (\,fWy1) \quad \underline{2.26}$$

$$Fx_2 = sin\varphi * (\,fWy2) \quad \underline{2.27}$$

$$Fx_3 = \sin\varphi * (\,fWy3) \quad \underline{2.28}$$

$$Fx_4 = sin\,\varphi * (\,fWy4) \quad \underline{2.29}$$

$$Fx_{Net} = Fx_1 + Fx_2 + Fx_3 + Fx_4 \tag{2.30}$$

Y axis Dynamics –

As X axis dynamics, the net force in Y direction in the inertial reference frame is calculated based on two methods: SMA and OG. The equations of motion in Y direction are as follows -

c) OG

$$Fy_1 = sin\,\varphi * (fWx1 * cos\delta_1 - fWy1 * sin\delta_1) + cos\,\varphi * (fWx1 * sin\delta_1 + \ fWy1 * cos\delta_1) \tag{2.31}$$

$$Fy_2 = sin\,\varphi * (fWx2 * cos\delta_2 - fWy2 * sin\delta_2) + cos\,\varphi * (fWx2 * sin\delta_2 + \ fWy2 * cos\delta_2) \tag{2.32}$$

$$Fy_3 = sin\,\varphi * (fWx3 * cos\delta_3 - fWy3 * sin\delta_3) + cos\,\varphi * (fWx3 * sin\delta_3 + \ fWy3 * cos\delta_3) \tag{2.33}$$

$$Fy_4 = sin\,\varphi * (fWx4 * cos\delta_4 - fWy4 * sin\delta_4) + cos\,\varphi * (fWx4 * sin\delta_4 + \ fWy4 * cos\delta_4) \tag{2.34}$$

$$Fy_{Net} = Fy_1 + Fy_2 + Fy_3 + Fy_4 \tag{2.35}$$

d) SMA

$$Fy_1 = cos\,\varphi * (\ fWy1) \tag{2.36}$$

$$Fy_2 = cos\varphi * (fWy2) \tag{2.37}$$

$$Fy_3 = \cos\varphi * (fWy3) \tag{2.38}$$

$$Fy_4 = cos\,\varphi * (fWy4) \tag{2.39}$$

$$Fy_{Net} = Fy_1 + Fy_2 + Fy_3 + Fy_4 \tag{2.40}$$

The net forces in X axis and Y axis are in inertial reference frame. To convert these forces in body reference frame. The coordinate transformation is performed on velocity and acceleration in both directions individually.

The transformation matrix between inertial reference frame and body reference frame is defined as below:

$$\begin{bmatrix} Cx \\ Cy \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & cos\varphi \end{bmatrix} \begin{bmatrix} Ex \\ Ey \end{bmatrix} \tag{2.41}$$

The transformed velocity and acceleration are used as vehicle feedback for further calculations.

<u>Load Transfer</u> –

Load transfer included the suspension contribution during acceleration/braking and cornering. The load transfer is calculated for each wheel and then the same is subtracted from the static normal force. The effect of anti-roll bars and motion ratios are included in the formulation.

$$\Delta Fz_1 = -kS1 * \left\{(-a * sin\theta) + \left(\frac{trf}{2} * \sin\emptyset * mrSF^2\right)\right\} + (-C1) * \left\{(-a * \dot{\theta} * cos\theta) + \left(\frac{trf}{2} * \dot{\emptyset}\, cos\emptyset * mrDF^2\right)\right\} + \left\{\frac{(-kAF * \emptyset * mAF)}{2}\right\} \quad 2.42$$

$$\Delta Fz_2 = -kS2 * \left\{(-a * sin\theta) + \left(-\frac{trf}{2} * \sin\emptyset * mrSF^2\right)\right\} + (-C2) * \left\{(-a * \dot{\theta} * cos\theta) + \left(-\frac{trf}{2} * \dot{\emptyset}\, cos\emptyset * mrDF^2\right)\right. + \left\{\frac{(kAF * \emptyset * mAF)}{2}\right\} \quad 2.43$$

$$\Delta Fz_3 = -kS3 * \left\{(b * sin\theta) + \left(\frac{trf}{2} * \sin\emptyset * mrSF^2\right)\right\} + (-C3) * \left\{(b * \dot{\theta} * cos\theta) + \left(\frac{trf}{2} * \dot{\emptyset}\, cos\emptyset * mrDF^2\right)\right\} + \left\{\frac{(-kAR * \emptyset * mAR)}{2}\right\} \quad 2.44$$

$$\Delta Fz_4 = -kS4 * \left\{(b * sin\theta) + \left(\frac{trf}{2} * \sin\emptyset * mrSF^2\right)\right\} + (-C4) * \left\{(b * \dot{\theta} * cos\theta) + \left(\frac{trf}{2} * \dot{\emptyset}\, cos\emptyset * mrDF^2\right)\right\} + \left\{\frac{(kAR * \emptyset * mAR)}{2}\right\} \quad 2.45$$

The weight transfer component is adjusted in static normal load to have dynamic normal load on individual wheel. The load transfer is applied based on the *loadtransfer* flag which can be defined in modelSetup script.

Pitch Dynamics –

The moments introduced due to different cases like suspension effects or drag forces are calculated for each wheel and pitch is calculated using net moment generation.

$$M_1 = -(K_{s1} * mrSF * a * sin\theta) - (c1 * mrDF * a * \dot{\theta} * cos\theta) \quad 2.46$$

$$M_2 = -(K_{s2} * mrSF * a * sin\theta) - (c2 * mrDF * a * \dot{\theta} * cos\theta) \quad 2.47$$

$$M_3 = -(K_{s3} * mrSF * a * sin\theta) - (c3 * mrDF * a * \dot{\theta} * cos\theta) \quad 2.48$$

$$M_4 = -(K_{s4} * mrSF * a * sin\theta) - (c4 * mrDF * a * \dot{\theta} * cos\theta) \qquad 2.49$$

$$M_{drag} = fx_{Drag} * h_\theta \qquad 2.50$$

$$M_A = -(mass * a_x * h_\theta * cos\theta) \qquad 2.51$$

$$M_g = -(mass * g * h_\theta * sin\theta) \qquad 2.52$$

$$M_{pitch} = M1 + M2 + M3 + M4 + M_{drag} + M_A + M_g \qquad 2.53$$

$$\ddot{\theta} = \frac{M_{pitch}}{I_{yy}} \qquad 2.54$$

<u>Roll Dynamics</u> –

In rolling, the moments are generated due to suspension effects, drag forces, acceleration and anti-roll bar effects. These moments are calculated using the following equations -

$$M1 = -\left(K_{s1} * mrSF * \frac{w_{SF}}{2} * sin\emptyset\right) - (c1 * mrDF * \frac{w_{DF}}{2} * \dot{\emptyset} * cos\emptyset) \qquad 2.55$$

$$M2 = -\left(K_{s2} * mrSF * \frac{w_{SF}}{2} * sin\emptyset\right) - (c2 * mrDF * \frac{w_{DF}}{2} * \dot{\emptyset} * cos\emptyset) \qquad 2.56$$

$$M3 = -\left(K_{s3} * mrSF * \frac{w_{SF}}{2} * sin\emptyset\right) - (c3 * mrDF * \frac{w_{DF}}{2} * \dot{\emptyset} * cos\emptyset) \quad 2.57$$

$$M4 = -\left(K_{s4} * mrSF * \frac{w_{SF}}{2} * sin\emptyset\right) - (c4 * mrDF * \frac{w_{DF}}{2} * \dot{\emptyset} * cos\emptyset) \quad 2.58$$

$$M_{drag} = -fy_{Drag} * h_{\emptyset} \quad 2.59$$

$$M_A = -\left(mass * a_y * h_{\emptyset} * cos\emptyset\right) \quad 2.60$$

$$M_g = -(mass * g * h_{\emptyset} * sin\emptyset) \quad 2.61$$

$$M_{ARF} = -(k_{AF} * mrAF * \emptyset) \quad 2.62$$

$$M_{ARR} = -(k_{AR} * mrAR * \emptyset) \quad 2.63$$

$$M_{roll} = M1 + M2 + M3 + M4 + M_{drag} + M_A + M_g + M_{ARF} + M_{ARR} \quad 2.64$$

$$\ddot{\emptyset} = \frac{M_{roll}}{I_{xx}} \quad 2.65$$

<u>Yaw Dynamics</u> –

The yaw angle is calculated using the two methods – SMA and OG.

a) OG

The following equations are used for calculating yaw angle -

$$M1 = \left\{ fx_1 * \left(-\frac{trf}{2} * cos\delta_1 + a * sin\delta_1 \right) \right\} + \left\{ fx_1 * \left(-\frac{trf}{2} * cos\delta_1 + a * sin\delta_1 \right) \right\} \tag{2.66}$$

$$M2 = \left\{ fx_2 * \left(-\frac{trf}{2} * cos\delta_2 + a * sin\delta_2 \right) \right\} + \left\{ fx_2 * \left(-\frac{trf}{2} * cos\delta_2 + a * sin\delta_2 \right) \right\} \tag{2.67}$$

$$M3 = \left\{ fx_3 * \left(-\frac{trf}{2} * cos\delta_3 + b * sin\delta_3 \right) \right\} + \left\{ fx_3 * \left(-\frac{trf}{2} * cos\delta_3 + b * sin\delta_3 \right) \right\} \tag{2.68}$$

$$M4 = \left\{ fx_4 * \left(-\frac{trf}{2} * cos\delta_4 + b * sin\delta_4 \right) \right\} + \left\{ fx_4 * \left(-\frac{trf}{2} * cos\delta_4 + b * sin\delta_4 \right) \right\} \tag{2.69}$$

$$M_{roll} = M1 + M2 + M3 + M4 \tag{2.70}$$

$$\ddot{\varphi} = \frac{M_{roll}}{I_{xx}} \tag{2.71}$$

b) SMA

The equations used for calculating yaw angle with small angle approximations are as below –

$$M1 = fy_1 * \left(\frac{trf}{2} * \delta_1 + a\right) \qquad 2.72$$

$$M2 = fy_2 * \left(\frac{trf}{2} * \delta_2 + a\right) \qquad 2.73$$

$$M3 = fy_3 * \left(\frac{trf}{2} * \delta_3 + b\right) \qquad 2.74$$

$$M4 = fy_4 * \left(\frac{trf}{2} * \delta_4 + b\right) \qquad 2.75$$

$$M_{roll} = M1 + M2 + M3 + M4 \qquad 2.76$$

$$\ddot{\varphi} = \frac{M_{roll}}{I_{xx}} \qquad 2.77$$

For yaw angle calculation, SMA flag is provided to switch yaw angle calculation from SMA to original. The switch is needed to define in modelSetup matlab file.

Integrating $\ddot{\theta}, \ddot{\emptyset}$ and $\ddot{\varphi}$, the pitch, roll and yaw angle are generated and are used in vehicle feedback.

To determine the forces at the tires, it is necessary to define some parameters from the chassis to feed into the tire model, namely body slip, tire slip angle, and tire slip ratio for each of the tires. The calculations for these parameters are as follows -

a) Body slip angle –

$$\beta = \tan^{-1}\frac{vCy}{vCx} \tag{2.78}$$

Where vCx and vCy are vehicle velocities in x and y direction respectively.

b) Slip angle –

The yaw rate effect is considered in the vehicle velocity which is in Car coordinate frame.

$$Vy = vCy + a * \dot{\varphi} \tag{2.79}$$

$$Vx = vCx - \frac{trf}{2} * \dot{\varphi} \tag{2.80}$$

These velocities are converted into Wheel coordinate frame to calculate the slip angles for each wheel.

$$\begin{bmatrix} vWx \\ vWy \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & cos\varphi \end{bmatrix} \begin{bmatrix} Vx \\ Vy \end{bmatrix} \tag{2.81}$$

$$\propto = \tan^{-1}\frac{vWy}{vWx} \tag{2.82}$$

c) Slip ratio –

After getting the wheel velocities, the following calculations are done to get the slip ratio for each wheel.

$$slip\ ratio = \frac{rwheel_F * wheelspeed - vWx}{\max\left(|vWx|, 0.001\right)} \qquad 2.83$$

1.3.5 ADAMS Model (N DOF)

An ADAMS Car model is a vehicle dynamics model created using MSC software. An ADAMS Car model is able to have many degrees of freedom without burdening the user with developing the mathematics behind it explicitly. Instead, the user can simply define the hard points of the suspension and the high-level information regarding the system such as the mass, inertia, and the spring and damping coefficients of certain components and the software can take care of the dynamics automatically. Once an ADAMS Car model is built, it can be exported as a plant to Simulink with predefined inputs and outputs to interface with the rest of the Simulink model. This is currently an ongoing development project and should be finished when more information about the UWAFT Blazer vehicle is revealed. Figure 6 shows the ADAMS Car model developed for the UWAFT Camaro for the EcoCAR 3 competition.

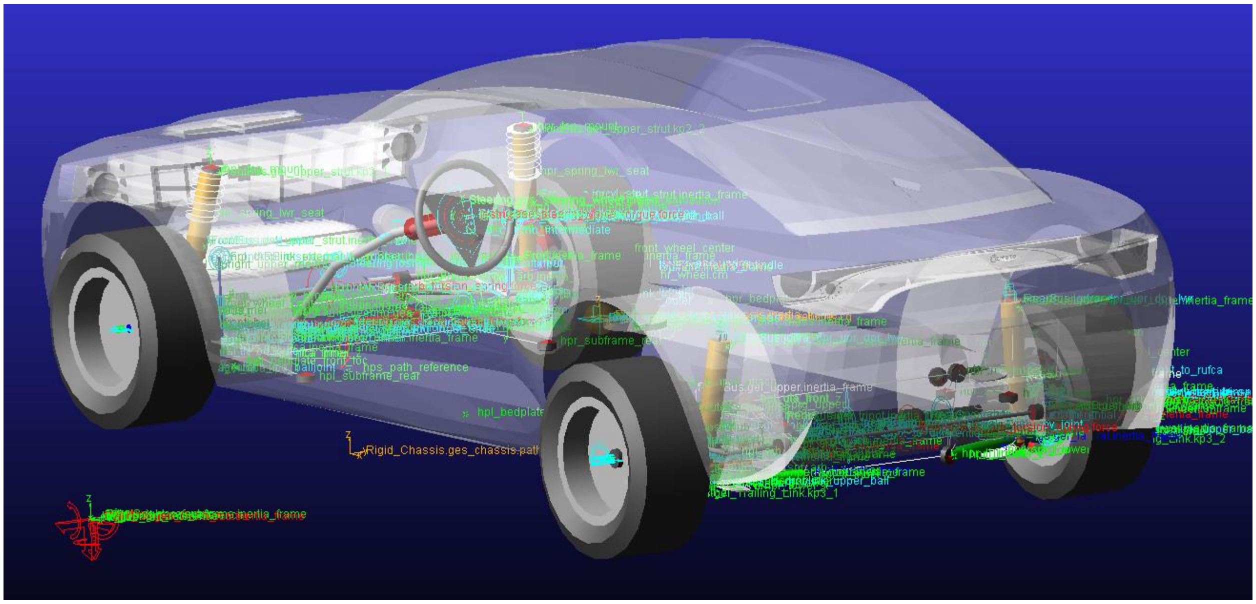


Figure 6 - ADAMS Car model developed for the UWAFT Camaro in the EcoCAR 3 competition

1.3.6 Motor Model

The motor model used was simply a look up table that contains the charge and discharge capabilities of the motor under specific voltages and speeds. This was included to ensure that the controller never exceeded the limits of capability of the motor for the speed the vehicle is travelling at. Data was not available for the UWAFT motor being used at the time of writing, so a generic set of motor data was used, as shown in Figure 7.

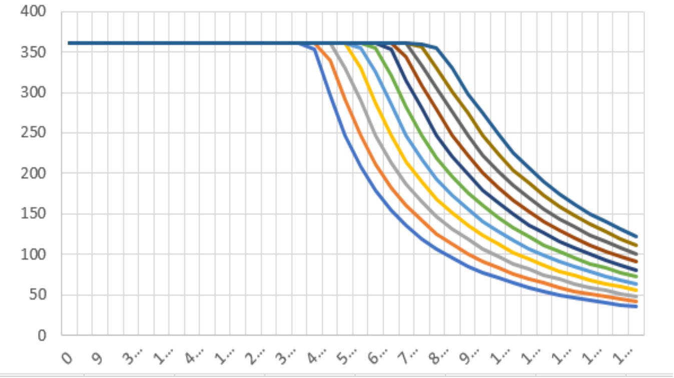


Figure 7 - Torque vs. Speed vs. Voltage curves for an Electric Motor

1.3.7 Tire Model

Tire modelling is a very difficult task. The reasons being that tires behave highly non-linearly and are time variant. The Pacejka tire model is the most common non-linear tire model that is used in industry. It was developed by Hans B. Pacejka over 20 years ago and has seen multiple iterations since. It is an empirical model that relies heavily on testing data to fit the model and has over 20 coefficients required for describing the lateral and longitudinal force, as well as self-alignment torque of the tire [18].

Some short-term time variance and non-linearities:

- The temperature of the tires and the environment
- Precipitation levels
- Dirt levels

- Static normal loading due to cargo
- Load Transfer
- Tire Pressure

Long-term time variance and non-linearities:

- Level of tread wear on the tire
- Wheel alignment (toe and camber)

It is currently impossible to capture all of these effects simultaneously with the tire models available today simply because there are too many variables. The Pacejka tire model is the most popular option and is very accurate under normal conditions.

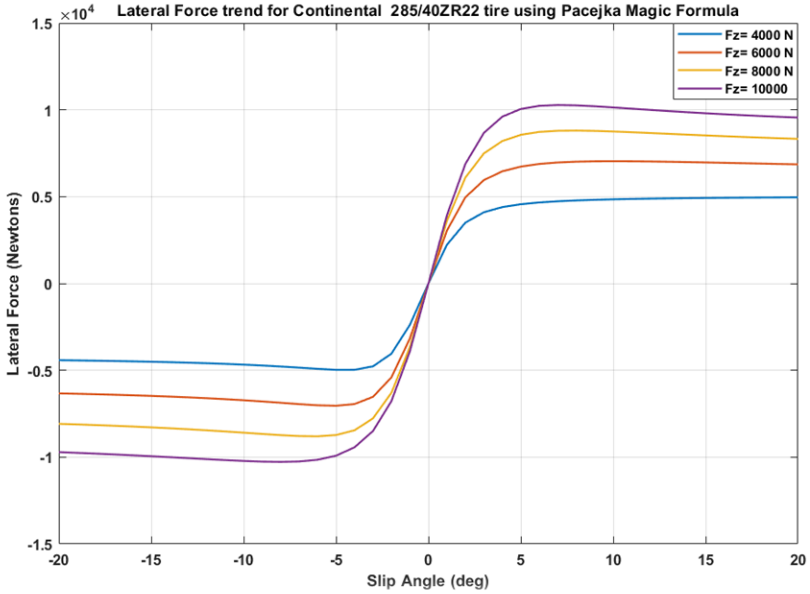


Figure 8 - Pacejka tire model showing the various curves representing the lateral loading capability of the tire as vertical load changes [19]

The tire was modelled using the Magic Formula 6.1 model created by the Tyre CAE Team within Chassis Engineering of Jaguar Land Rover [19].

The Magic Formula is a specific form for the tire characteristic function. It contains four dimensionless coefficients, B, C, D, E, or stiffness, shape, peak, and curvature that characterizes the tire. The general form of the Magic Formula is shown in the following [18]:

$$F_x = f(\kappa, F_z) = F_z \cdot D \cdot sin(C \cdot arctan\{B\kappa - E[B\kappa - arctan(B\kappa)]\}) \quad 2.84$$

Where:

$$df_z = \frac{F_z - F_{z0}}{F_z} \quad 2.85$$

$$\kappa_x = \kappa + S_{Hx} \quad 2.86$$

$$C_x = pCx1 \quad 2.87$$

$$D_x = \mu_x F_z \quad 2.88$$

$$\mu_x = pDx1 + pDx2 * df_z \quad 2.89$$

$$E_x = (pEx1 + pEx2 * df_z + pEx3 * df_z^2)[1 - pEx4 * sgn(\kappa_x)] \quad 2.90$$

$$K_{X\kappa} = F_z * (pKx1 + pKx2 * df_z) *^{Kx3*df_z} \quad 2.91$$

$$B_x = \frac{K_{X\kappa}}{C_x D_x + \epsilon_x} \quad 2.92$$

$$S_{Hx} = pHx1 + pHx2 * df_z \tag{2.93}$$

$$S_{VX} = F_z * (pVx1 + pVx2 * df_z) \tag{2.94}$$

Some common values for the B, C, D, and E coefficients are shown in Table 1.

Table 1 - Common Tire-Road Contact Coefficients [18]

Surface	B	C	D	E
Dry tarmac	10	1.9	1	0.97
Wet tarmac	12	2.3	0.82	1
Snow	5	2	0.3	1
Ice	4	2	0.1	1

The purpose of the Pacejka tire model is to determine the correct amount of combined lateral and longitudinal grip of a tire in response to tire loading. A table with the descriptions and units of the variables can be found in Appendix A

Tire Road Contact Variables.

A common method of simplifying the tire model is to consider a linear tire. A linear tire considers the initial slope of the slip angle vs lateral force curve and assumes a linear relationship. This is sufficiently accurate for situations with little deviation in normal force and low demand situations. It can be considered a compromise between a kinematic model and a fully developed Pacejka tire model. Tires performing in the linear region are considered to follow the following relation to provide the lateral force acting on the vehicle:

$$F_y = C_s * \propto \qquad 2.95$$

Where Cs is the cornering stiffness and ∝ is slip angle of the tire.

1.4 Model Validation

With model development, it is important that the model itself is validated otherwise the simulations are not useful as it may not representative of reality. Model validation requires data with a physical vehicle to compare the model against. Since the vehicle does not physically exist at the time of writing, the best option for model validation of the higher order models is to compare with lower order models. In lower demand situations, the behavior of higher order models and lower order models should be identical as dynamics are minimized and kinematics dominate.

This section will focus on the validation of the 9 DOF Vehicle Model. It is the most complex and intensive vehicle model derived in this paper.

1.4.1 9 DOF Vehicle Model Validation

A representation of the 9 degree of freedom vehicle dynamics model is shown in the Simulink block diagram in Figure 9. It contains the dynamics for X, Y translational dynamics, load transfer dynamics, as well as pitch, roll, and yaw rotational dynamics. These dynamics can be tested independently by "locking" the remaining degrees of freedom and subjecting the vehicle to a set of tests. To test the pitching dynamics, all the degrees of freedoms, with the exception of the pitching degree of freedom were locked. The vehicle was given a forward accelerative step in the x-translational direction and the response in the pitching direction was observed. The results of the vehicle pitching response is shown in Figure 10. The vehicle body experienced second-order dynamics behavior followed by a steady state final value. This was the expected response and when compared with hand calculations, the values matched. To test body roll, a similar testing procedure was used. All degrees of freedom were locked with the exception of the body roll degree of freedom. A step acceleration in the y-translation direction was injected into the system and the body roll response observed. Once again, the vehicle body experienced second-order dynamics behavior followed by a steady state final value. When compared with hand calculations, the results were comparable, as shown in Figure 11.

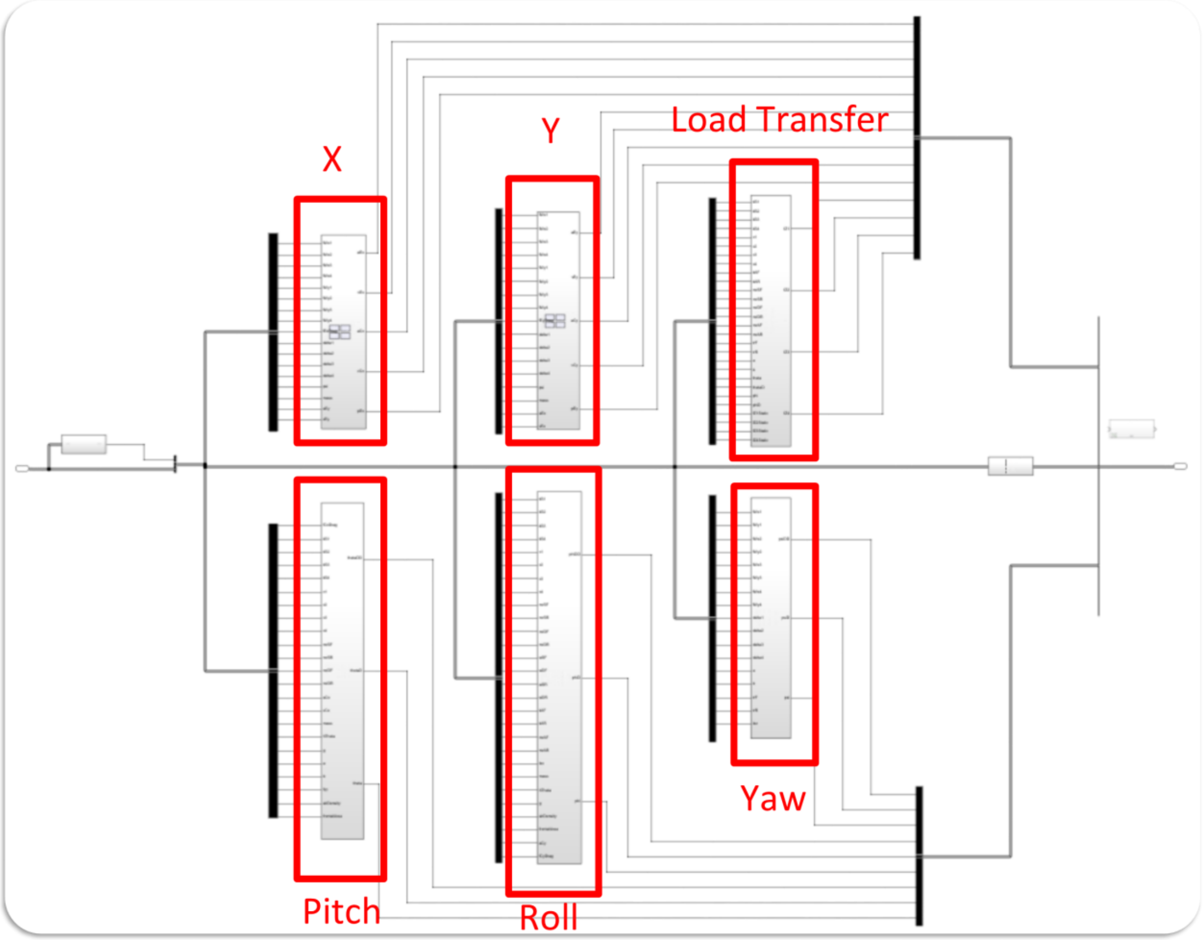
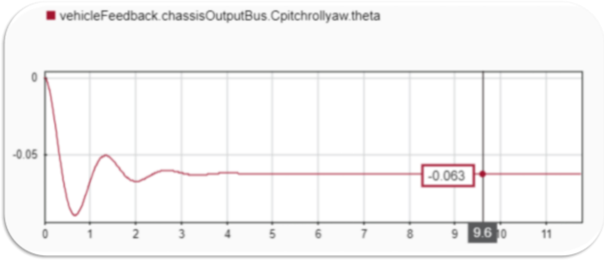


Figure 9 - Simulink block diagram for the vehicle dynamics model

Testing θ (Body Pitch):



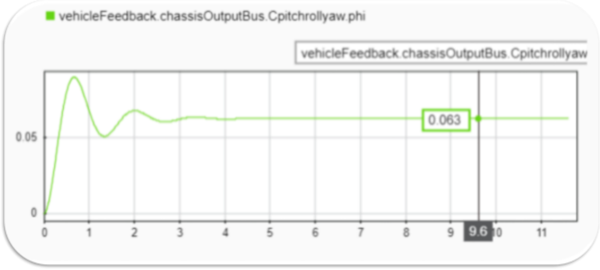
Pitch Response to Step Forward Acceleration

- Steady State Value Matches hand calculations ✓
- Equation for steady state pitch angle:

$$\theta = \sin^{-1}\left(\frac{mA_y h_\theta}{-2(k_{s_{Fr}}) - 2(k_{s_{Re}}) + mGh_\theta}\right) * \frac{180}{pi}$$

Figure 10 - Testing body pitch for the vehicle dynamics model

Testing ϕ (Body Roll):



Roll Response to Step Lateral Acceleration

- Steady State Value Matches hand calculations ✓
- Equation for steady state roll angle:

$$\phi = \sin^{-1}\left(\frac{mA_y h_\phi}{-2(k_{s_{Le}}) - 2(k_{s_{Ri}}) + mGh_\phi}\right) * \frac{180}{pi}$$

Figure 11 - Testing body roll for the vehicle dynamics model

It is important to note that this "verification" is only accurate from a modelling standpoint and not necessarily representative of the physical vehicle. Experimentation on the vehicle will be required to properly compare and correlate the model with the physical vehicle. This will need to done once the UWAFT team has finished modifications on the vehicle and all new components have been installed.

1.4.2 ADAMS Car Model Validation

MSC ADAMS Car is the primary multibody vehicle dynamics model used by UWAFT. It is built within the ADAMS Car environment and can be exported as a Simulink block for co-simulation in Simulink. The following highlights some of the validation approaches that were completed for the UWAFT Camaro for EcoCAR 3, as shown in Figure 12, Figure 13, Figure 14, Figure 15. The full vehicle can be simulated and validated by running the simulation in ADAMS Car, as shown in Figure 12 and Figure 15. Simulations such as double lane changes, fishhook maneuvers, and more can be completed in the software. The data can then be compared with physical experimentation to confirm model accuracy. Suspension kinematics can be analyzed using the suspension kinematics tool in ADAMS Car as shown in Figure 13 and Figure 14. This can provide data that can be used for developing lower-order models built in MATLAB or Simulink.

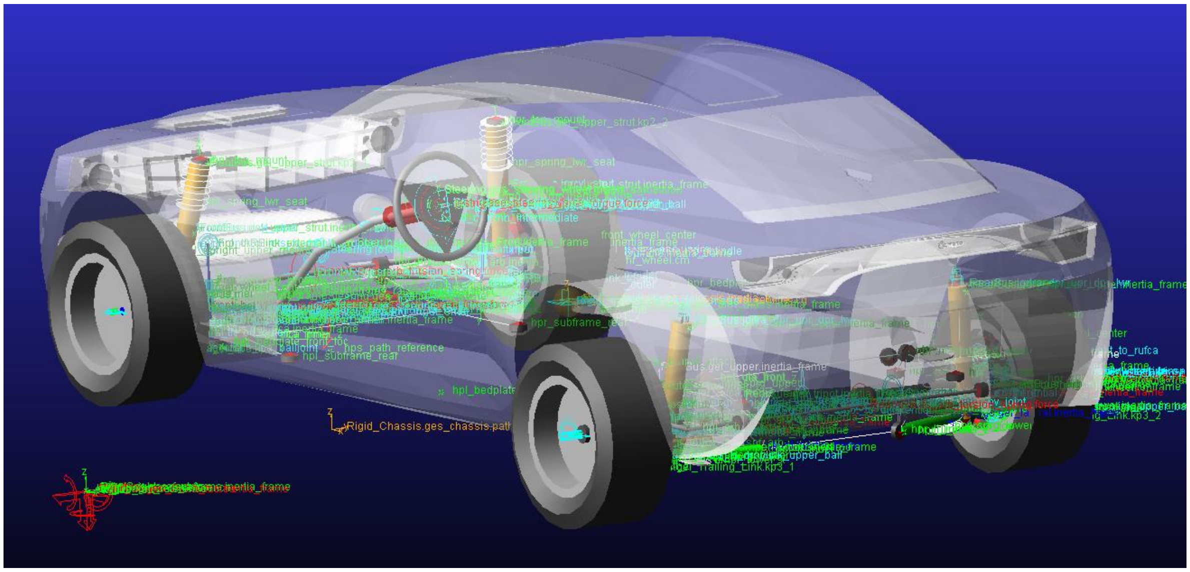
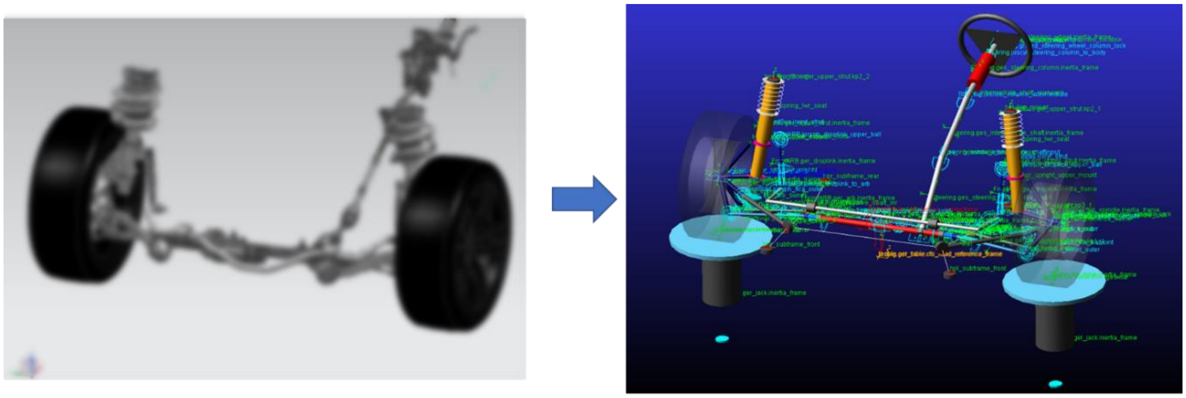


Figure 12 - ADAMS Car model built for the UWAFT Camaro



GM CAD

ADAMS CAR

Figure 13 - ADAMS Car suspension modelled by converting CAD hardpoints into a suspension model in ADAMS Car

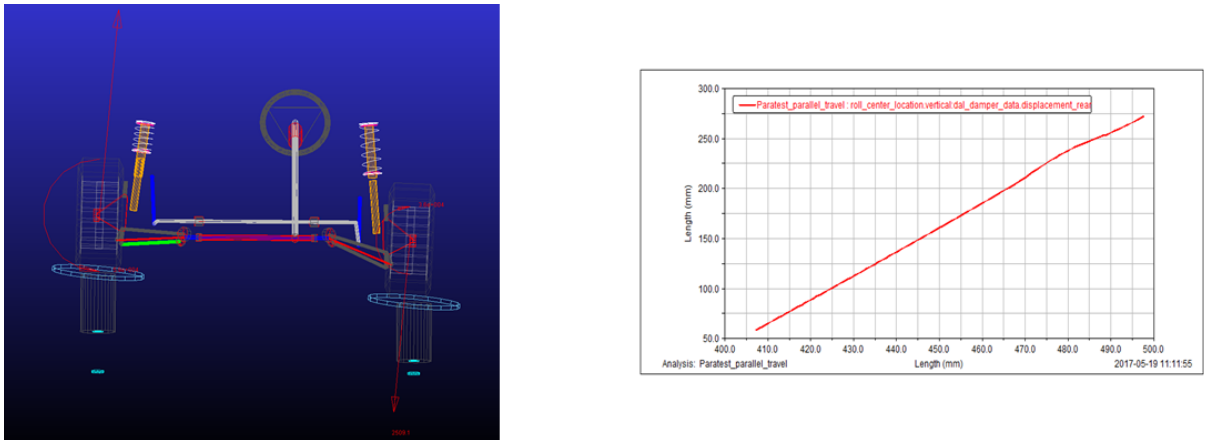


Figure 14 - Suspension kinematics were evaluated in ADAMS Car and the data was exported to fit the lower-order vehicle models

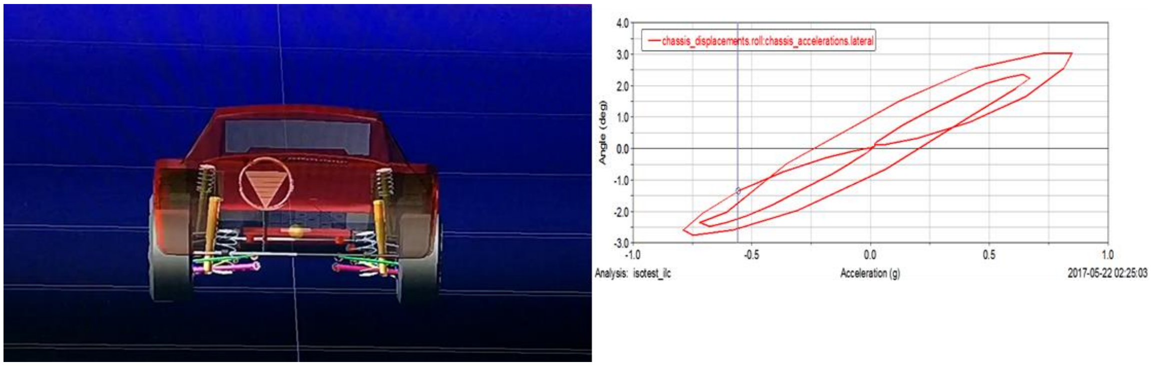


Figure 15 - Full vehicle analysis was performed in ADAMS Car; the data was used to fit lower-order vehicle models

A similar procedure will need to be completed for the UWAFT Blazer when it arrives on campus to validate the ADAMS Car model as well as lower-order models.

1.4.3 Suspension Characterization

UWAFT will utilize an aftermarket set of suspension components to compensate for the additional mass and altered mass distribution. It is important to characterize the suspension in order to supply vehicle models with the most accurate data for simulation. Figure 16 shows the damper dyno used at the Multimatic Technical Centre in Markham for characterizing the aftermarket suspension used for the UWAFT Camaro vehicle. Figure 17 and Figure 18 show the data collected for the front and rear dampers.

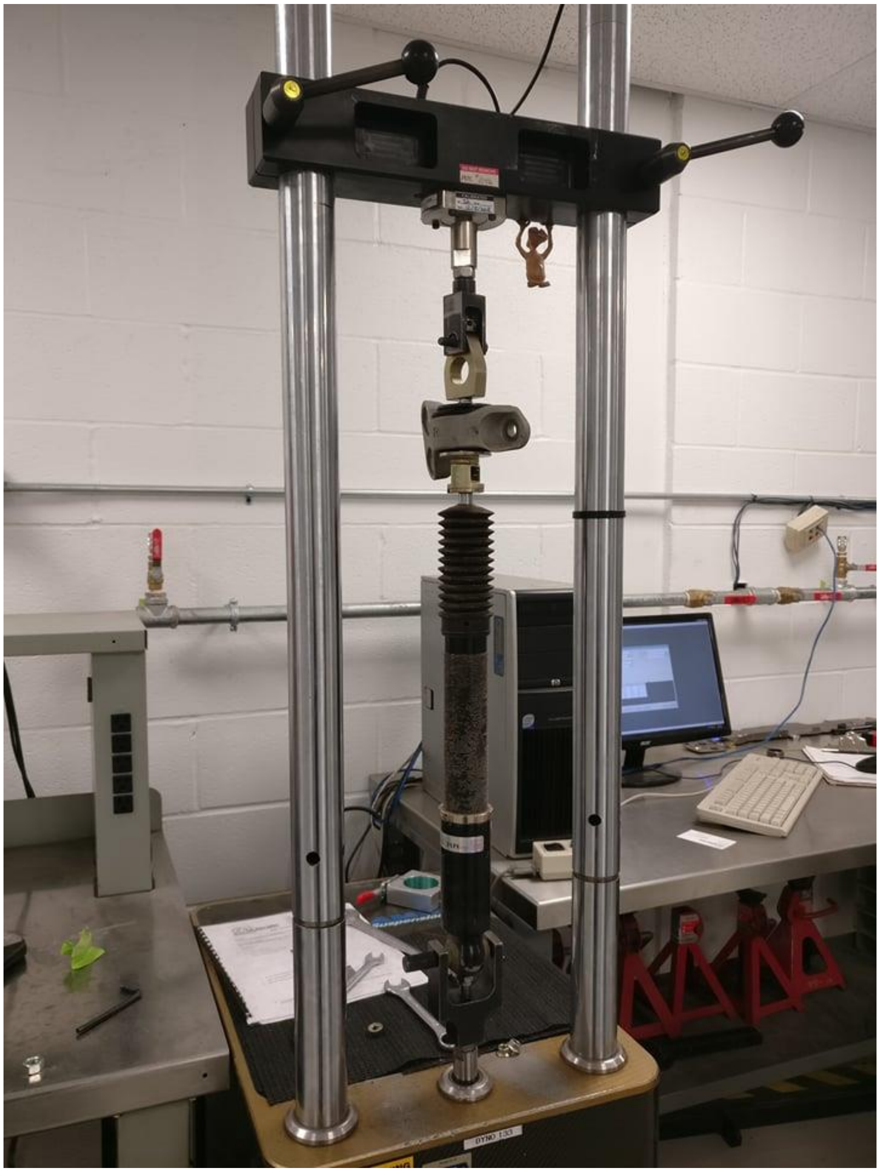


Figure 16 - Damper dyno at Multimatic Technical Centre, Markham

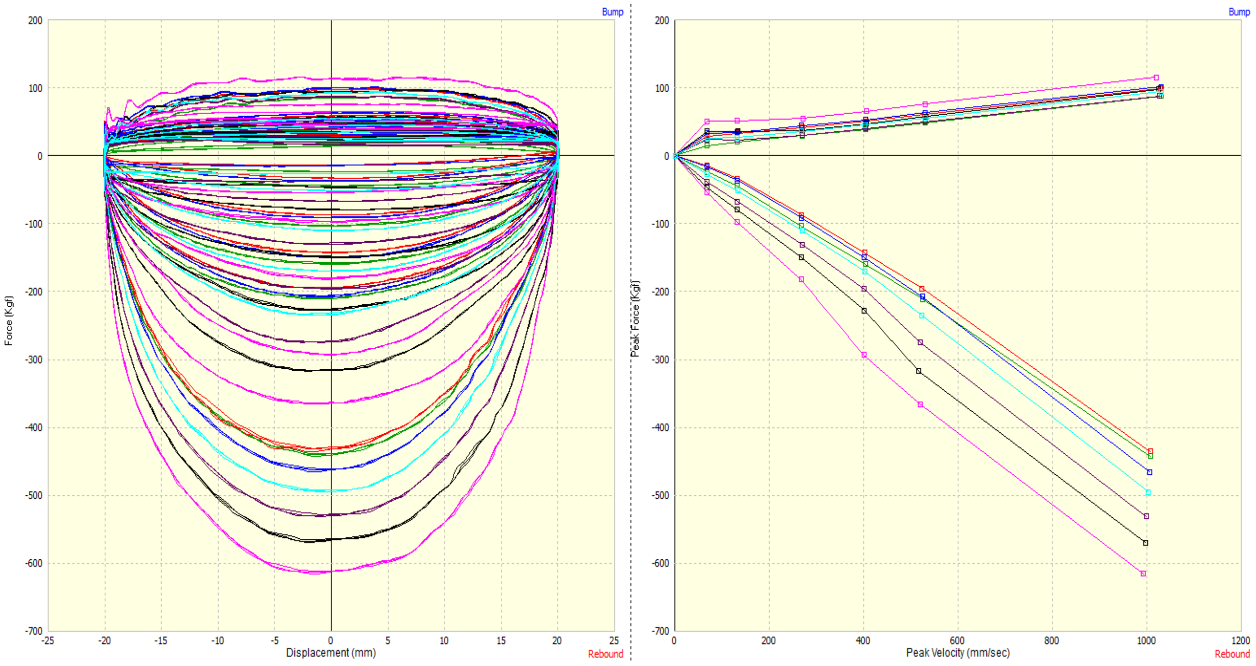


Figure 17 - Aftermarket front dampers tested at Multimatic Technical Centre, Markham

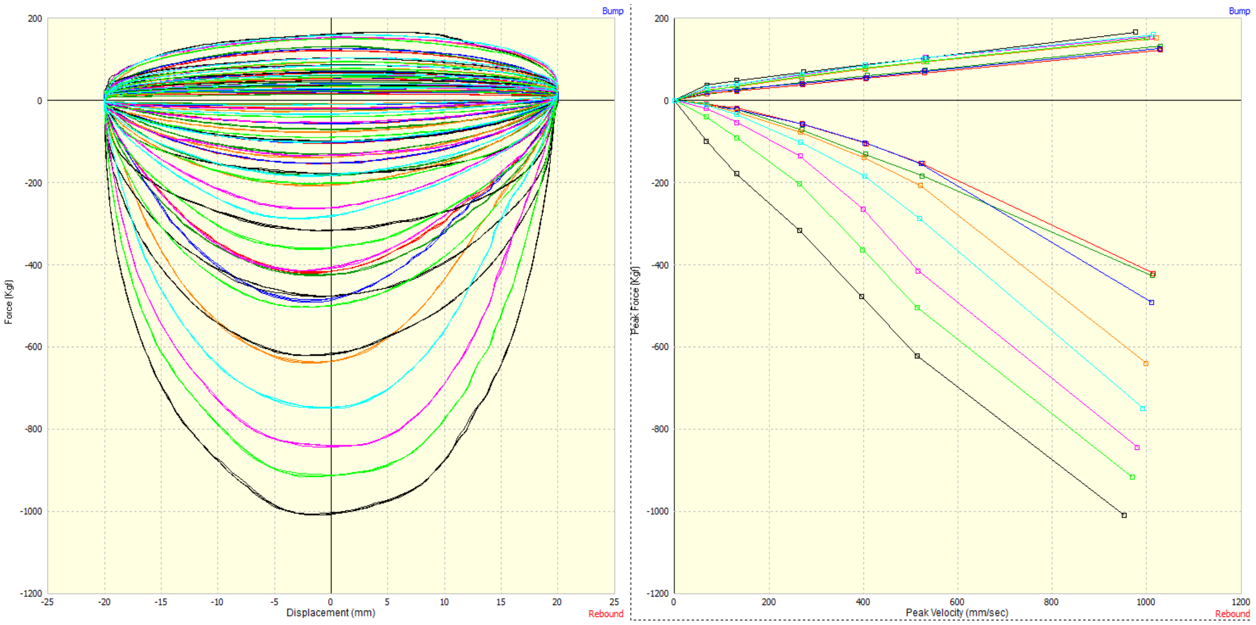


Figure 18 - Aftermarket rear damper tested at Multimatic Technical Centre, Markham

One important characteristic of a vehicle is the center of gravity height (CGH). This parameter is critical for modelling the vehicle under all dynamic events and so it was imperative that the team had an accurate value to use in vehicle dynamics models. A way to determine the center of gravity height is by using simulations and measuring the entire vehicle to determine mass properties. However, this is difficult as this requires significant computer resources to accomplish and it is not certain that all mass properties of components are correct. For these reasons, a physical test is required. To determine the center of gravity height (CGH) of the vehicle, a procedure in which the suspension is fixed, and the vehicle lifted from one side is used. This is highlighted in Figure 19 through Figure 26. By measuring the weight transfer as a function of vehicle's lifted height, an estimation of the height for the center of gravity can be made following the CGH equation. The suspension is required to be fixed in order to improve accuracy of the calculation, and so a set of "solid" springs were fabricated for the front and rear suspensions. This prevented suspension articulation entirely, allowing the team to make the best estimate for the center of gravity height.

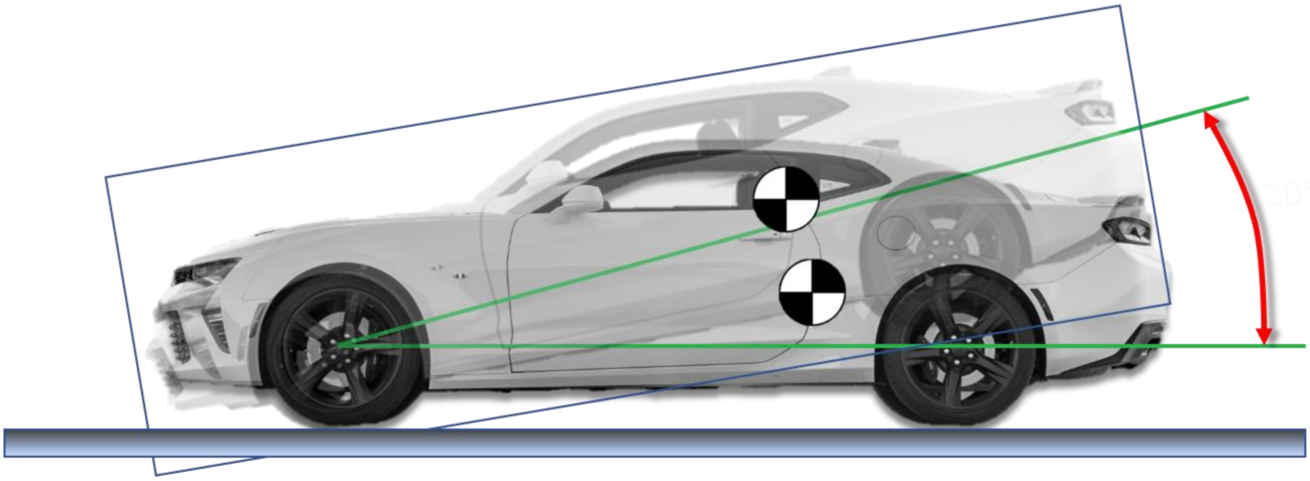


Figure 19 - To find the center of gravity height, a procedure requiring one side of the vehicle to be lifted and change in weight distribution measured at the tires was used

The equation for determining the CGH is [20]:

$$CGH = \frac{WB * FWc}{TW * Tan(\theta)} \qquad 2.96$$

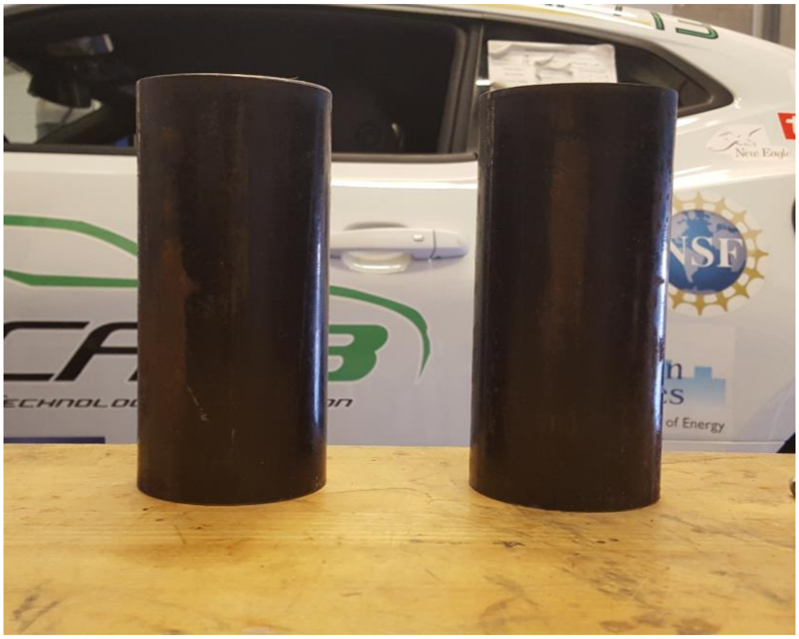


Figure 20 - Custom spring replacements for the front suspension were fabricated and used to prevent suspension travel during CGH testing

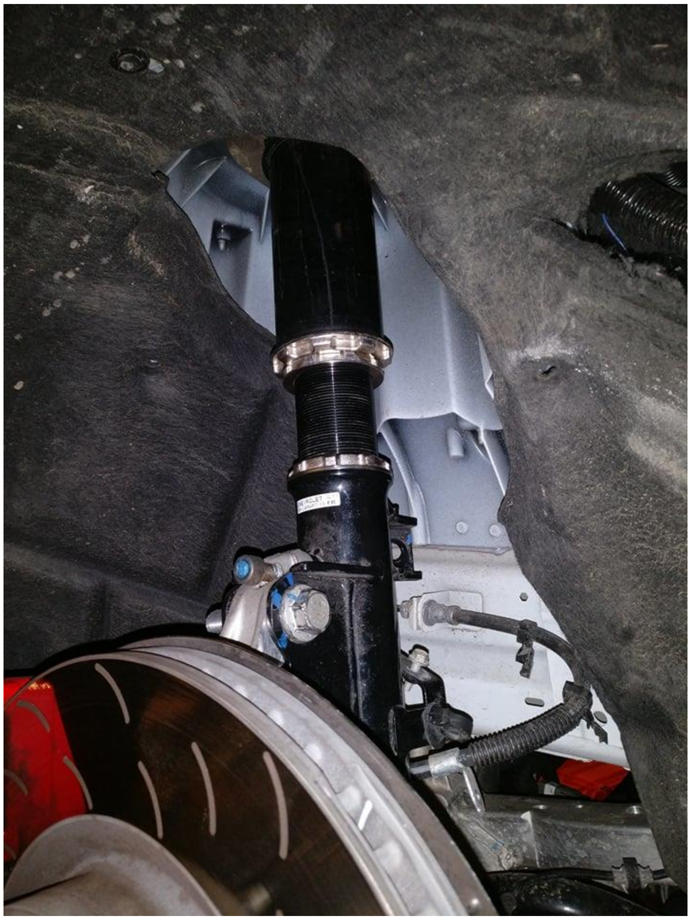


Figure 21 - Front suspension with "solid" spring to prevent suspension articulation during testing process

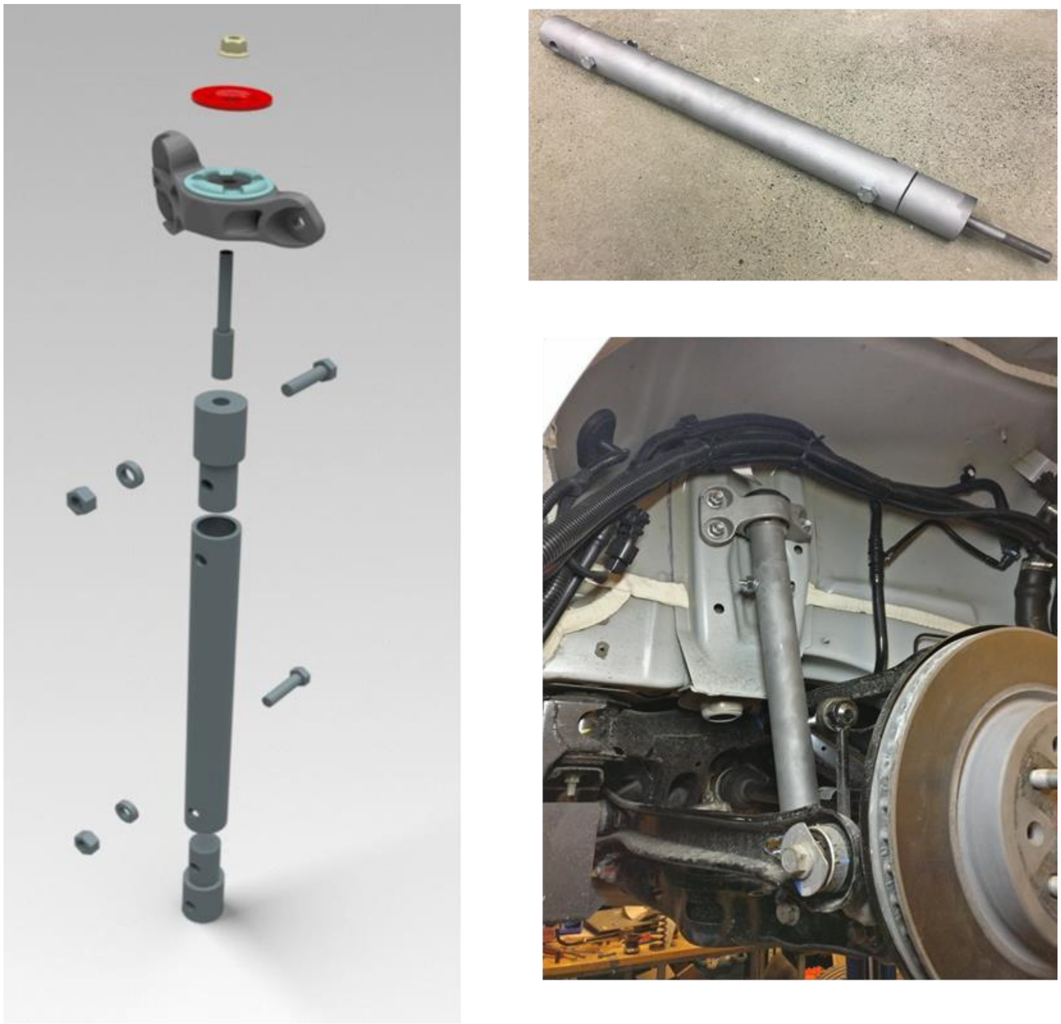


Figure 22 - Rear suspension fixture for preventing suspension travel during CGH testing

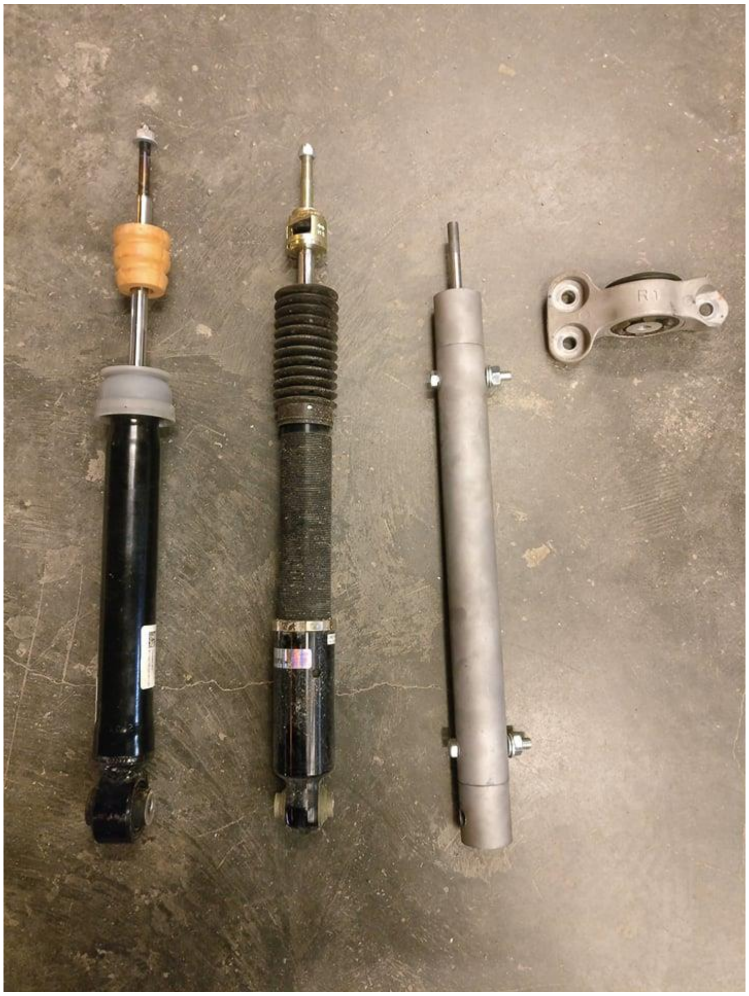


Figure 23 - Side by side comparison of the stock rear damper, aftermarket rear damper, and solid damper used for testing

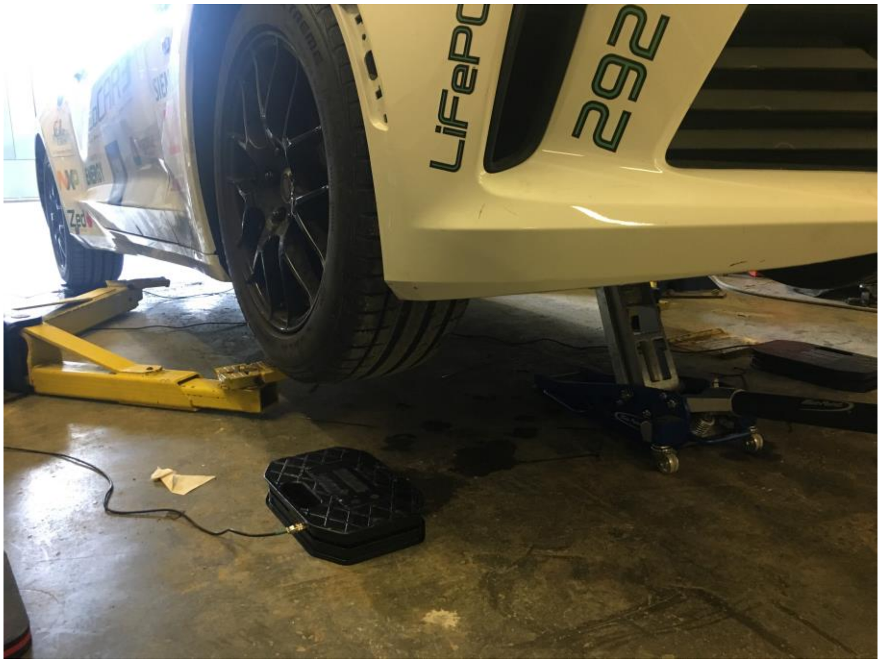


Figure 24 - Lifting the vehicle to measure the weight transfer to the un-lifted wheels to measure CGH

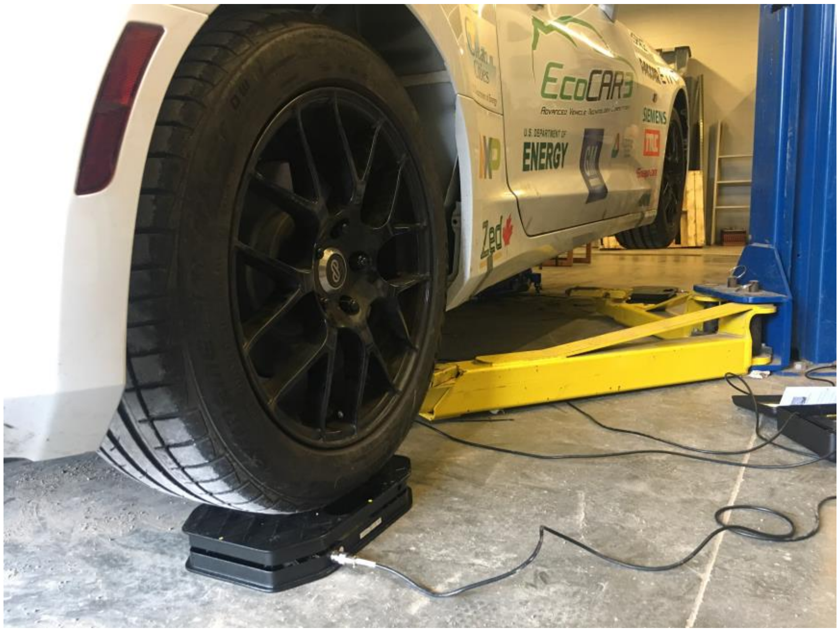


Figure 25 - Lifting the vehicle to measure the weight transfer to the un-lifted wheels to measure CGH

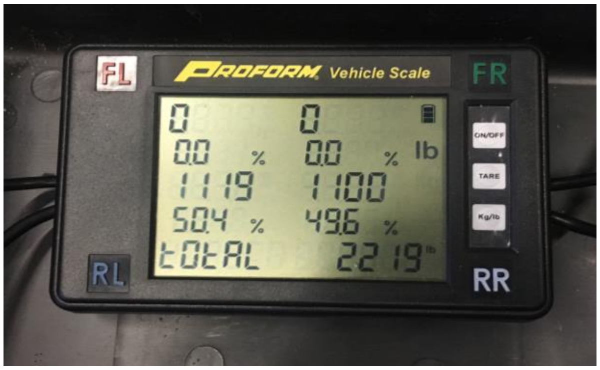


Figure 26 - Measurement of the weight seen at the un-lifted wheels during testing

A similar procedure will be required for characterizing the UWAFT Blazer suspension when the suspension for that vehicle has been chosen. This is critical to the development of an accurate vehicle model for simulation.

To understand the suspension compliance, a testing device was designed and fabricated to simulate suspension deflection under lateral loading, as shown in Figure 27 through Figure 31. A ratchet and a load cell are located between the fixtures to apply load to the suspension and measure the force of the load, respectively. All the deflection of the suspension and the corresponding load were measured. The results are plotted as shown in Figure 31. This allowed the team to understand the suspension compliance during cornering conditions and correlate the vehicle compliance model with physical test data.

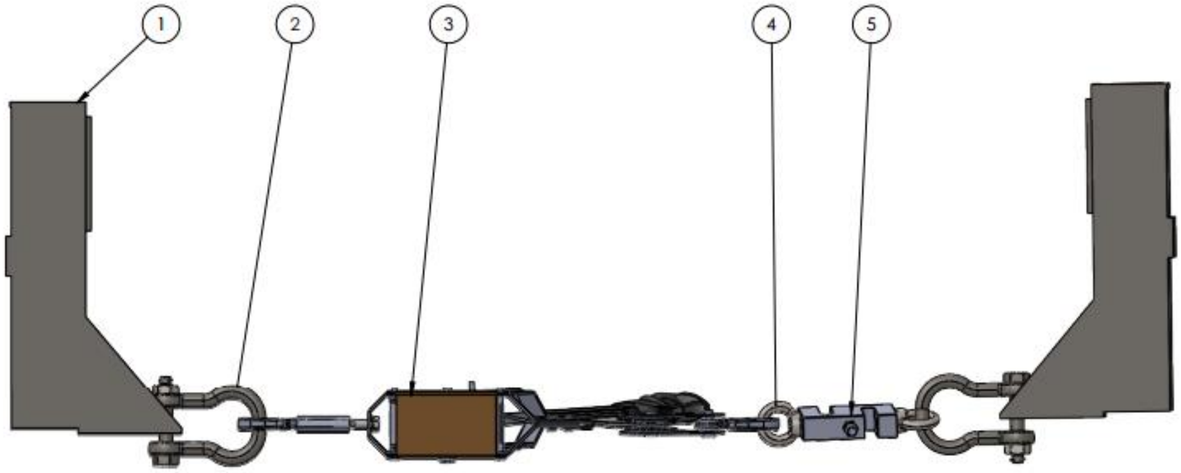


Figure 27 - Suspension compliance testing fixture designed in CAD

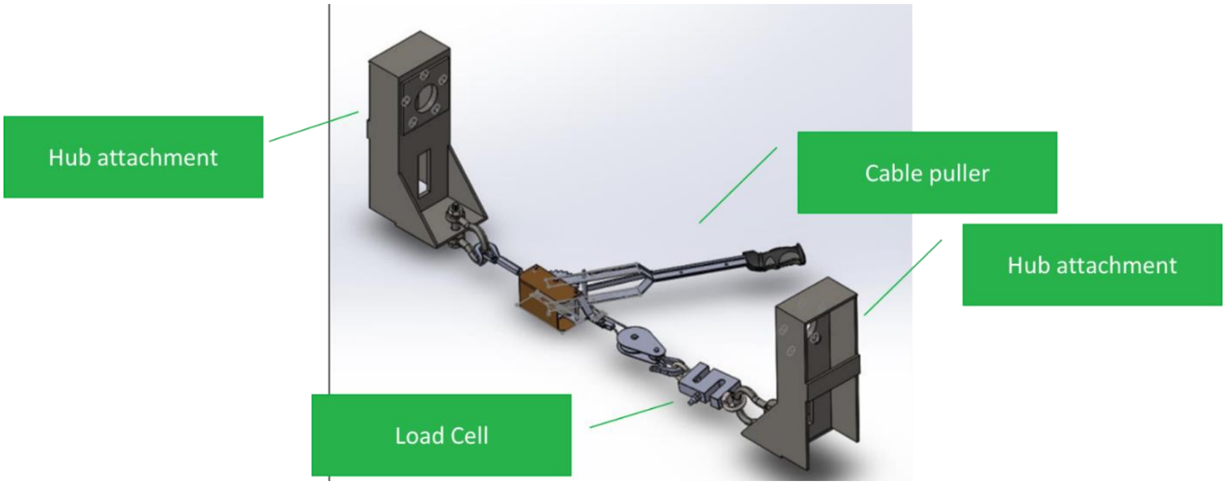


Figure 28 - Suspension compliance testing device designed in CAD

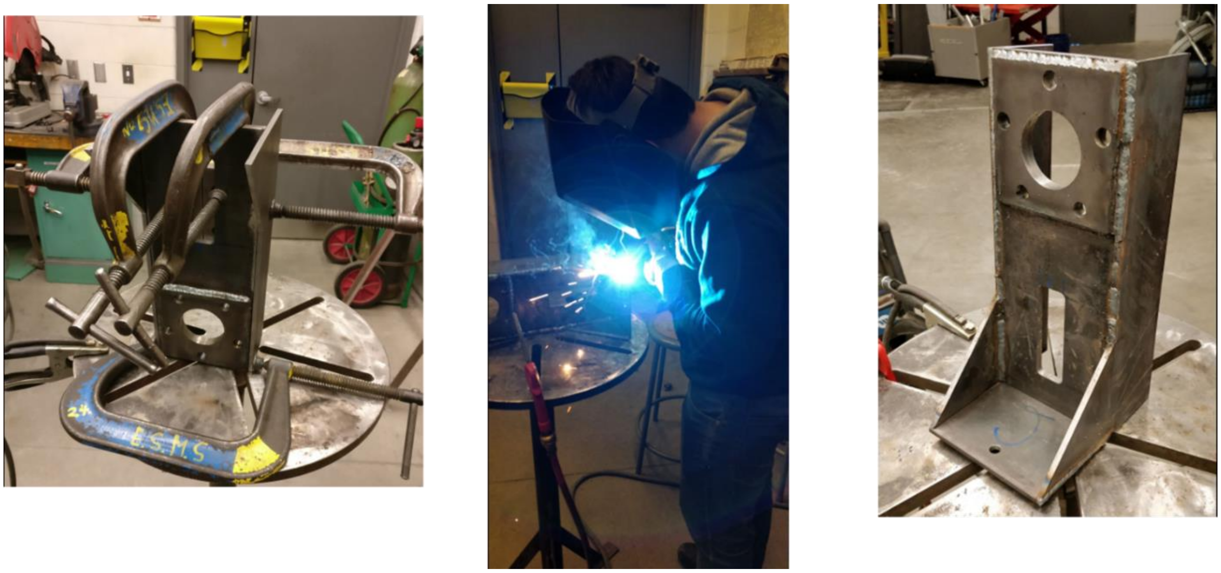


Figure 29 - Left to right: fixturing, fabricating, and final product of suspension compliance testing device

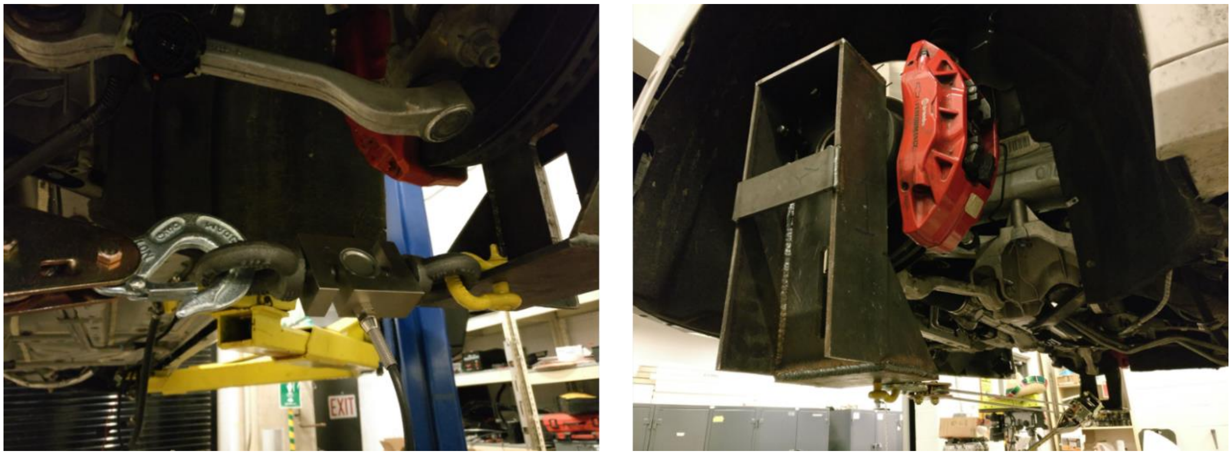


Figure 30 - Suspension compliance testing device being used on the front suspension

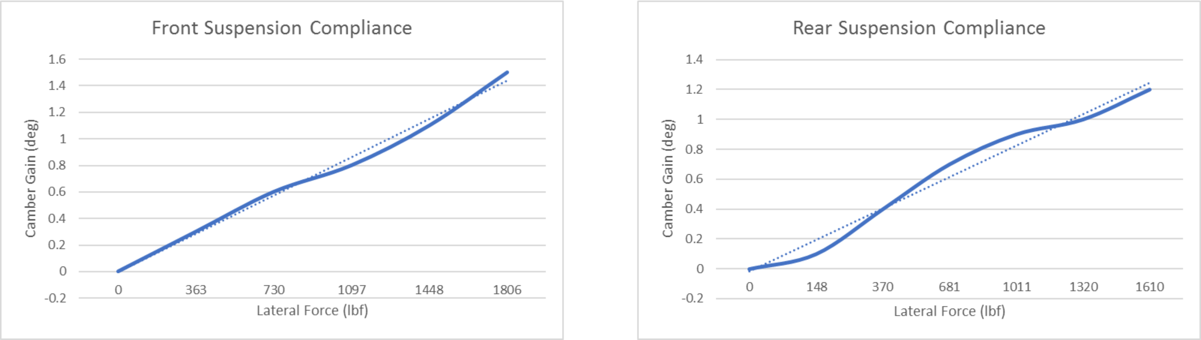


Figure 31 - Suspension compliance testing result for the front and rear suspension

The final result for the camber gain gradient was 8.2E-4 degrees per pound-force of lateral force for the front suspension, 7.45E-4 degrees per pound-force of lateral force for the rear suspension. This procedure will need to be reproduced for the UWAFT Blazer when the vehicle arrives.

1.4.4 Tire Characterization

One aspect of tire that is useful to understand is the spring and damping characteristic and behavior. We determine these characteristics by bouncing the tire and measuring the subsequent bounce heights. Then, using the following equations, the coefficient of restitution, spring rate, and damping rate of the tire can be determined [21].

$$e = \sqrt{{h_1}/{h_0}} \tag{2.97}$$

$$k = \frac{m}{(\Delta T)^2}[\pi^2 + (In\,\mathrm{e})^2] \tag{2.98}$$

$$c = \frac{-2m}{\Delta T} In\, e \tag{2.99}$$

Where e, k, and c are the coefficient of restitution, tire spring rate, and tire damping rate respectively.

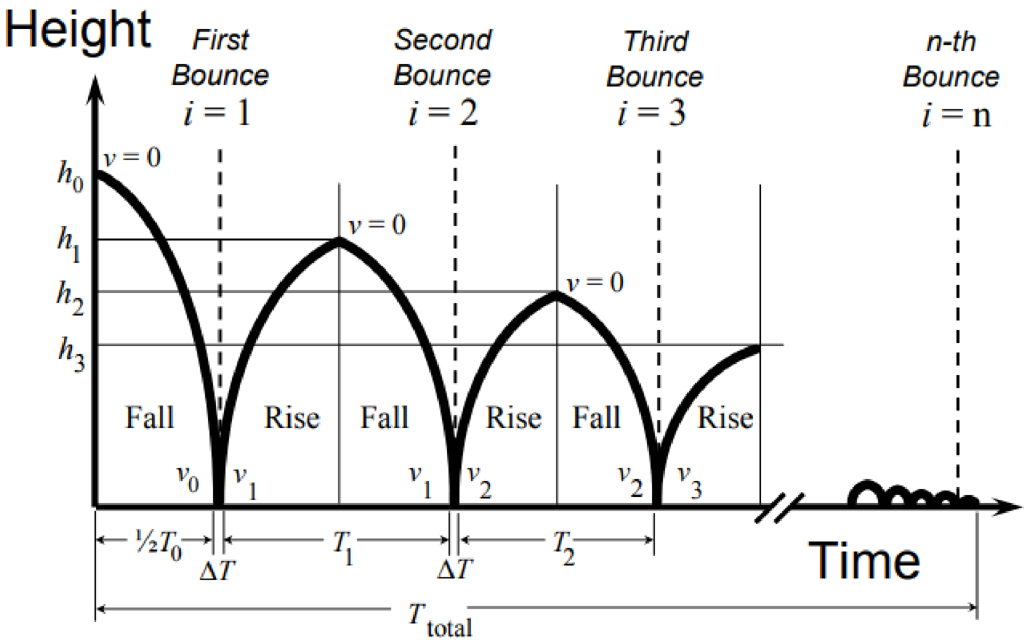


Figure 32 - Diagram explaining the process of determining the coefficient of restitution, spring rate, and damping rate of a mass-spring--damper system [22]

Results for the Camaro tire: e=0.8719, k=310 N/mm, c=0.217 Ns/mm. This procedure will need to be repeated for the Blazer tire when the Blazer arrives on campus.

Chapter 3 – Controller Design

1.5 Control Architecture

The control structure required for an automated vehicle to navigate an urban environment is extremely complex and requires the integration of multiple subsystems. A high level generalized and abstracted control architecture for an autonomous vehicle is shown in Figure 34. Each subsystem presents their own set of challenges and most pipeline data to a subsequent subsystem. Therefore, it is important that the data outputted from each subsystem is correct as to prevent erroneous actions down the pipeline. It is critical that verification and testing occurs at every step of the integration process to vet out edge cases that could break the system. Redundancies and fail-safes are required to address faults or failures in the system to ensure the safe operation of the vehicle at all times as lives are at stake.

The controllers developed do not span the entire range of controllers required to implement an automated vehicle, as shown in Figure 34, nor is this an exhaustive list of options but rather covers a selection that were perceived practical in terms of ease of implementation, tuning and suitable for implementation due to computational efficiency.

The number of systems in the software and hardware stacks place limitations and constraints on the system, notably the real-time operability of subsystems. The control architecture contains consists of many layers. The overall challenge can be broken down into manageable and modular subproblems. This section focuses on the development and evaluation of two controller types: an optimal control-based controller and a rule-based controller.

For the optimal control system, the focus will consider a high-level controller for planning a trajectory within approximately a 10-20 second time horizon and a low-level controller for regulating the vehicle states over a smaller time horizon of about 1-2 seconds. Both the high-level and low-level controllers both implement an MPC.

For the rule-based controller, the goal will be to develop a controller that can augment vehicle behavior using rules to determine the regulation targets and actuation sequence to best optimize vehicle performance; a PID approach is implemented in this controller.

The overall high-level structure of the Simulink model is shown in Figure 33. The Simulink model contains a road profile generator, a driver block (when using the rule-based controller), a vehicle dynamics controller (running either an MPC controller or the rule-based controller depending on the variant subsystem activated), custom vehicle dynamics model, and a visualization block.

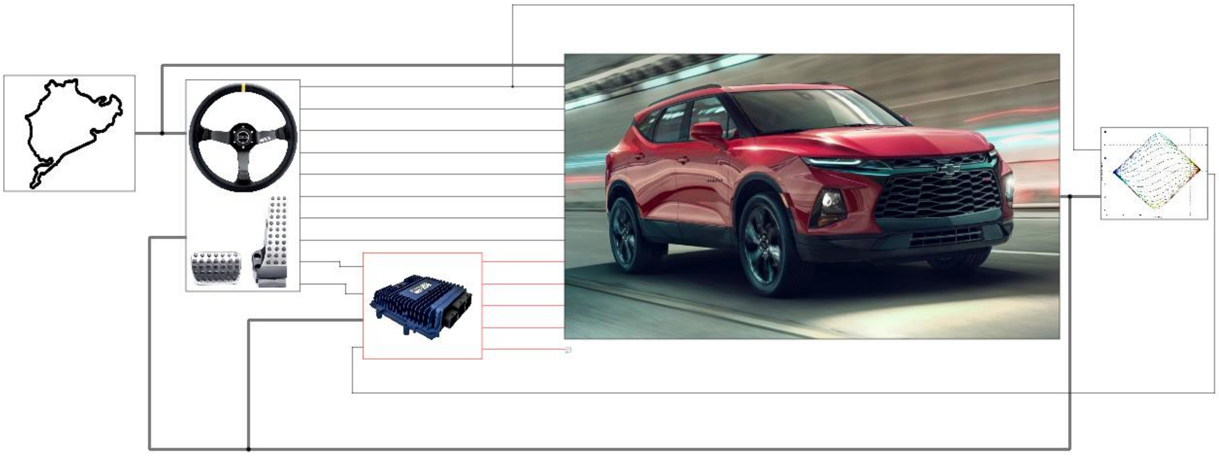


Figure 33 - Top level overview of Simulink model. From left to right: road profile, driver block, vehicle dynamics controller, vehicle model, visualization block. Blazer image [23]

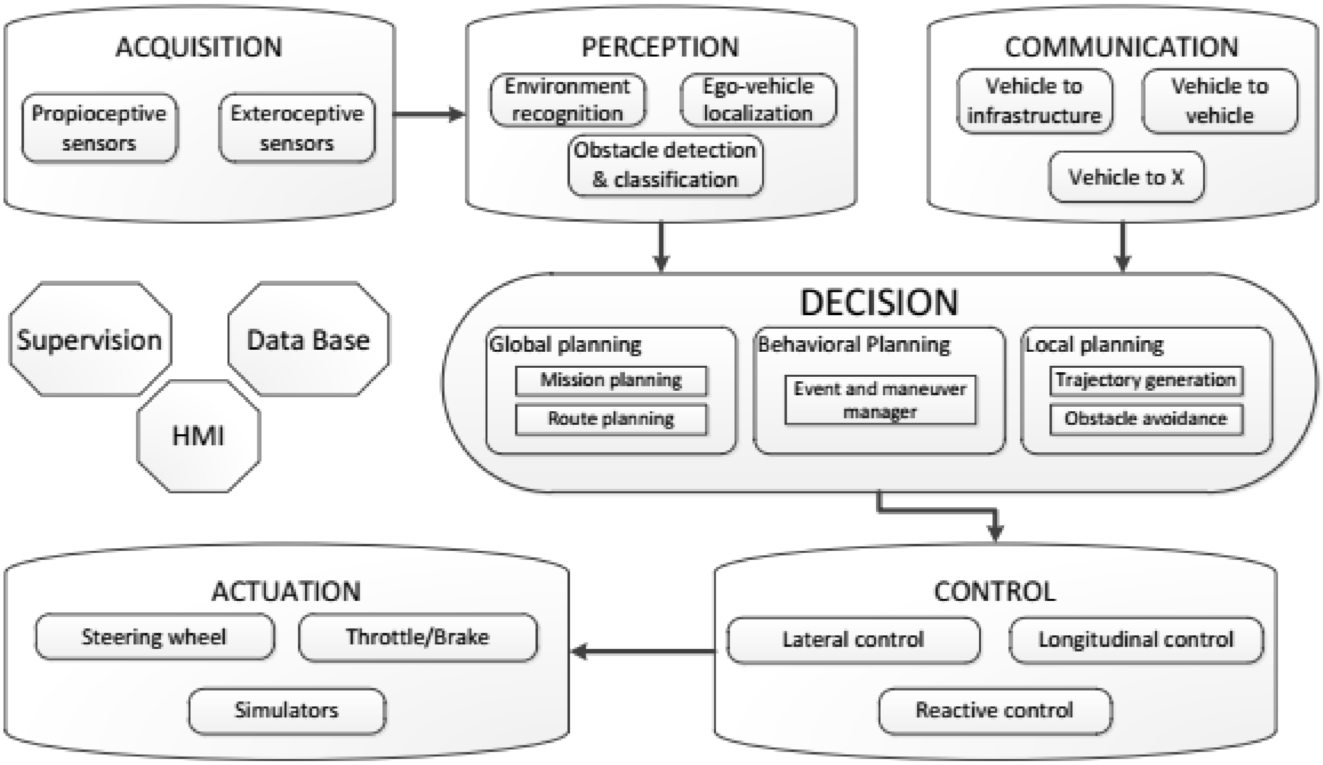


Figure 34: Control Architecture for Automated Vehicles, abstracted generally [24].

The limit inducing situations include emergency maneuvers required due to sudden obstacle avoidance need from a change in traffic conditions and sudden low road friction conditions due to precipitation during a driving maneuver. Vehicles that carry a single driver and ones that are fully loaded (incl. cargo) will have very different vehicle dynamic properties and limits in terms of handling.

The feedback that drivers get from a vehicle is generally insufficient as humans aren’t well trained to process feedback quickly and effectively. In this section, a feasible autonomous driver that can react quickly and within safe handling limits is developed. This is particularly important for vehicles using low rolling resistance tires, where the grip is generally lowered to allow for greater fuel economy as is applicable to the UWAFT Blazer.

It was assumed that the traffic scheduling and current states of vehicles in the vicinity were known through V2I, V2V and perception systems.

The concept of path planning and trajectory planning are similar in nature. Both describe the sequence of positions in space to which an object (and in this case a vehicle) takes from point A to point B, ensuring that the configuration (orientation) of the object is viable at all times while also avoiding obstacles. There are many algorithms that have been devised to solve this problem. Notable algorithms include A* (pronounced A-star), D*, Rapidly-exploring random tree (RRT), and Probabilistic Roadmap. Algorithms vary in their complexity and ultimately the computation speed required, making some solutions better to be solved offline and others more suited to be solved online. Furthermore, the time complexity also presents limitations on the applications of each algorithm as real-time Implementability is often a requirement.`

Trajectory planning takes it a step further by considering all the possible states of the object during the motion, including velocity and acceleration, in addition to position and orientation. In other words, there is the additional element of time in which there are an infinite number of trajectories per path since there are an unlimited number of varying speed patterns. The optimal speed pattern would consider many factors such as *energy efficiency*, *speed*, and *safety*.

1.6 Requirements

As part of the controller design, it is important to set out requirements for the controllers to determine success. There are two levels of control considered here: a high-level trajectory planning controller and a low-level regulation controller. The general requirements for these planners and controllers are set out in the Table 2 below:

Table 2: Approximate Requirements Table

Metric	High-Level Trajectory Controller	Low-Level Trajectory Controller
Speed	Near Real-Time (NRT)	Real-Time (RT)
Complexity	Low	High
Fidelity and Accuracy	Low	High

1.7 Model Predictive Control

A Model Predictive Control (MPC) based controller can perform system optimization based on a defined cost function while honoring constraints of the system. The cost function as well as the constraints can be non-linear in nature and it can also be time-varying as the controller performs a new prediction every cycle and can be updated between cycles; the time between cycles is the sample time of the controller [25] [26]. The way it works is the MPC uses a discretized control-oriented model of the plant dynamics and predict the system dynamics based on a time-ordered set of control inputs over a prediction horizon. After each time step, the controller makes a new prediction over the prediction horizon. The prediction horizon itself can be any number of multiples of the sample time. A common prediction horizon is 10x the sample time of the controller, but can be extended to several seconds or even minutes or hours if desired depending on the application. The drawback of an extended prediction horizon into the future is the time required to compute such long horizons and the accuracy of such predictions. A lab setting may permit the prediction horizon to span an arbitrarily long duration as long as a known accurate model is used, however real-time applications limit the prediction horizon to orders of seconds and minutes at most [27].

1.8 High Level Trajectory Planning Controller

The high-level trajectory planning controller takes in the initial state of the vehicle (position, velocity, acceleration of the chassis and wheels), the final desired state of the vehicle, and determines the optimal path such that the total amount of energy consumed, time taken, and error from the path is minimized while staying within the limits of control for the vehicle.

A nonlinear MPC is an ideal tool for solving trajectory planning problems since it is able to determine the optimal solution to an open-loop, constrained optimization problem. A low-order prediction model is used due to its simplicity and speed of calculations. Due to the simplicity of the model itself, a prediction horizon can span several seconds into the future while still meeting NRT requirements.

The fact that the vehicle is operating in a connected environment means that the surroundings, even well beyond the vision of the vehicle can be known. This includes the timing and scheduling of the infrastructure. With this information, the path planner can determine an optimal path and velocity profile along the path with terminal constraints considered (stop lights, stop signs, etc) while minimizing the user-defined cost function.

The high-level controller is based on a simple point-mass model as described in Section Point Mass Model (2 DOF)1.3.1. The model represents the vehicles center of gravity and its position is described with respect to a global inertial reference frame. The velocity and acceleration targets are determined for each position of the vehicle by the high-level controller. The controller considers spatial constraints such as the road boundaries, obstacles in the path, terminal constraints such as stopped positions and speeds at intersections, as well as safety constraints such as a maximum acceleration vector allowed for the vehicle at all times.

1.8.1 Model Definition

Model predictive control requires the definition of a vehicle model for performing system prediction in response to control input. The model used in the high-level trajectory controller was a point mass model. The simplicity of the model allows the controller to project and predict several seconds into the future to generate an optimal path and speed profile along the path in NRT. More information about the point mass model can be found in Section 1.3.1.

1.8.2 Cost Function

For computational efficiency and to increase the likelihood of convergence, the cost function is quadratic/convex in nature and it seeks to minimize user-defined costs. For this high-level controller however, since it is being used to generate an optimal trajectory, there is no feedback element; the high-level controller seeks to generate an optimal reference for the lower-level controller to follow only.

The objectives of the cost function are to maintain set speed (as defined by the driver or speed limits) and generate a path between point A and point B efficiently all the while staying within the boundaries of the road and avoiding obstacles. By placing a cost on powertrain usage (manipulated variables), a spatially efficient solution is naturally generated without being defined explicitly.

The cost function is defined as follows [28]:

$$J(z_k) = J_u(z_k) \qquad 3.1$$

Where J_y, $J_u(z_k)$, $J_{\Delta u}(z_k)$, $J_\epsilon(z_k)$ represent the reference tracking cost component, manipulated variable cost component, manipulated variable move cost component, and constraint violation cost component respectively. For this controller, the cost function only considers the manipulated variable (powertrain) usage as terms in the cost function as the purpose of this controller is to generate an optimal path that minimizes energy usage; the low-level controller in the next section will touch on the feedback control to track and regulate the reference trajectory outputted from this controller.

$$J_u(z_k) = \sum_{j=1}^{n_u} \sum_{i=0}^{p-1} \left\{ \frac{w_{i,j}^u}{s_j^u} \left[u_j(k+i|k) - u_{j,target}(k+i|k) \right] \right\}^2 \qquad 3.2$$

Where:

k = current control interval

P = prediction horizon (number of intervals)

n_u = Number of manipulated variables

z_k = QP decision

$u_{j,target}$ = Target value for jth input at the ith prediction horizon step, in engineering units.

1.8.3 System Constraints

The constraints for this controller are all hard constraints in which the controller cannot at any point propose a solution in which the constraints are violated. The constraints for this controller are the boundaries of the road, the boundary surrounding impassable obstacles, as well as acceleration limits so that the vehicle never exceeds safe limits of the vehicle. An example of a road boundary used is defined in Figure 35, in which a double lane change maneuver is illustrated [29].

$$P_l(x) := \begin{cases} 0 & \text{if} \quad x \leq 44, \\ 4\,h_2\,(x-44)^3 & \text{if} \quad 44 < x \leq 44.5, \\ 4\,h_2\,(x-45)^3 + h_2 & \text{if} \quad 44.5 < x \leq 45, \\ h_2 & \text{if} \quad 45 < x \leq 70, \\ 4\,h_2\,(70-x)^3 + h_2 & \text{if} \quad 70 < x \leq 70.5, \\ 4\,h_2\,(71-x)^3 & \text{if} \quad 70.5 < x \leq 71, \\ 0 & \text{if} \quad 71 < x. \end{cases}$$

$$P_u(x) := \begin{cases} h_1 & \text{if} \quad x \leq 15, \\ 4\,(h_3-h_1)\,(x-15)^3 + h_1 & \text{if} \quad 15 < x \leq 15.5, \\ 4\,(h_3-h_1)\,(x-16)^3 + h_3 & \text{if} \quad 15.5 < x \leq 16, \\ h_3 & \text{if} \quad 16 < x \leq 94, \\ 4\,(h_3-h_4)\,(94-x)^3 + h_3 & \text{if} \quad 94 < x \leq 94.5, \\ 4\,(h_3-h_4)\,(95-x)^3 + h_4 & \text{if} \quad 94.5 < x \leq 95, \\ h_4 & \text{if} \quad 95 < x. \end{cases}$$

where $B = 1.5\ \mathrm{m}$ is the car's width and

$$h_1 := 1.1\,B + 0.25, \quad h_2 := 3.5, \quad h_3 := 1.2\,B + 3.75, \quad h_4 := 1.3\,B + 0.25.$$

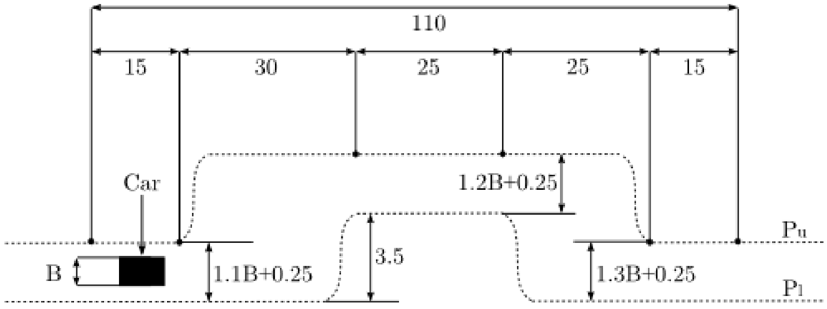


Figure 35 - Double Lane Change Test Course [29]

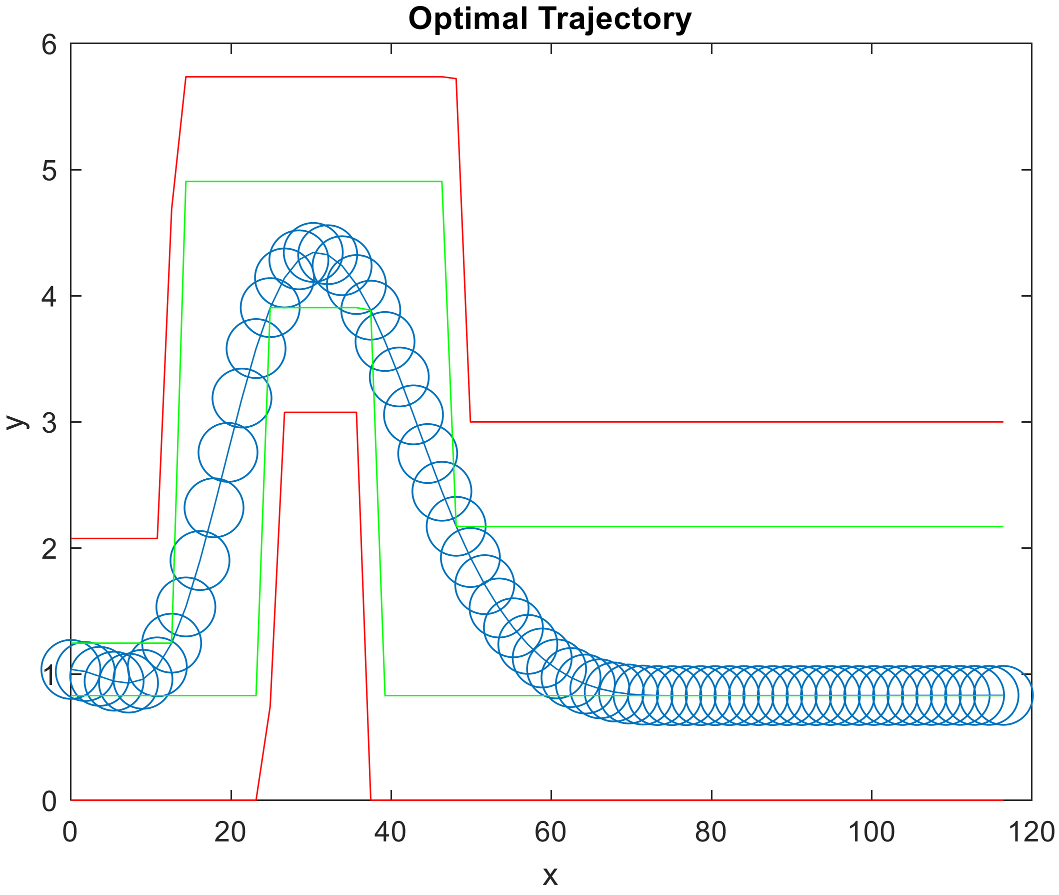


Figure 36 - Optimal path generated by the nonlinear MPC controller. Red lines indicate road boundaries; green line represents the safe boundary for the vehicle give its track width; blue line represents the vehicles path during the maneuver

It is interesting to note that the controller determined that the best path was to dip rightward during the initial straight section before the initial turn-in of the course. This has been observed in literature as well, as [30] has shown. The path generated is acceptable and within the boundaries set out in the constraints. It is logical and stays within acceptable acceleration limits as shown in Figure 39. The acceleration is slightly above conservative limits so that the low-level controller can operate and regulate the vehicle. Conservative limits in this context means any acceleration above 0.5 G's of acceleration – this is generally the point in which tires start to become significantly nonlinear in behavior. The smoothness of the acceleration plots indicate that the high-level controller plans to execute a maneuver that is not particularly jerky and should be comfortable for the occupants in general. Additionally, the overall speed is maintained closely to the set speed as shown in Figure 38, indicating that the cost function is effective as well.

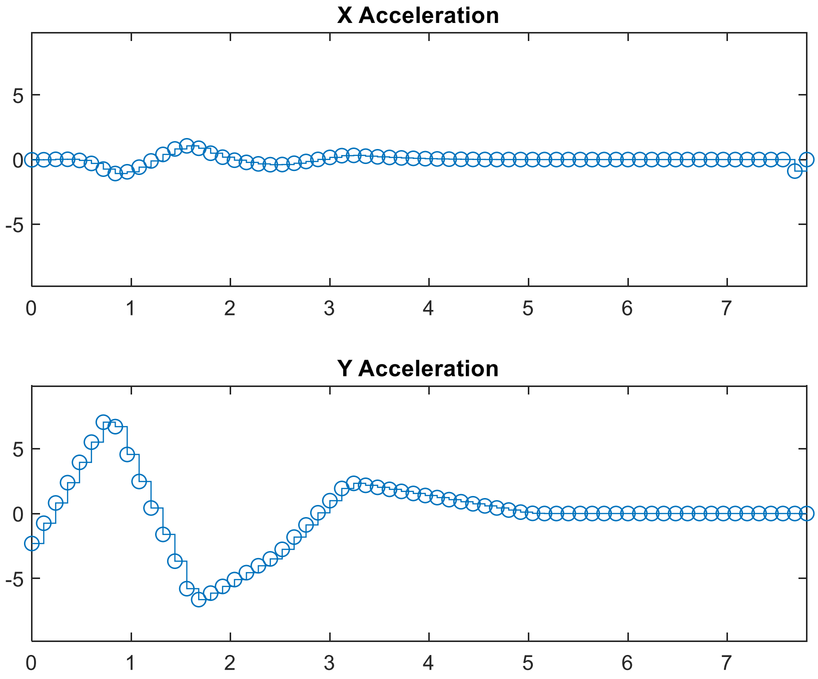


Figure 37 - Acceleration of the vehicle in the global reference frame

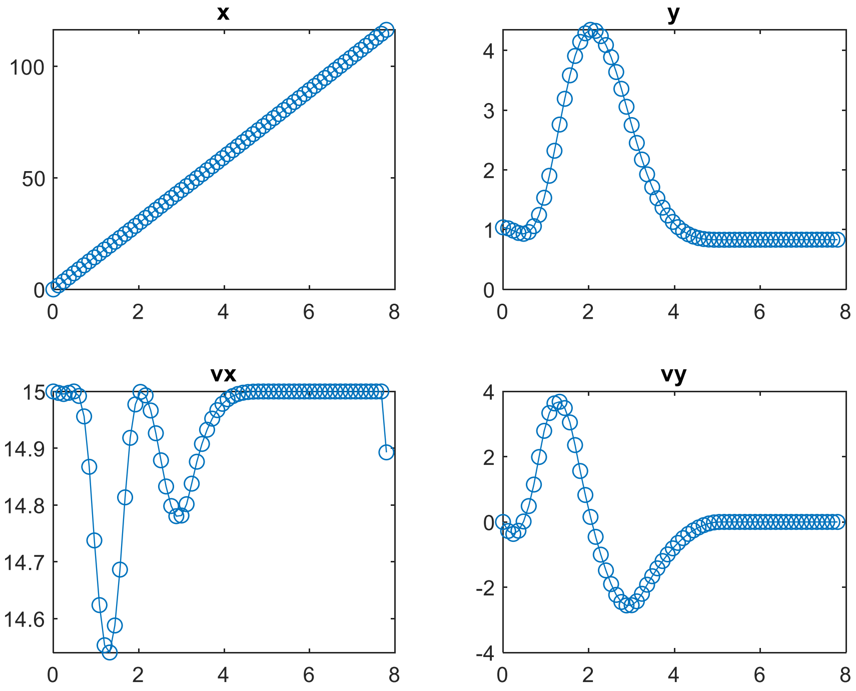


Figure 38 - Various plots showing the global position, as well as longitudinal velocity and lateral velocity over time

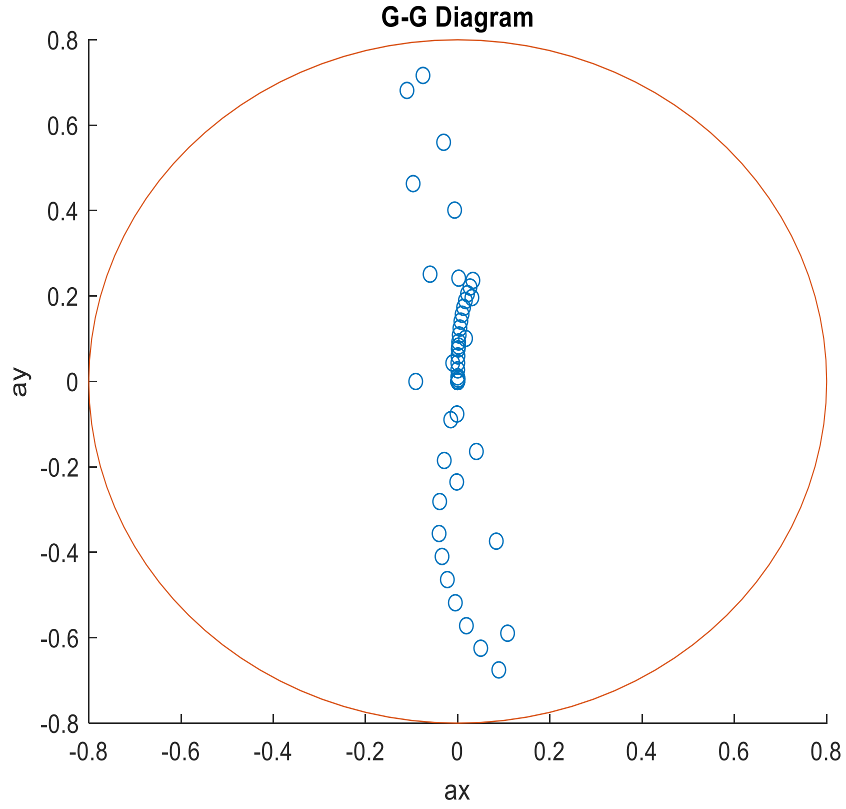


Figure 39 - G-G Diagram showing the planned combined lateral and longitudinal acceleration of the vehicle for the double lane change. The blue dots represent the instantaneous acceleration over time.

1.9 Low Level Regulation and Tracking Controller

In a perfect world, the vehicle will be able to achieve the desired trajectory passed down from the high-level controller. However, no model is 100% accurate and thus a regulation controller is needed to minimize tracking error. This controller is used at a local level and operates on a shorter time horizon but utilizes a more sophisticated model for predictive control. Regulation was achieved using a nonlinear MPC approach, where lateral velocity, longitudinal velocity, yaw rate, lateral deviation, heading error, road curvature, and noise are the inputs; steering angle and wheel torque are the outputs of the controller. This is represented as a non-linear state space formulation shown in Section 1.9.1. A figure depicting the set up can see seen in Figure 40.

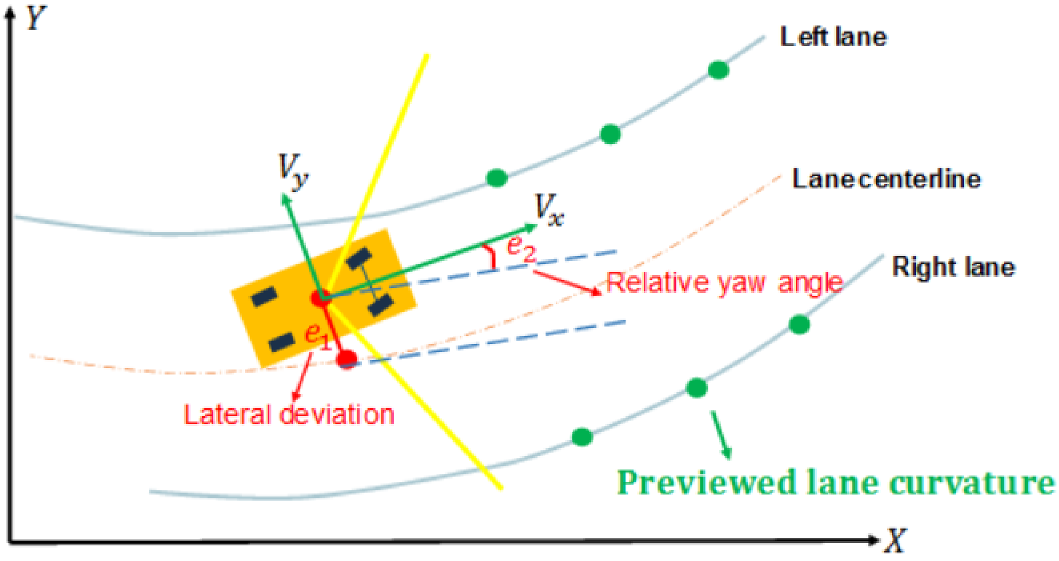


Figure 40 - Lane following system that keeps the vehicle travelling along a target reference path [31]

The road information is passed down from the high-level controller in the form of waypoints for the vehicle to pass through. One method of regulation is to track these waypoints directly and minimize the error of the vehicle. This method can be computationally expensive as many points are evaluated. An alternative method is to convert these waypoints and generate a curvature vector to track. This method allows road information to be condensed and is less computationally expensive to optimize for. The road information is passed to the controller in the form of a vector containing information about the road curvature over a prediction horizon that the controller can relate back to the two objectives for minimizing tracking error for: lateral deviation and heading error.

Each position in along the trajectory has an associated curvature, but an issue arises when the vehicle strays from the planned trajectory as there is no associated curvature with positions off the planned course. For small deviations from the planned course, the controller determines the closest waypoint along the planned path and passes a sequence of curvatures to the controller the size of the prediction horizon, with the first curvature value representing the curvature from the nearest waypoint to the vehicle. This curvature sequence, as well as the state and control predictions, are updated at each time step of the controller.

1.9.1 Model Definition

Model predictive control requires the definition of a vehicle model for performing system prediction in response to control input. The nonlinear state space model formulation used in the low-level trajectory controller is defined as follows:

$$\frac{d}{dt}\begin{bmatrix} V_y \\ \dot{\phi} \\ V_x \\ e_1 \\ e_2 \\ x_{od} \end{bmatrix} = \begin{bmatrix} -\frac{2C_f + 2C_r}{mV_x} & -V_x - \frac{2C_f l_f - C_r l_r}{mV_x} & 0 & 0 & 0 & 0 \\ -\frac{2C_f l_f - 2C_r l_r}{I_z V_x} & -\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z V_x} & 0 & 0 & 0 & 0 \\ \dot{\phi} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & e_2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_y \\ \dot{\phi} \\ V_x \\ e_1 \\ e_2 \\ x_{od} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{2C_f}{m} \\ -\frac{W_r}{2I_z} & \frac{W_r}{2I_z} & \frac{2C_f l_f}{I_z} \\ \frac{1}{m} & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_{x_1} \\ F_{x_2} \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ x_{od} \end{bmatrix} [\rho] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [w_{od}] \qquad 3.3$$

$$Y = [V_x \quad e1 \quad e2 + x_{od}] \qquad 3.4$$

Where:

V_y = Lateral velocity in the body reference frame

$\dot{\phi}$ = Yaw rate of the vehicle

V_x = Longitudinal velocity in the body reference frame

e_1 = Lateral deviation from reference path

e_2 = Deviation in heading angle from the road tangent

x_{od} = Noise injected into the system

1.9.2 Cost Function

The cost function for this controller contains 8 parts that are weighted and scaled relative to each other in the order of priority. These 8 parts include components for error tracking, vehicle stability, and energy minimization. Terms related to error tracking are: i) set speed error; ii) lateral deviation error; iii) heading error. Minimizing these components mean that the vehicle will minimize deviation from the target trajectory. Terms related to maintaining order stability include: i) steering angle penalty; ii) steering rate penalty. Minimizing these terms will encourage the controller to minimize extreme control inputs. Terms related to minimizing energy consumption are: i) motor usage; ii) torque vectoring usage. The excessive use of the motors will adversely affect energy consumption, as will the unnecessary usage of torque vectoring (where opposing wheels work in opposite directions to provide additional yaw rate without providing propulsive torque).

The cost function takes the form as follows:

$$J(z_k) = J_y(z_k) + J_u(z_k) \tag{3.5}$$

$$J_y(z_k) = \sum_{j=1}^{n_y} \sum_{i=0}^{p} \left\{ \frac{w_{i,j}^y}{s_j^y} \left[r_j(k+i|k) - y_j(k+i|k) \right] \right\}^2 \tag{3.6}$$

$$J_u(z_k) = \sum_{j=1}^{n_u} \sum_{i=0}^{p-1} \left\{ \frac{w_{i,j}^u}{s_j^u} \left[u_j(k+i|k) - u_{j,target}(k+i|k) \right] \right\}^2 \tag{3.7}$$

Where:

k = current control interval

P = prediction horizon (number of intervals)

n_u = Number of manipulated variables

z_k = QP decision

y_j = Predicted value of j^th^ plant output at i^th^ prediction horizon step, in engineering units

r_j = Reference value for j^th^ plant output at i^th^ prediction horizon step, in engineering units

s_j^u = Scaling factor for the j^th^ input, in engineering units

$w_{i,j}^y$ = Tuning weight for j^th^ plant output at i^th^ prediction horizon step (dimensionless)

$u_{j,target}$ = Target value for j^th^ input at the i^th^ prediction horizon step, in engineering units.

1.9.3 System Constraints

Constraints for the system are generally based around safety. Constraints are set such that the controller will not propose solutions that would exceed the limits of any of the constraints. The constraints used placed limitations on the maximum values on certain vehicle states and inputs.

The tire force vector at any of the tires must not exceed a specified amount equivalent to the user-defined maximum tire force determined from the vehicle mass and user-defined maximum vehicle acceleration:

$$\sqrt{F_x^2 + F_y^2} < F_{max} \qquad 3.8$$

This can be further illustrated by viewing Figure 41, noticing that the force vector is a combination of longitudinal force and lateral force. Friction circle theory states that the friction provided of the tire cannot exceed the boundaries of the friction circle, else controllability of the vehicle risks being lost. In reality, the circle often takes the form of an ellipse, and varies according to the normal load of the vehicle on the tire as well as external factors such as dirt, precipitation, and tire wear. The friction circle is simply a tool used to illustrate the concept of force vectors on a tire and is a conservative estimate on the level of grip a tire can produce.

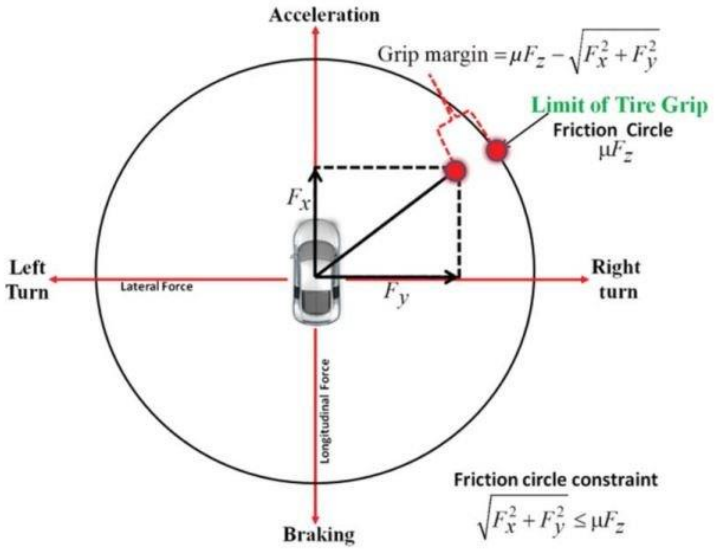


Figure 41 - Friction circle of a tire [32]

A tire force rate constraint places a limit on how quickly the controller can command a change in tire force as to prevent chattering and improve ride comfort. An approximated value for the inequality constraint is shown in the following:

$$\frac{d}{dt}(F_x) \; < 500\frac{N}{s} \qquad 3.9$$

The steering angle is also limited to prevent the controller from generating solutions that would cause the steering angle to extend beyond the physical limitations of the vehicles steering system. An approximated value for the inequality constraint is shown in the following:

$$abs(\delta) < 1.13\,rad \qquad 3.10$$

The steering angle rate placed a limit on how quickly the controller could command a change in steering wheel angle as to prevent chattering and improve ride comfort, similar to the tire force rate limit constraint defined above. An approximated value is shown:

$$\frac{d}{dt}(\delta) < 2\frac{rad}{s} \qquad 3.11$$

1.9.4 Rule-Based Controller

The rule-based controller is a control allocator controller that determines the appropriate control input of the vehicle powertrain in order to maintain stability of the vehicle during adverse driving maneuvers or environmental conditions. It seeks to optimize the control of the vehicle using rules as opposed to a formal optimal-control based optimal controller such as the MPC controller discussed in the previous section. The rule-based controller contains several individual components and is based around the yaw rate control. The theory is that a given a set of inputs from the driver, the vehicle should exert a certain yaw rate in order to be stable, while at the same time meet acceleration expectations of the driver. In this controller, the driver can be either a physical driver or a simulated driver; in contrast to the optimal controller, the driver is decoupled from the vehicle dynamics controller.

1.9.4.1 Yaw Rate Generator

A suitable yaw rate is necessary to serve as a reference for the controller to follow. This is similar to the approach in [33]. To find a suitable yaw rate, a linear bicycle model was used to generate the optimal yaw rate response. It is used as a benchmark because a stable linear bicycle model is inherently relatable as drivers expect vehicles behave as such; humans do not do well when systems behave nonlinearly so the reference for controller should be such that the target reference is linear in nature. By using linear bicycle model with neutral steering characteristics, the driver can provide input to the vehicle and expect the vehicle dynamics controller to ascertain the driver intention and control the vehicle accordingly to match.

Using the bicycle model defined in Section 1.3.2, the model was run swept through a series of driving maneuvers and the steady state response was measured. The vehicle variables were vehicle velocity and steering wheel angle. From there, the steady state yaw rate was measured and recorded. The result is a yaw rate gain map that can be used as a yaw rate reference for the controller. This is shown in Figure 42.

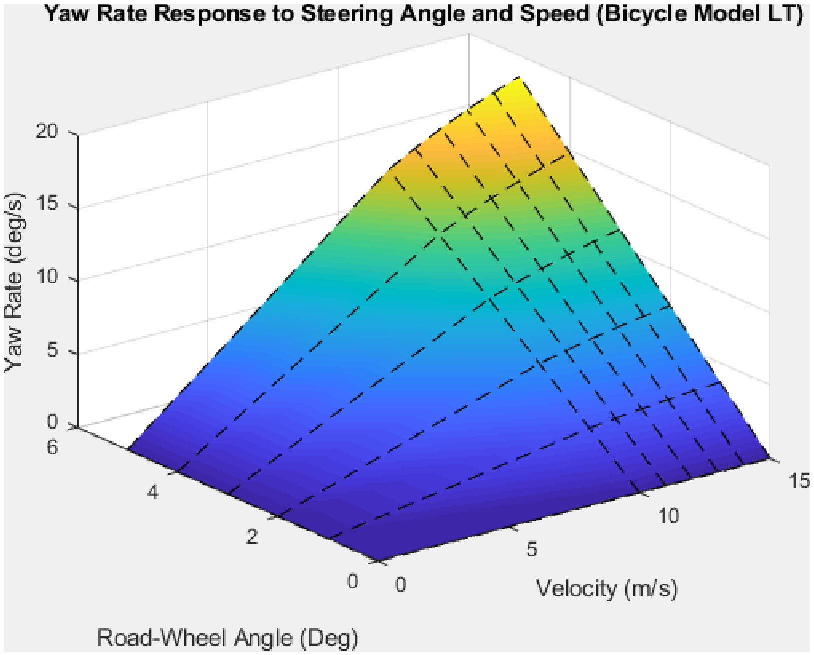


Figure 42 - Yaw Rate Gain map for a neutral steering linear bicycle model

Additional plots generated by subjecting an understeering vehicle and a nonlinear bicycle model were made as comparisons, as shown in Figure 43 and Figure 44.

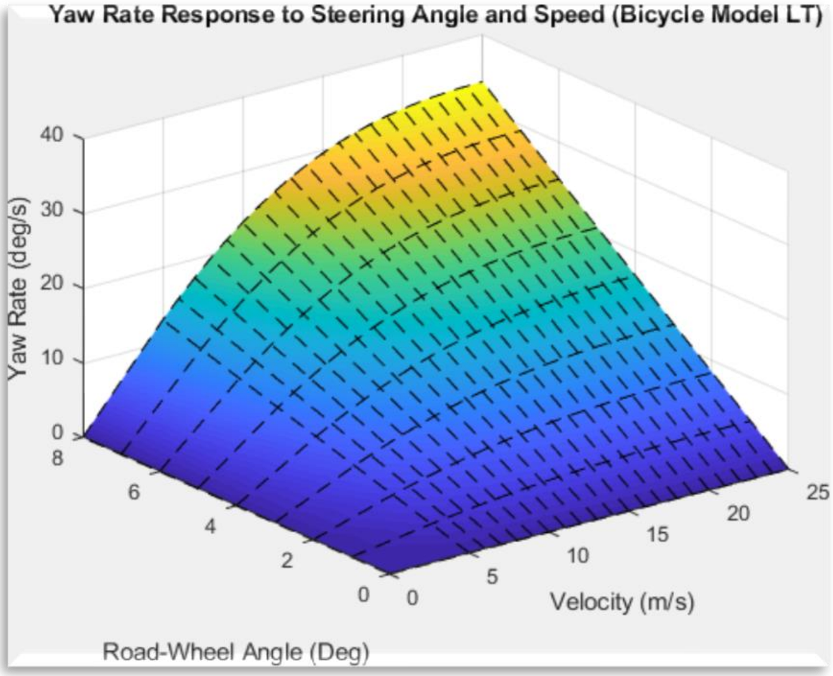


Figure 43 - Yaw Rate Gain map for an understeering linear bicycle model

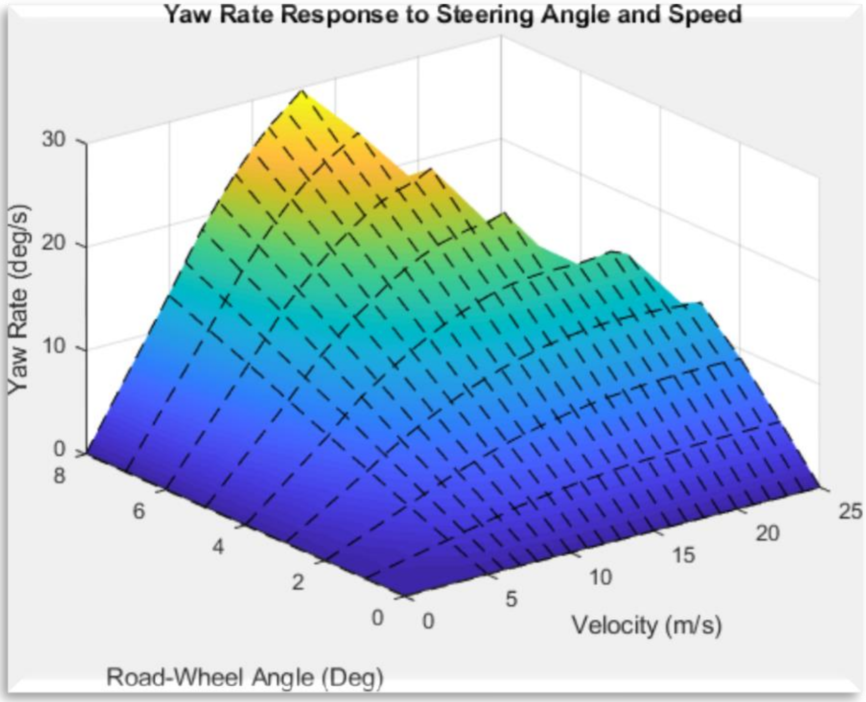


Figure 44 - Yaw Rate Gain map for a nonlinear bicycle model

1.9.4.2 PID Control

A proportional-integral-derivative (PID) controller is used for determining the correct tractional torque and yaw moment demands on the vehicle in order to achieve acceleration and yaw rate targets. Initial values for the PID controller were simply estimated and tuned by trial and error. This controller determines the correct combination of forward torque and rotating torque at a high level. This information is passed to the control allocator in the subsequent layer for distributing the torque amongst the wheels.

1.9.4.3 Control Allocator

The control allocator takes the high-level information from the PID controller and determines which wheels are used for providing the vehicle forward and rotational torque. The control allocator follows a certain torque schedule for the wheels. This torque schedule was devised through intuition and experience of vehicle dynamics. A diagram illustrating the torque schedule strategy is shown in Figure 45. The torque schedule utilizes the rear tires for generating yaw moment initially until the one of the wheels reaches zero torque. After which point, the front tires contribute a small amount of counter torque to generate additional yaw moment. If that is still not enough, then the rear tires contribute additional counter torque for generating additional yaw moment. Beyond this point, if further rotational yaw moment is required, then the front tires generate additional counter torque, placing all tires at complete saturation. This method is similar to the one used in [34]. The results of the controller in its entirety is demonstrated in Chapter 4 – Vehicle Performance.

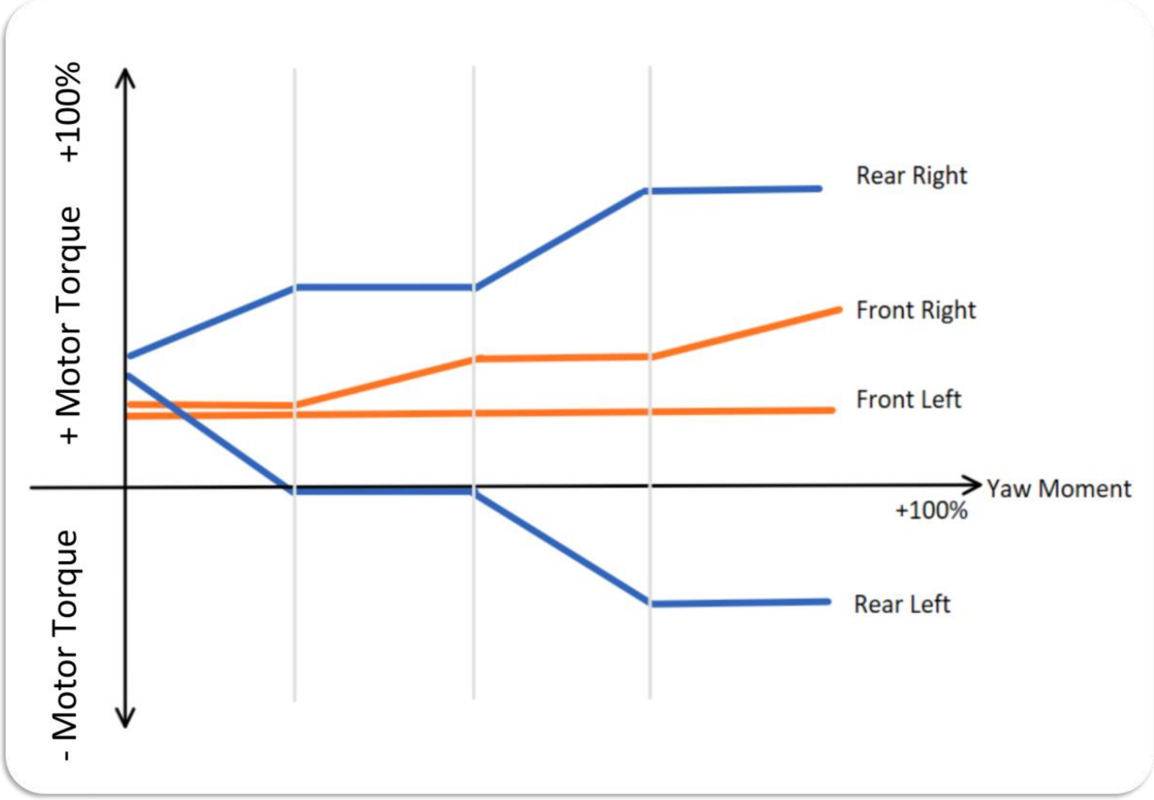


Figure 45 - Torque Scheduling strategy for an all-wheel drive vehicle with independent wheel actuation

1.10State Estimation

Many vehicle states can be measured directly with sensors on a vehicle. For example, wheel speed steering wheel angle and rate, yaw rate, lateral and longitudinal acceleration, and vehicle position can be measured directly using sensors. However, certain useful states cannot be measured directly, such as lateral velocity or sideslip angle. These are very important for estimating the state of the tires and ultimately how close to or beyond the saturation limit the tires are. Since tires are the only points of connection between the vehicle and the road, it is critical for the controller to understand the limits of its available grip in order to suggest and execute safe commands to the vehicle. In such situations, the vehicle is operating in the nonlinear region range of the tire curve and is highly unstable. Small variations in the combined lateral and longitudinal load in the tires can induce a rapid variation in available grip.

The knowledge of sideslip angle is important for some controllers as it is often tracked and regulated by some vehicle dynamics controllers. Sideslip angle can actually be measured directly, however, it requires the use of very expensive sensors such as optical sensors pointed towards the ground. Optical sensors are also prone to damage due to their proximity to the road surface and are sensitive to obscuration from dirt. These would only be available for prototype and research vehicles as they would be too expensive to deploy on production vehicles. The alternative is to estimate sideslip using observers.

In control theory, observers are systems that provide an estimate of internal states of a given real system from measurements of the inputs and outputs of the system. They are used since often internal states of a system cannot be measured directly.

One of the most common approaches to estimate sideslip is using a variation of a Kalman filter, in which only IMUs are needed. However, using a variety of sensors such as cameras in addition to IMUs, a higher accuracy estimation of the vehicle state can be realized. This is important because the ultimately, the state of the vehicle drives the decisions made by the vehicle’s body controllers.

It is becoming increasingly more common for vehicles to be outfitted with high quality forward-facing cameras. The development of autonomous vehicles requires that the camera systems operate at a high performance (high frame rate and high resolution). While camera systems are becoming increasingly fast, it is generally still not as fast as traditional state estimation algorithms. Hence, the interim solution is to use a multi-rate solution, in which the camera system performs state estimation at a set rate while the traditional state estimator (e.g. Kalman filter) operates at a different frequency and the results of the two estimators are combined. This can be extended further by utilizing multiple cameras to perform state estimation to increase the robustness and accuracy of the overall estimation of the vehicle. This was the approach used by [35]. It is important to understand the states of the vehicle to predict when the vehicle will approach the limits of traction. Estimating when the vehicle will reach saturation and provide driver feedback will be important for future development.

This paper simply implemented a traditional Kalman filter estimator as input into the vehicle dynamics controllers to simulate sensor usage. The model used for the Kalman filter is the same model used for the low-level controller from Section 1.9.

Chapter 4 – Vehicle Performance

1.11Virtual Testing

One of the most effective ways to communicate data is through the use of a visualization tool such as the Unreal Engine. The Unreal Engine was used as MathWorks has recently begun integrating it with Simulink as part of their automated vehicles development toolchain. The Unreal Engine offers excellent flexibility to develop custom virtual scenery as well as objects that can be interacted with in the simulated environment. In addition to the visualization of the simulation, feedback can also be provided in the form of road friction changes and bumps on the road surface. Cameras, radars, and lidar sensors can also be simulated in the Unreal Engine virtual environment to provide additional data for developing automated vehicles. It is a powerful tool that is continuing to be developed on an ongoing basis and should enjoy a greater audience in the coming years as automated vehicles technology development progresses. An example of using the Unreal Engine is shown in Figure 46 and Figure 47, where a double lane change test case is illustrated.

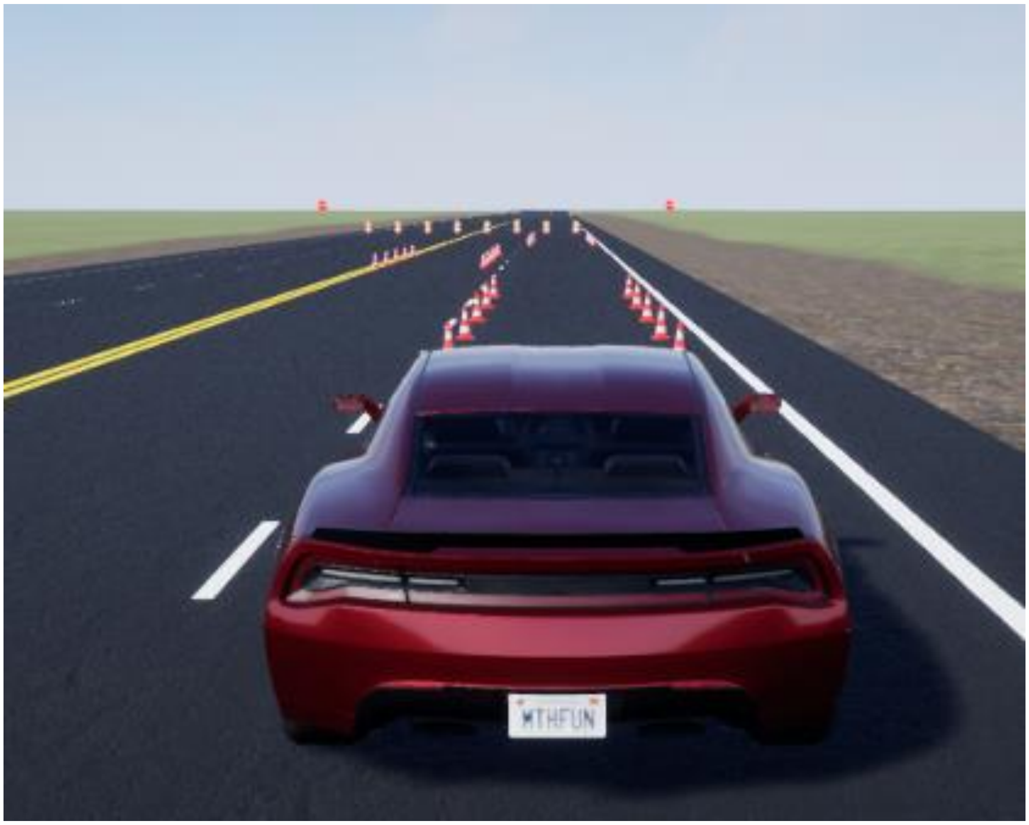


Figure 46 - Double Lane change test case, visualized using the Unreal Engine [36]

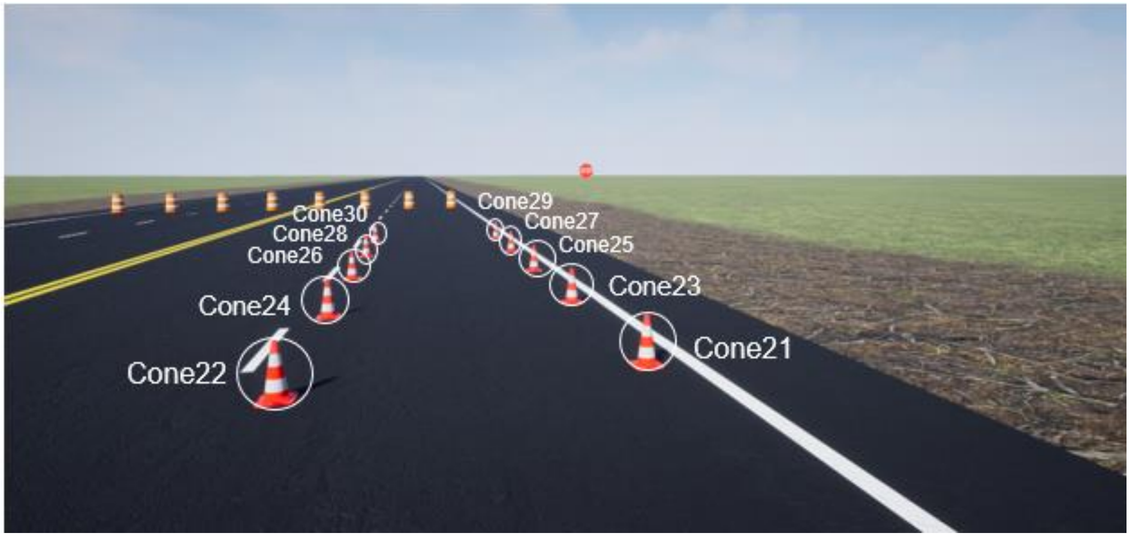


Figure 47 - Cones defined in the Unreal Engine virtual environment for a double lane change test case [36]

As shown in Figure 47, assets such as pylons can be created and widely used in the Unreal Engine. Other assets such as humans, animals, traffic infrastructure, and other vehicles can also be modelled and simulated. More information regarding asset and environment development is shown in Appendix B

Unreal Engine Assets and Pre-built Environments.

1.12Controller Performance

1.12.1Optimal Controller Performance

A few test cases of the vehicle through a double lane change maneuver were tested. The parameters varied include speed as well as weighting factors of the cost function used by the controller. Through trial and error, a set of feasible weighting factors were found to yield reasonable results.

The following set of test results stemmed from a double lance change maneuver with set nominal vehicle velocity of 13 m/s, as shown in Figure 48, Figure 49 and Figure 50. The controller yielded promising results as the vehicle was able to stay closely along the target path. The steering and acceleration profiles were smooth, and the error deviation was small. The vehicle was able to closely sustain its set velocity of 13 m/s without trouble. The torque vectoring system can be seen working effectively and as expected in Figure 50.

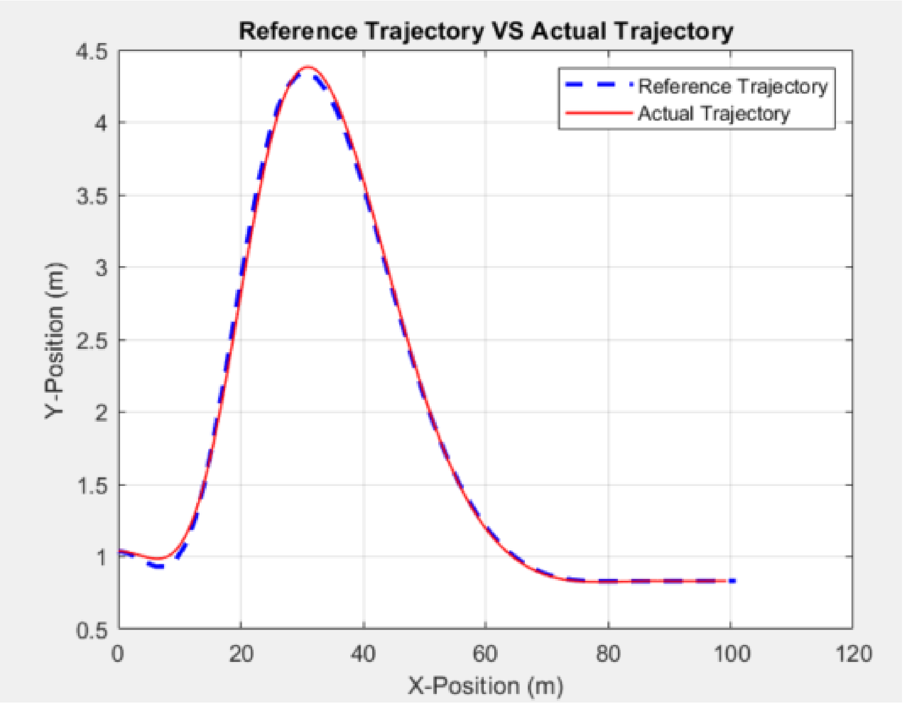


Figure 48 - Reference Trajectory vs. Actual Trajectory of the vehicle during a double lane change maneuver at 13 m/s nominal velocity

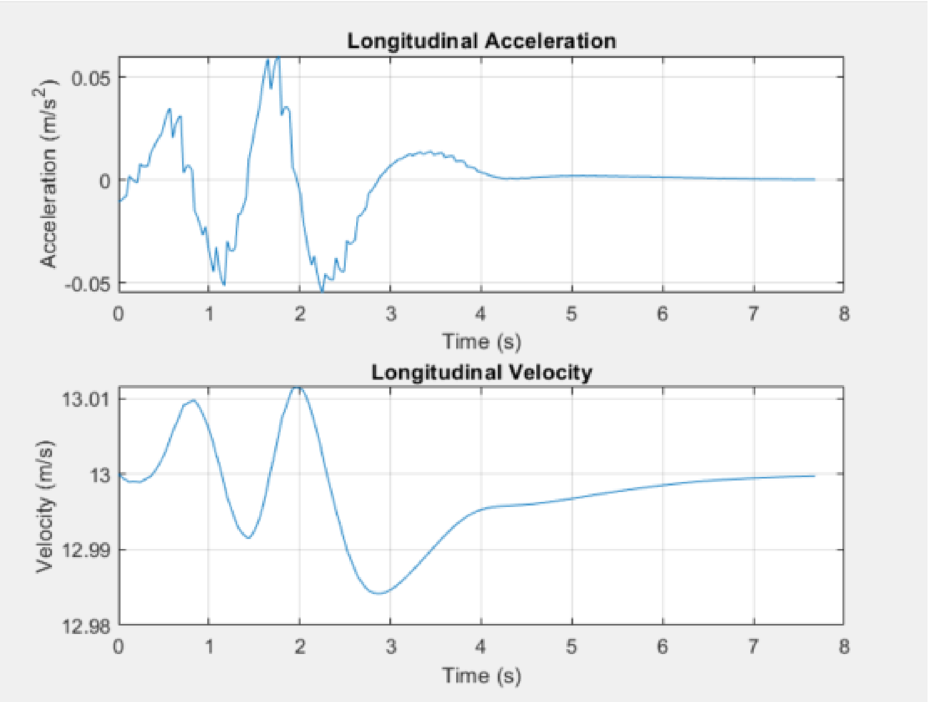


Figure 49 - Longitudinal and Lateral velocity of the vehicle during a double lane change maneuver at 13 m/s nominal velocity

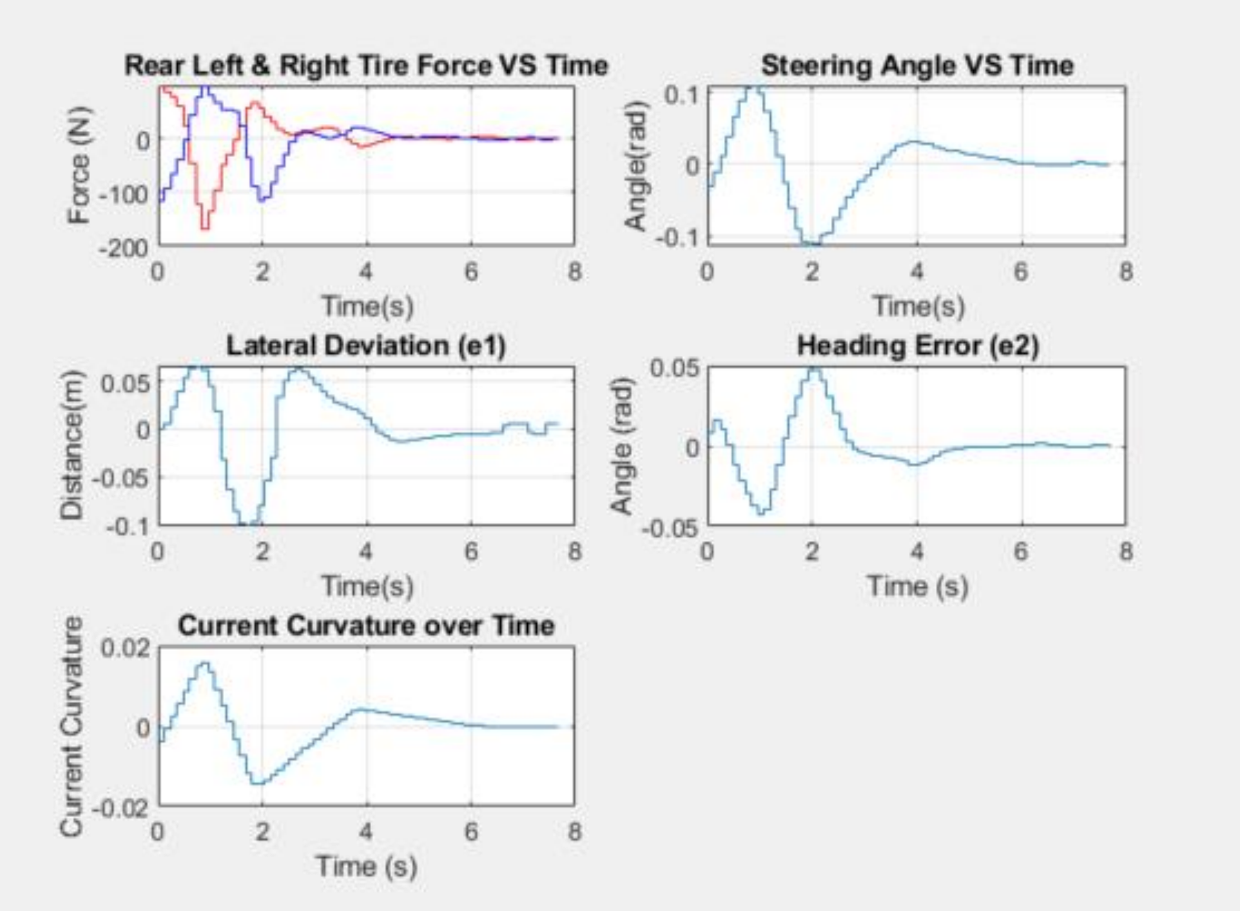


Figure 50 - Vehicle States during a double lane change maneuver at 13 m/s nominal speed

The following set of test results stemmed from a double lance change maneuver with set nominal vehicle velocity of 16 m/s, as shown in Figure 51, Figure 52, and Figure 53. The vehicle was still able to maintain the target trajectory, however, there is some signs of instability near the end of the maneuver where an oscillatory behavior can be seen in Figure 51. The acceleration plots also showed jerky behavior as a result of the oscillatory and opposing torques at the road wheels as well as the steering wheel as shown in Figure 53. This is indicative that the cost function needs to be reworked or weights tuned for this test case to lower sensitivity of certain parameters in the cost function.

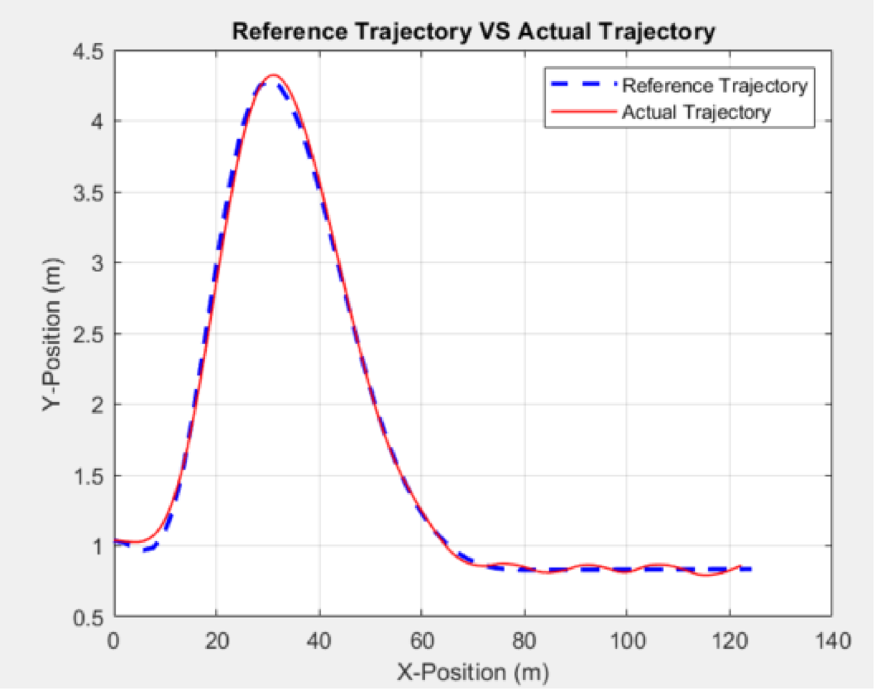


Figure 51 – Reference Trajectory vs. Actual Trajectory of the vehicle during a double lane change maneuver at 16 m/s nominal velocity

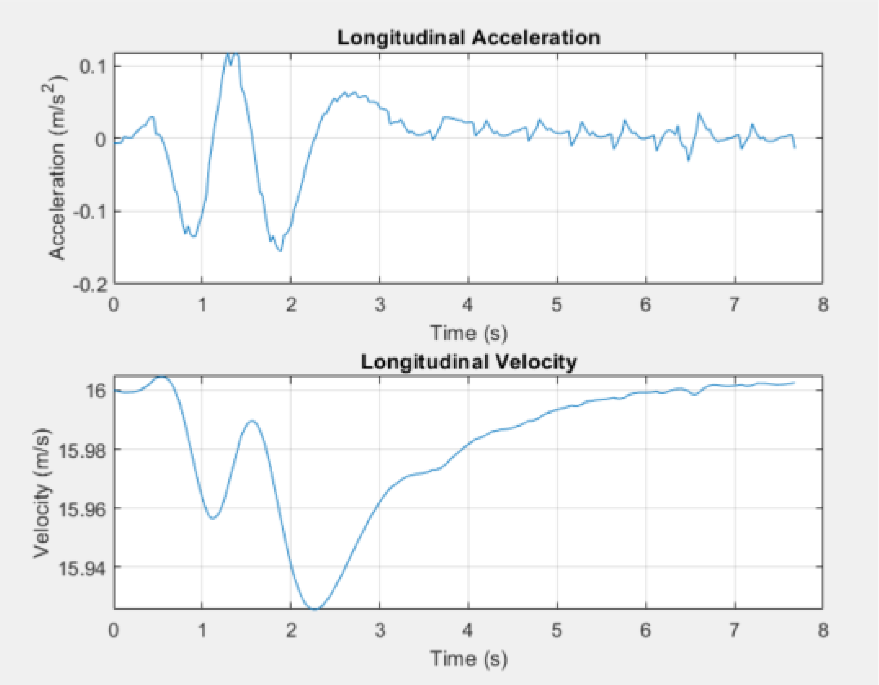


Figure 52 - Longitudinal and Lateral velocity of the vehicle during a double lane change maneuver at 16 m/s nominal velocity

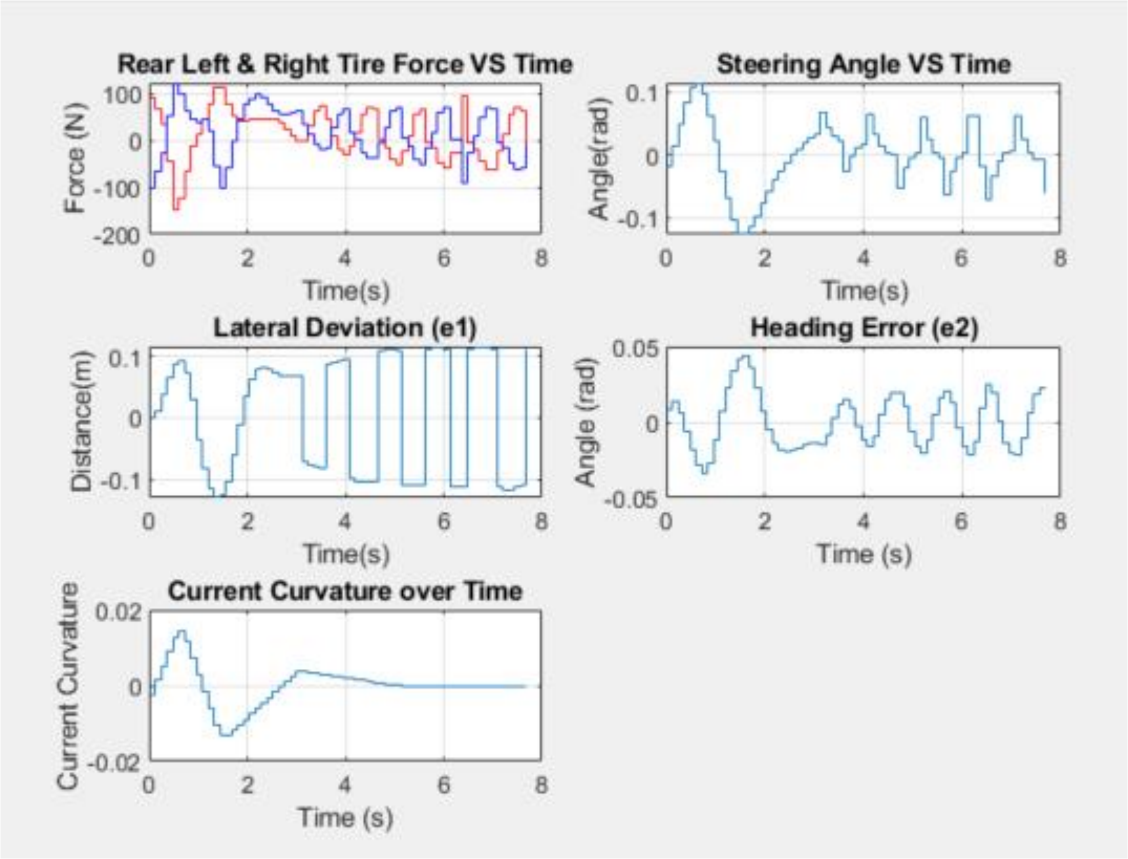


Figure 53 - Vehicle States during a double lane change maneuver at 16 m/s nominal speed

1.12.2Rule-Based Controller Performance

To test the rule-based controller, a skidpad test was used. In Figure 54, a reference path generated by a bicycle model under step steer was created and the ego vehicle (ego refers to the self) was subject to the same step steer maneuver. The ego vehicle was able to match the trajectory of the reference path by maintaining the yaw rate required for this maneuver, using the yaw rate of the bicycle model was the reference.

Next, more test cases were conducted to test the ability of the controller to augment the cornering ability of the vehicle by using torque vectoring while maintaining a fixed steering angle. This was done by offsetting the target yaw rate by 10% and passing that information to the vehicle dynamics controller as reference for feedback control. This is illustrated in Figure 55, where top left plot shows the ego vehicle achieving a tighter turning radius than the bicycle model reference path. This was due to the yaw moment generated by the electric motors resulting in higher yaw rate than previously shown in Figure 54, where only open loop steering created the yaw moment. It can be seen in Figure 55 that the motors are providing opposing torques to achieve this additional yaw moment to augment the yaw moment generated by the steered wheels.

Figure 56 is another test case in which the yaw rate was manipulated in an attempt to augment the cornering ability of the vehicle. In this test case, the yaw rate reference was set to 10% below the bicycle yaw rate reference. This caused the vehicle to generate opposing torque at the wheels that worked to counter the yaw moment generated by the steered wheels, resulting in a lower yaw moment overall on the vehicle as consequently, a lower yaw rate overall.

It is important to note the slight noise in the torque commanded to the motors. This can be observed in both Figure 55 and Figure 56. This noise in the torque command trickles down to the lateral acceleration and yaw rate of the vehicle. Whether this amount of noise is acceptable will require in-vehicle testing to quantify subjectively to see if it meets occupant expectations for ride comfort. This noise can be removed given more time to test and tune the controller.

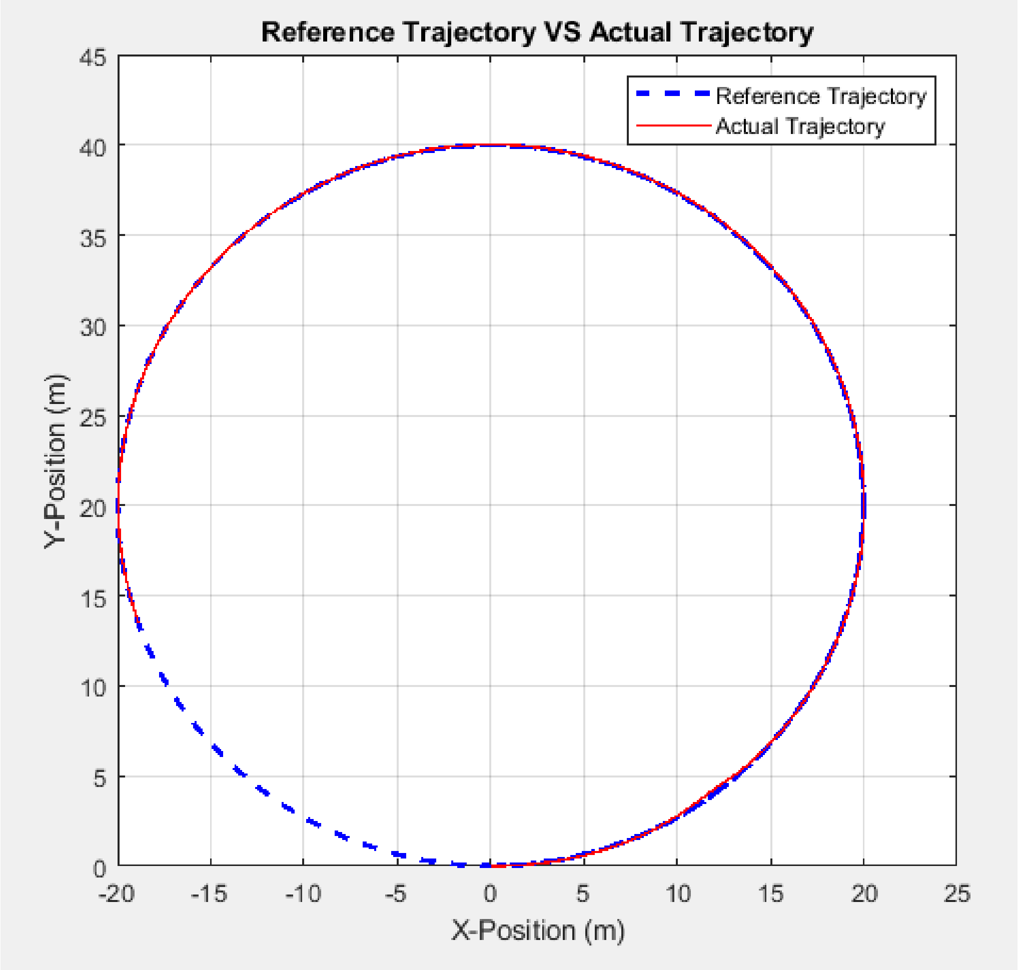


Figure 54 - Open Loop Driver, Closed Loop Controller Skid Pad Test

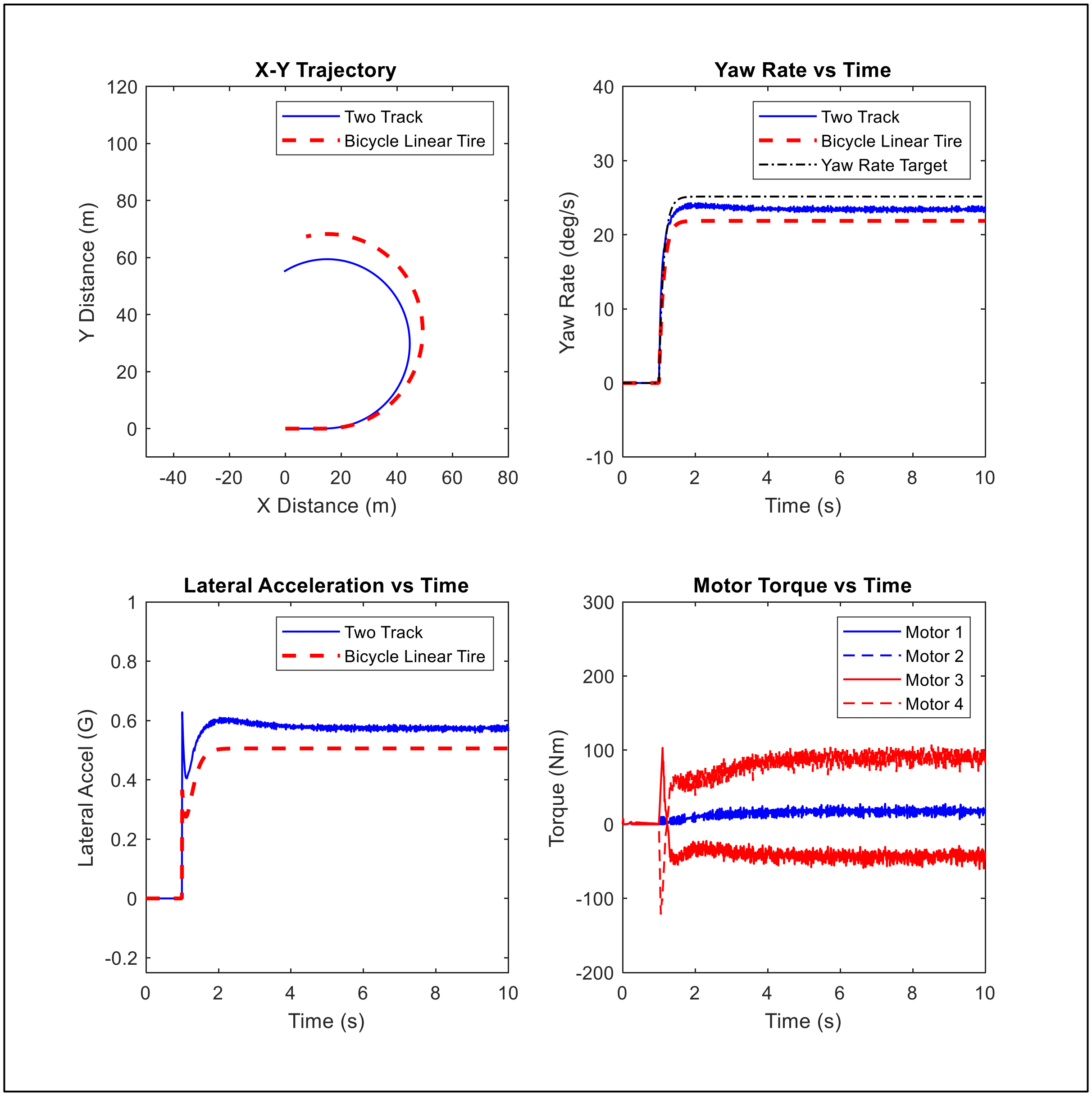


Figure 55 - Open Loop Driver, Closed Loop Controller Skid Pad Test while setting target yaw rate 10% higher than bicycle yaw rate reference. This caused the turning radius to tighten in comparison to the bicycle model.

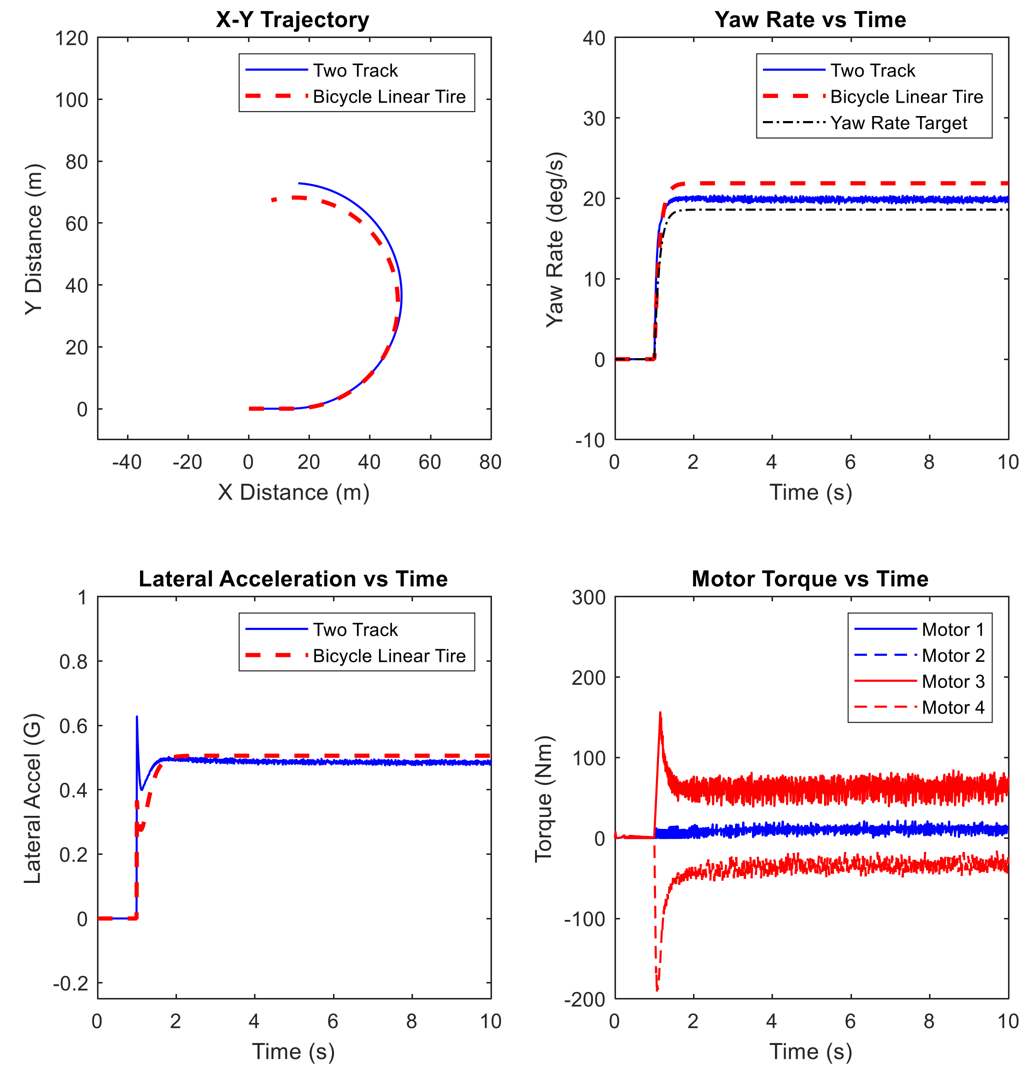


Figure 56 - Open Loop Driver, Closed Loop Controller Skid Pad Test while setting target yaw rate 10% lower than bicycle yaw rate reference. This caused the turning radius to loosen in comparison to the bicycle model.

Another advantage of using torque vectoring is the ability to increase lateral handling capability of the vehicle. In a traditionally front wheel-steered vehicle, the yaw moment is only generated by the front wheels. This means that the front tires need to support lateral force as well as provide the force to rotate the vehicle. This often leads to the underutilization of the rear tires, and ultimately, underutilization of the total performance of the tires overall. Torque vectoring allows the rear tires to assist the front tires in providing the yaw moment required in a cornering situation and thus offload the load from the front tires. This leads to a greater total lateral acceleration capability of the vehicle and require less steering angle to provide the same centripetal force to maintain a constant turning radius. This is shown in Figure 57, where various test cases were examined and their lateral response to steering angle are illustrated.

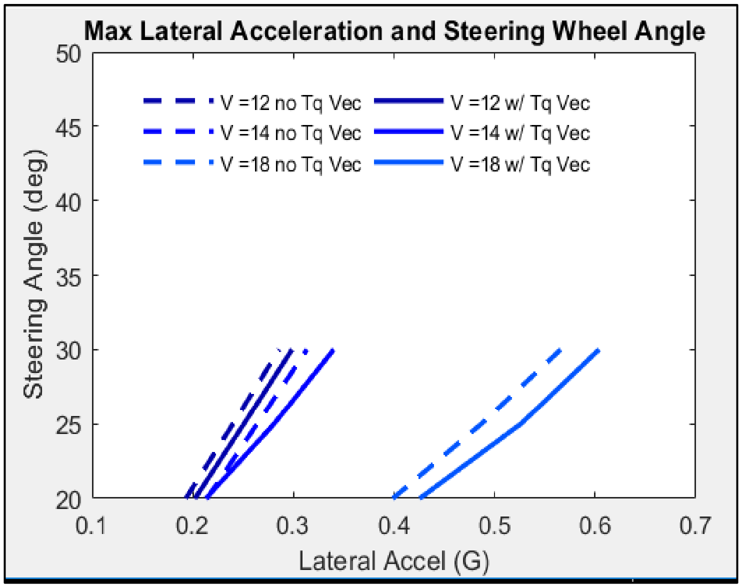


Figure 57 - Maximum lateral acceleration improvement in cornering as a result of the torque vectoring controller

In addition to the increased lateral handling capability, the vehicle is also more responsive. Figure 58 shows that the vehicle using the torque vectoring system was able to reduce the delay between driver input and yaw rate response. The plots are normalized to one so that the trend can be compared.

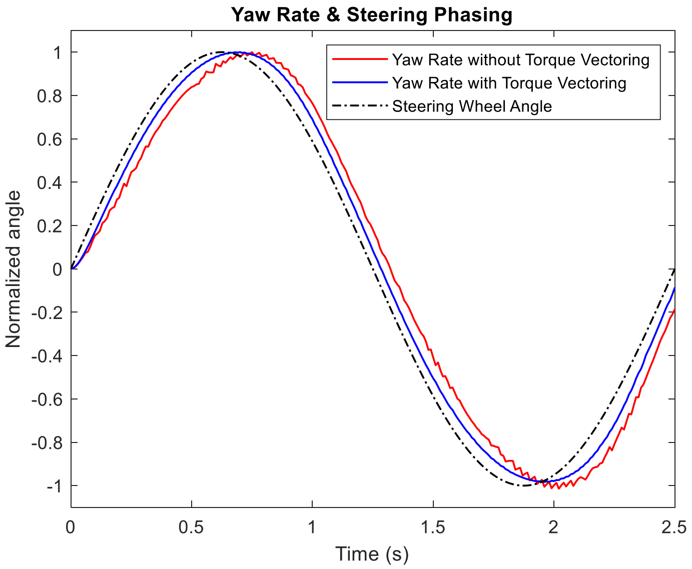


Figure 58 - Yaw Rate & Steering Phasing improvement as a result of the usage of the torque vectoring controller

Chapter 5 – Conclusions

1.13Summary

Contribution:

This research achieved three objectives:

1. Vehicle models and different controllers were developed for automated driving tasks for the EcoCAR Mobility Challenge.
2. A hierarchical framework for vehicle dynamics control for a custom hybrid electric Chevrolet Blazer was developed.
3. A visualization framework for developing CAV and control systems was set up.

This research has presented various vehicle dynamics models that will be used by UWAFT for simulating vehicle dynamics as part of the effort to develop powertrain and ADAS controllers. The models are robust and span a variety of complexities to meet the various needs of the team. Validation methods and processes were developed for both the EcoCAR 3 Camaro as well as the EcoCAR Mobility Challenge Blazer. Further work in correlating vehicle models with the physical vehicle will need to be done in order to properly develop the controllers. This process will take place once the UWAFT Blazer has been physically built and in running order. Currently, only hand calculations were used to validate the vehicle dynamics models.

A hierarchical control framework was also proposed and implemented for the development of powertrain and ADAS controllers for the UWAFT Blazer. The framework allows the various sub systems to be developed simultaneously by multiple sub teams. This allows the subsystems to be evaluated and independently tested. These individual sub systems make use of the various vehicle dynamics models developed in the earlier chapter to streamline the development process by providing the correct balance of speed and accuracy to meet the demands of the sub team’s requirements. A high-level controller was used to determine the optimal trajectory for the vehicle by taking into account the constraints of the vehicle, road, and the user-defined cost function. The cost function penalized actions that potentially put the user at risk by minimizing the peak accelerations of the vehicle. This high-level controller was based on optimal control theory and utilized a point mass model of the vehicle in order to project and predict over an extended time horizon. A low-level controller was developed to regulate and track the vehicle based on the inputs from the high-level controller. Two low-level controllers were developed: an optimal control-based controller and a rule-based controller. The optimal control-based controller utilized a more complicated model than the one found in the high-level controller as the low-level controller operated on a shorter time horizon and needed the increased fidelity to properly and accurately track the reference trajectory. A two-track model was developed and used for this purpose. The second low-level controller developed was a rule-based controller and this controller contained several layers within to achieve various tasks. It contained a PID controller within to determine the overall torque required to propel the vehicle forward while also meeting yaw rate requirements. It also contained a torque scheduled strategy for distributing the torque among the wheels so that a pseudo-optimal distribution of load effort was achieved.

Lastly, a framework for integrating the Unreal Engine with MATLAB/Simulink was set up. This will allow future development of ADAS systems to be more effective as highly representative visualizations of the environment can be simulated, including traffic infrastructure, pedestrians, animals, and other vehicles. This simulated environment can greatly supplement ADAS development and reduce the need to exclusively collect data from a physical test vehicle.

1.14Future Work and Recommendations

1. The vehicle models developed were vetted to the greatest extent possible with the resources available. Once the UWAFT Blazer has arrived and a test plan developed, model correlation can begin and further improve the accuracy of the vehicle dynamics models. The team is on track to begin testing and model correlation by early October 2019.

2. An accurate tire model is very important to accurately represent the vehicle. Tire data should be pursued; GM may be able to provide data for tires.

3. Further refinement of the controllers will increase their robustness and optimality. While the latest version of the UWAFT Blazer does not have the components allowing for torque vectoring, the other aspects of optimal control can still be utilized and improved upon for increasing safety and efficiency of the powertrain and ADAS systems. This will be the focus of UWAFT in the coming years as the competition has placed a much greater emphasis on efficiency and intelligent vehicle control in a connected environment.

4. With model predictive control, constraints can be defined such that they must be adhered to and has precedence over the minimization of the MPC's objective function, normally used for system regulation and tracking. However, it does not natively consider the priority of constraints over other constraints. The ability to prioritize certain constraints over other constraints add another layer of control. For example, constraints that prevent collision with obstacles (obstacle constraints), can be prioritized to minimize injury and property damage. In another example, collisions with pedestrians could cause injury, whereas collisions with other vehicles may only cause property damage. Similarly, collisions with an empty sidewalk may not cause any damage or injury at all. Therefore, a prioritization would likely hold pedestrians, vehicles, and empty sidewalks prioritized in that order [37]. This would be a useful feature to incorporate in an MPC and would be suitable to explore in the future.

5. The current MPC model uses the same vehicle dynamics model for simulation as for the prediction model in the MPC. This is a simplification to minimize variables. The vehicle dynamics model will need to be updated with a higher fidelity model in order to more accurately represent reality.

6. It is very important that all controllers are tested thoroughly for edge case to ensure the safe operation of the vehicle at all times. In particular, the vetting of a linear MPC is pertinent to the work completed in this thesis.

7. Further development of custom maps using the Unreal Engine will increase the range of scenarios possible for testing. Increased exposure to unique environments and test cases will improve the robustness of the controllers ADAS controllers and should be prioritized for the ADAS team.

2 References

[1] P. K. A. M. E. S. Kersten Heineke, "Mckinsey," [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/self-driving-car-technology-when-will-the-robots-hit-the-road. [Accessed 5 February 2019].

[2] Public Sector Consultants, "CarGroup," [Online]. Available: https://www.cargroup.org/wp-content/uploads/2017/03/Planning-for-Connected-and-Automated-Vehicles-Report.pdf. [Accessed 22 October 2018].

[3] R. Verger, "Popular Science," [Online]. Available: https://www.popsci.com/self-driving-car-race-track/. [Accessed 9 May 2019].

[4] Ayers, "Ayers and Whitlow," 5 January 2018. [Online]. Available: https://www.ayersandwhitlow.com/blog/2018/01/nhtsa-nearly-all-car-crashes-are-due-to-human-error.shtml.

[5] A. Hawkins, "Waymo gets green light in California to pick up passengers in self-driving cars," The Verge, 3 July 2019. [Online]. Available: https://www.theverge.com/2019/7/3/20680938/waymo-self-driving-cars-passengers-permit-california-pick-up.

[6] Intel, "Autonomous Driving," Intel, [Online]. Available: https://newsroom.intel.com/press-kits/autonomous-driving-intel/. [Accessed 23 March 2019].

[7] "The Journey to Zero Accidents," [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-platform/. [Accessed 12 February 2019].

[8] S. o. A. Engineers, "SAE," January 2014. [Online]. Available: https://web.archive.org/web/20161120142825/http://www.sae.org/misc/pdfs/automated_driving.pdf. [Accessed June 2019].

[9] "Todays Motor Vehicles," 5 March 2019. [Online]. Available: https://www.todaysmotorvehicles.com/article/40-countries-agree-to-automatic-emergency-braking-mandate-us--to-rely-on-voluntary-compliance/. [Accessed 19 June 2019].

[10] M. Laris, Washington Post, 5 December 2018. [Online]. Available: https://www.washingtonpost.com/local/trafficandcommuting/waymo-launches-nations-first-commercial-self-driving-taxi-service-in-arizona/2018/12/04/8a8cd58a-f7ba-11e8-8c9a-860ce2a8148f_story.html?noredirect=on&utm_term=.98eb5ae4e1bc. [Accessed 19 June 2019].

[11] NHTSA, August 2014. [Online]. Available: http://www.nhtsa.gov/staticfiles/rulemaking/pdf/V2V/V2V-ANPRM_081514.pdf. [Accessed 20 June 2019].

[12] J. G. Fernandez, "A Vehicle Dynamics Model for Driving Simulators," Chalmers University of Technology, Sweden, 2012.

[13] S. S. Selahattin Caglar Baslamisli, "Construction of a Rational Tire Model for High Fidelity Vehicle Dynamics Simulation Under Extreme Driving and Environmental Conditions," in ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, 2010.

[14] "Modelling a Vehicle Dynamics System," MathWorks, [Online]. Available: https://www.mathworks.com/help/ident/examples/modeling-a-vehicle-dynamics-system.html. [Accessed 6 January 2019].

[15] J.-H. O. J.-H. P. S. K. D. L. Jaemiun Yoon, "Autonomous Dynamic Driving Control of Wheel Mobile Robots," IEEE, 2014.

[16] D. B. Ayalew, AuE850 Lecture 3, Clemson University.

[17] B. M. X. M. D. M. A. T. Moad Kissai, "Adaptive Robust Vehicle Motion Control for Future Over-Actuated Vehicles," 17 April 2019. [Online]. Available: https://www.mdpi.com/2075-1702/7/2/26/htm.

[18] "Tire-Road Interatction (Magic Formula)," [Online]. Available: https://www.mathworks.com/help/physmod/sdl/ref/tireroadinteractionmagicformula.html. [Accessed 5 January 2019].

[19] M. Furlan, "MFeval," MATLAB Central File Exchange, [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/63618-mfeval. [Accessed 1 April 2018].

[20] "Center of Gravity Height," [Online]. Available: http://www.longacreracing.com/technical-articles.aspx?item=42586. [Accessed 5 October 2016].

[21] "The Coefficient of Restitution of a bouncing ball," [Online]. Available: https://physics.stackexchange.com/questions/172127/the-coefficient-of-restitution-of-a-bouncing-ball. [Accessed 14 September 2017].

[22] M. Nagurka, "ResearchGate," Jan 2003. [Online]. Available: https://www.researchgate.net/figure/Ball-height-vs-time-for-first-few-bounces-assuming-no-aerodynamic-drag_fig2_228575622.

[23] "Car Buzz," [Online]. Available: https://carbuzz.com/news/2020-chevrolet-blazer-receiving-a-slick-styling-upgrade. [Accessed 24 October 2018].

[24] J. P. V. M. F. N. David Bautista, "A Review of Motion Planning Techniques for Automated Vehicles," Transactions on Intelligent Transportation Systems, 2015.

[25] Gorinevsky, Lecture 14 - Model Predictive Control, Palo Alto: Stanford.

[26] "Understanding Model Predictive Control," [Online]. Available: https://www.mathworks.com/videos/series/understanding-model-predictive-control.html. [Accessed 25 November 2017].

[27] A. Bemporad, "Model Predictive Control," University of Trento, 1 January 2010. [Online]. Available: http://cse.lab.imtlucca.it/~bemporad/teaching/ac/pdf/AC2-10-MPC.pdf.

[28] MathWorks, "Optimization Problem," 8 June 2019. [Online]. Available: https://www.mathworks.com/help/mpc/ug/optimization-problem.html.

[29] M. Gerdts, "Solving mixed-integer optimal control problems by Branch & Bound: A case study from automobile test-driving with gear shift," Optimal Control Applications and Methods, vol. 26, pp. 1-18, 2005.

[30] "Global Optimal Path Planning of an Autonomous Vehicle for Overtaking a Moving Obstacle," January 2014. [Online]. Available: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1679-78252014001400002.

[31] "Lane Following Using Nonlinear Model Predictive Control," [Online]. Available: https://www.mathworks.com/help/mpc/ug/lane-following-using-nonlinear-model-predictive-control.html. [Accessed 3 May 2019].

[32] A. Marashian, "ResearchGate," [Online]. Available: https://www.researchgate.net/post/What_is_Circle_of_Friction2. [Accessed 8 Feb 2019].

[33] E. V. D. C. D. T. E.N. Smith, "Evaluation of Optimal Yaw Rate Reference for Electric Vehicle Torque," CRC Press, Munich, 2016.

[34] D. S. J. T. L. D. N. Bernhard Knauder, "Electric Torque Vectoring for Electric Vehicles".

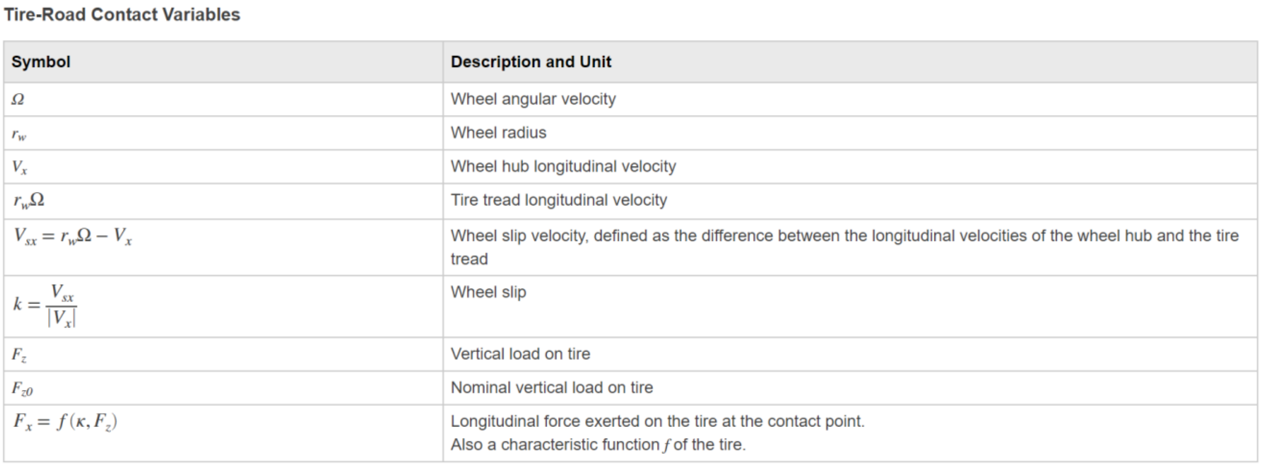
[35] B. M. N. Yafei Wang, "Vision-based vehicle body slip angle estimation with multi-rate Kalman filter considering time delay," The University of Tokyo, Tokyo, 2012.

[36] MathWorks, "Vehicle Dynamics Blockset User Guide," MathWorks, 2019. [Online]. Available: https://www.mathworks.com/help/pdf_doc/vdynblks/vdynblks_ug.pdf.

[37] Y. Rasekhipour, "Prioritized Obstacle Avoidance in Motion Planning of Autonomous Vehicles," University of Waterloo, 2017.

Appendix A

Tire Road Contact Variables



Tire-Road Contact Variables

Symbol	Description and Unit
Ω	Wheel angular velocity
r_w	Wheel radius
V_x	Wheel hub longitudinal velocity
$r_w\Omega$	Tire tread longitudinal velocity
$V_{sx} = r_w\Omega - V_x$	Wheel slip velocity, defined as the difference between the longitudinal velocities of the wheel hub and the tire tread
$k = \frac{V_{sx}}{\lvert V_x \rvert}$	Wheel slip
F_z	Vertical load on tire
F_{z0}	Nominal vertical load on tire
$F_x = f(\kappa, F_z)$	Longitudinal force exerted on the tire at the contact point. Also a characteristic function f of the tire.

Appendix B

Unreal Engine Assets and Pre-built Environments

(https://www.mathworks.com/help/pdf_doc/vdynblks/vdynblks_ug.pdf)

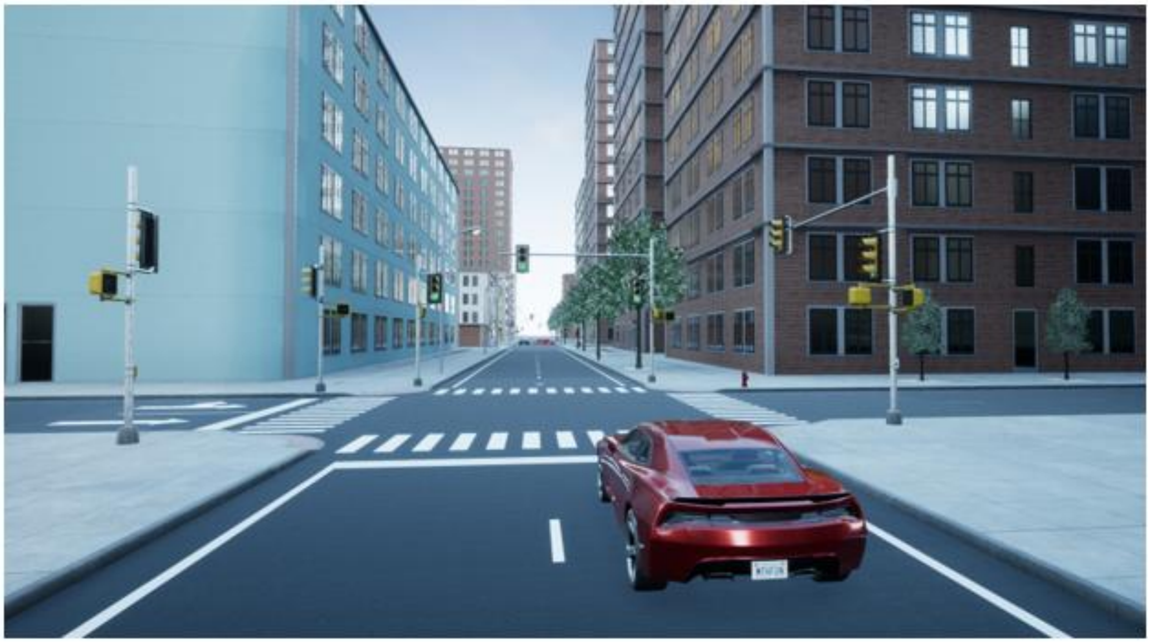


Figure 59 - Vehicle in City Block Map

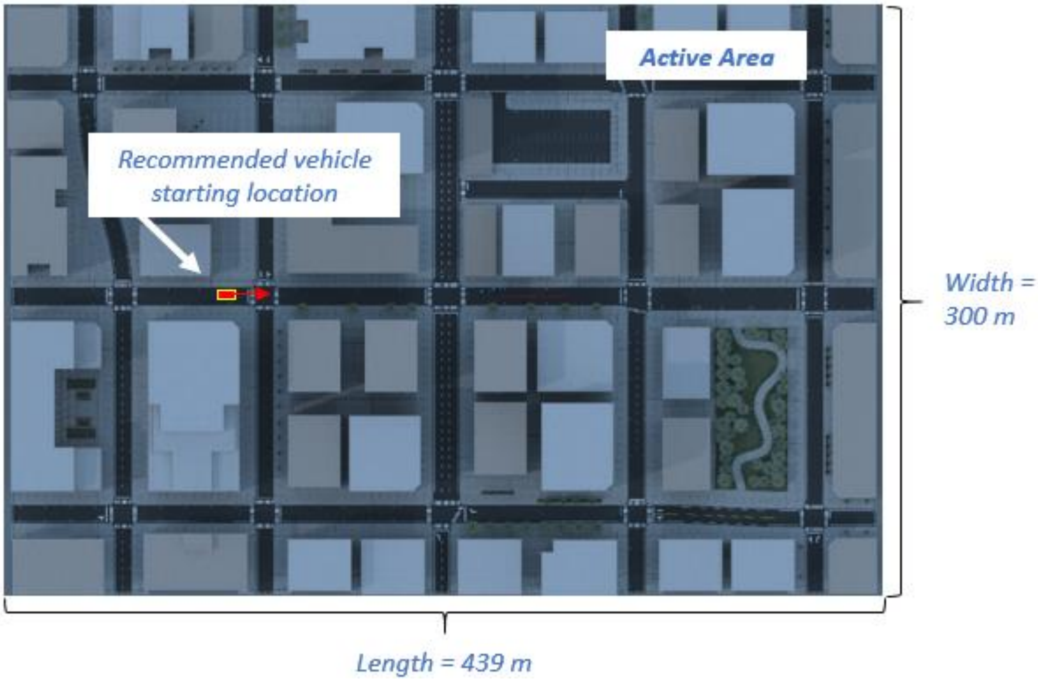


Figure 60 - Schematic of City Block Map

Figure 61 - Vehicle in M-City

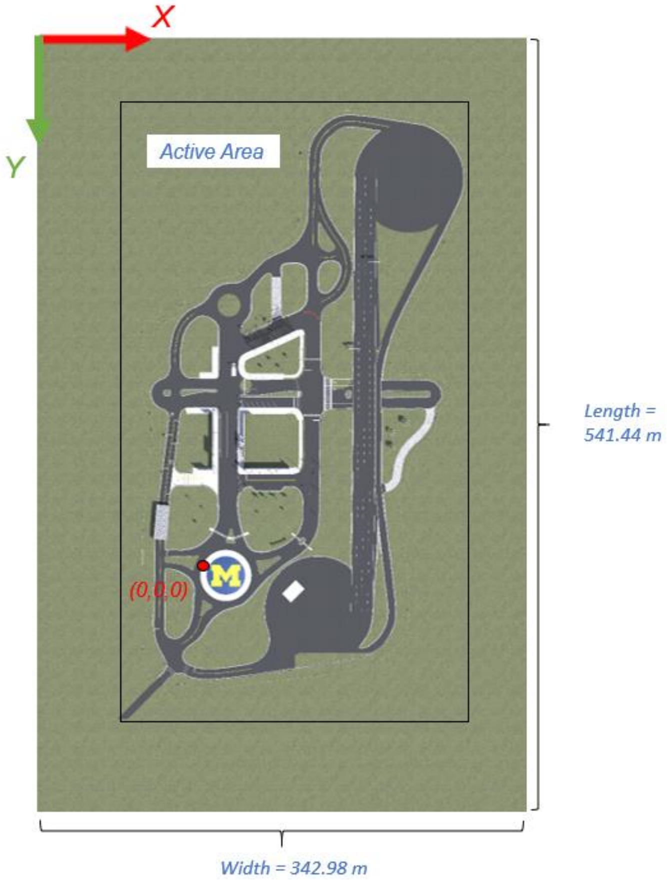


Figure 62 - Schematic of M-City

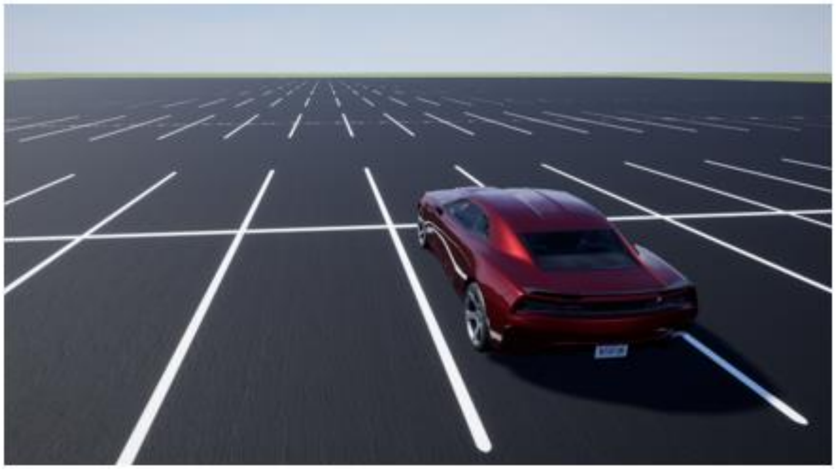


Figure 63 - Vehicle in Parking Lot Map

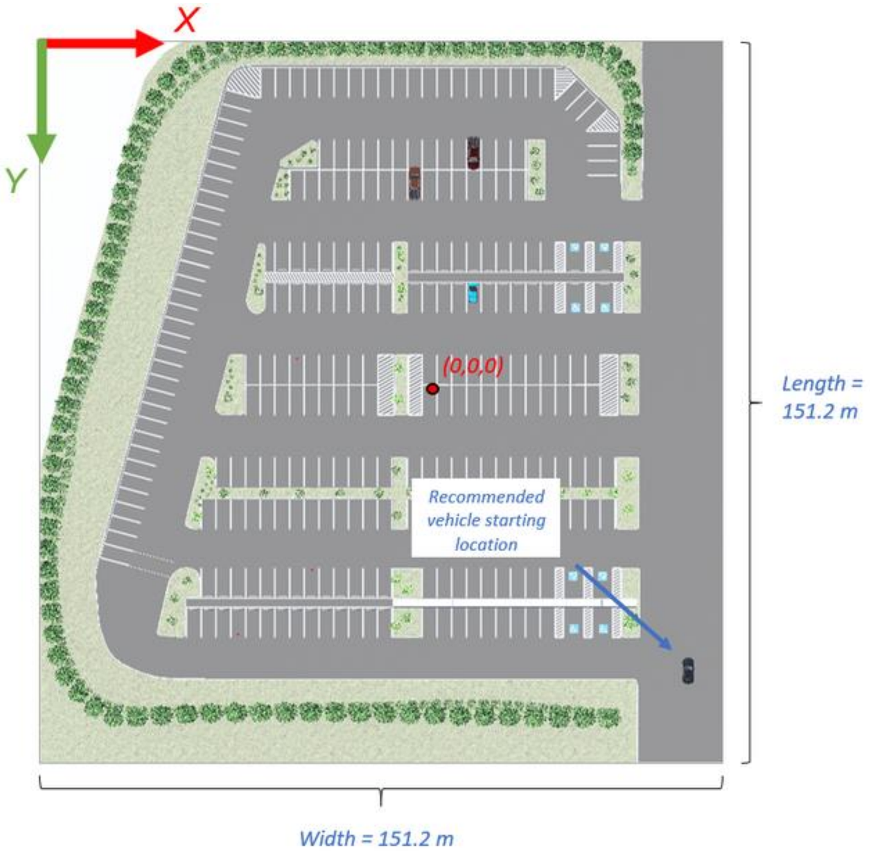


Figure 64 - Parking Lot Map

Figure 65 - Vehicle in Highway Map

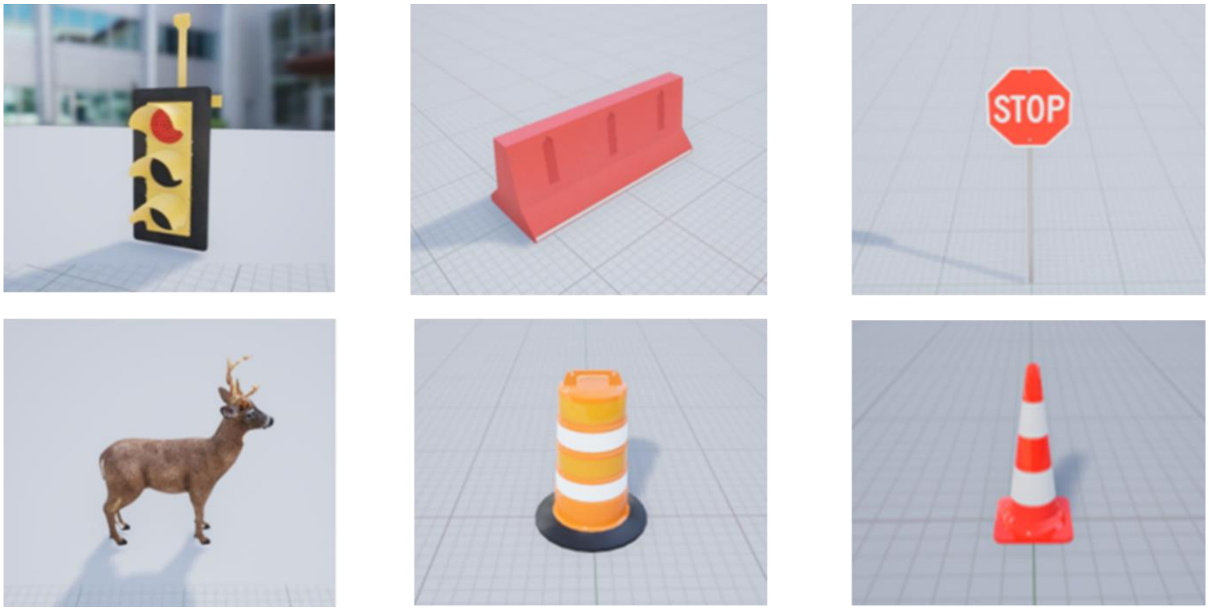


Figure 66 - Some assets available in Unreal Engine

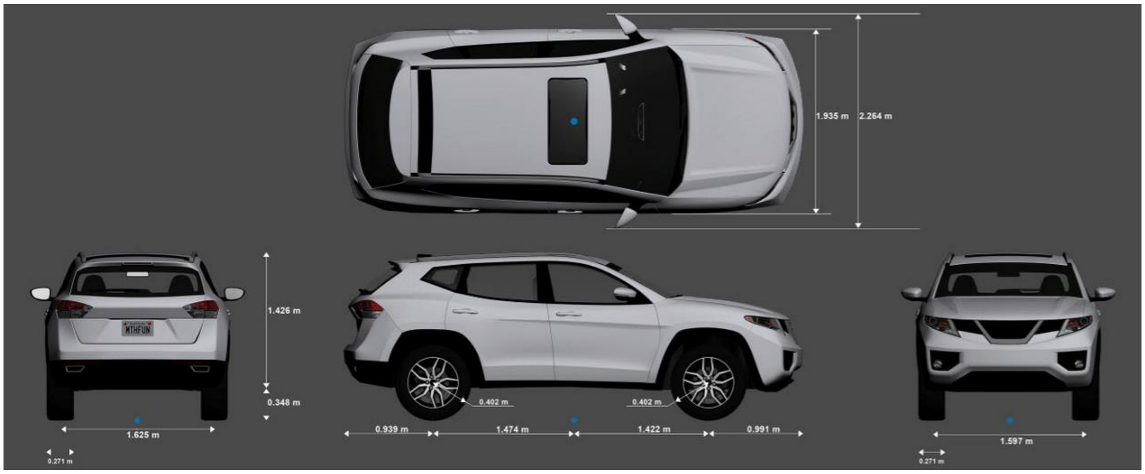


Figure 67 - Prebuilt SUV in Unreal Engine

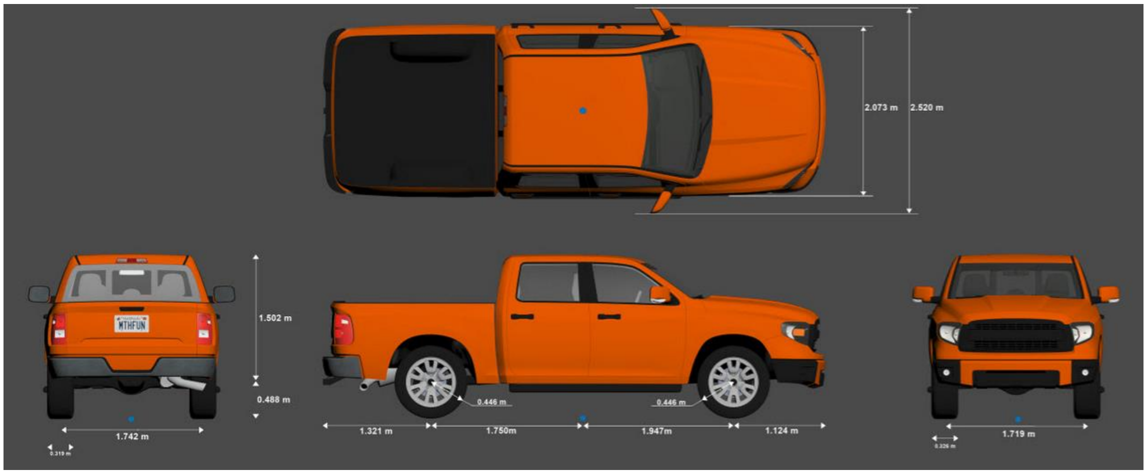


Figure 68 - Prebuilt Truck in Unreal Engine

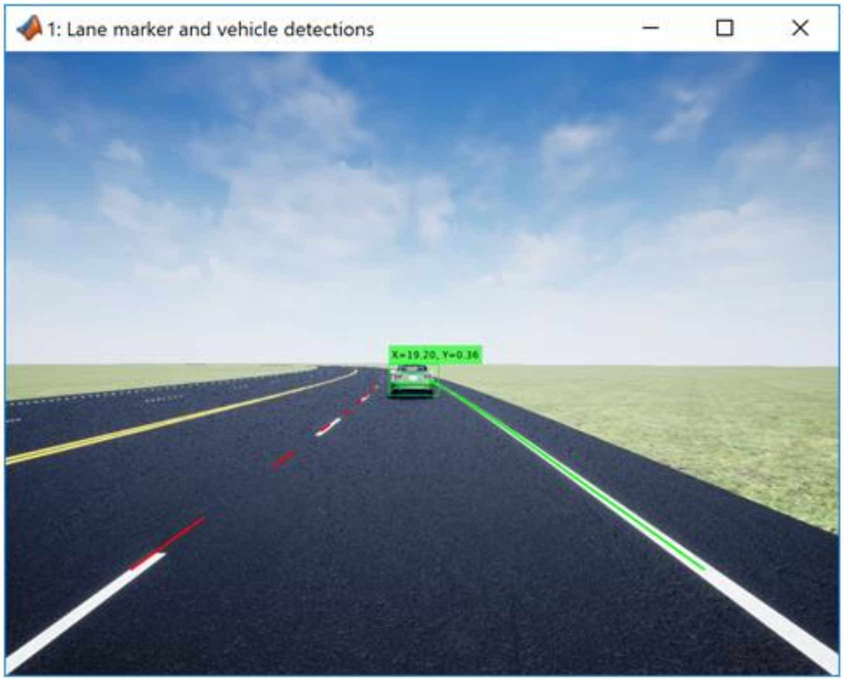


Figure 69 - Vehicle and Lane detection within the Unreal Engine

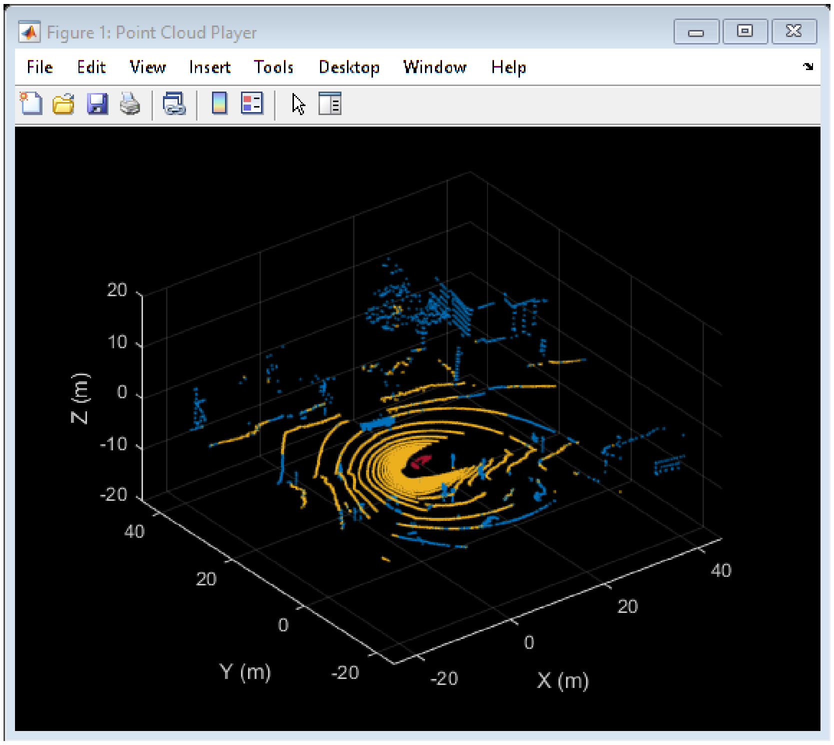


Figure 70 - Lidar map of surroundings within Unreal Engine, plotted using MATLAB