Modeling and Control of a Four Wheel Drive Formula SAE Car

by

Cheyenne D. Hua

Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2019

Signature redacted

Signature of Author: _____
Department of Mechanical Engineering
May 17, 2019

Signature redacted

Certified by: _____
David L. Trumper
Professor of Mechanical Engineering
Thesis Supervisor

Signature redacted

Accepted by: _____
Maria Yang
Associate Professor of Mechanical Engineering & Engineering Systems
Undergraduate Officer

Modeling and Control of a Four Wheel Drive Formula SAE Car

by

Cheyenne D. Hua

Submitted to the Department of Mechanical Engineering
on May 19, 2019 in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

ABSTRACT

Formula SAE is a collegiate design competition in which student teams design, build, and race an electric formula racecar every year. In 2019, the MIT team built its first four wheel drive vehicle. The new architecture requires more robust and performant control systems. One major challenge is that the vehicle is not functional for the majority of the year. A longitudinal vehicle simulation was written and tested for the purpose of testing control algorithms without a physical testbed, as well as to learn more about vehicle behavior in general. The simulation was written in Simulink and the structure kept versatile so that it could be easily expanded in complexity in future years. Test data was used to successfully correlate the model to the actual system. Several launch control algorithms were also tested using this simulation, for both a rear wheel drive and four wheel drive architecture. Although basic, the control schemes produced promising results for both speed and stability, notably the normal force proportional controller.

Thesis Supervisor: David L. Trumper

Title: Professor of Mechanical Engineering

TABLE OF CONTENTS

1. INTRODUCTION

1.1 FORMULA SAE

Formula SAE (FSAE) is a collegiate design competition where student teams design, build, and race an open wheel formula-style racecar. The competition is structured around a series of scored events, both dynamic – the car races against the clock – or static – the team presents the design and other aspects of the vehicle. The dynamic events consist of acceleration (75m drag race), skidpad (steady state cornering), autocross, and endurance (22km autocross). The determination of success relies upon many factors, the main factors being: vehicle reliability, vehicle performance, and driver performance and preparation. The electric category in the United States was started in 2011, and has since grown to over thirty teams. The competition is held annually in Lincoln, Nebraska.

1.2 TEAM BACKGROUND

MIT Motorsports consists of about 40 people, mostly undergraduates from mechanical, electrical, and aerospace engineering, and computer science. The team is entirely student-run, including auxiliary functions such as fundraising and recruiting.

MIT has competed in the electric vehicle category since 2014. The first few vehicles were barely functional; the team's goal in those years was simply to make the car work and learn how to engineer a racecar. The Model Year 2017 (MY17) vehicle was the first major success for the team. It was a rear wheel drive car and its primary goal was reliability.

MY18 was MIT's last rear wheel drive car (Figure 1). It was designed as an iteration of the MY17 car, maintaining its simple and reliable architecture, while reworking some subsystems to increase performance. It performed remarkably well at the Lincoln competition, winning the acceleration and autocross events and placing second in skidpad. In the endurance event, the team was disqualified after the race due to a violation of the 80 kW power limit. A post-mortem analysis indicated that the root cause of this violation was lack of testing time and a power limiting controller that was not tuned well enough. This motivated the creation of a controls team whose focus is to understand the vehicle as a system and ensure stable, and legal, vehicle performance. Despite the disqualification, the vehicle was clearly reliable, fast, and comfortable to drive. Following the momentum of MY18's success, the team chose to make an engineering leap and design MY19 to have a four wheel drive architecture. This put further pressure on the control systems as they are much more essential to the stability of a four wheel drive vehicle than a rear wheel drive vehicle. The specifications for the two vehicles are detailed in Table 1.

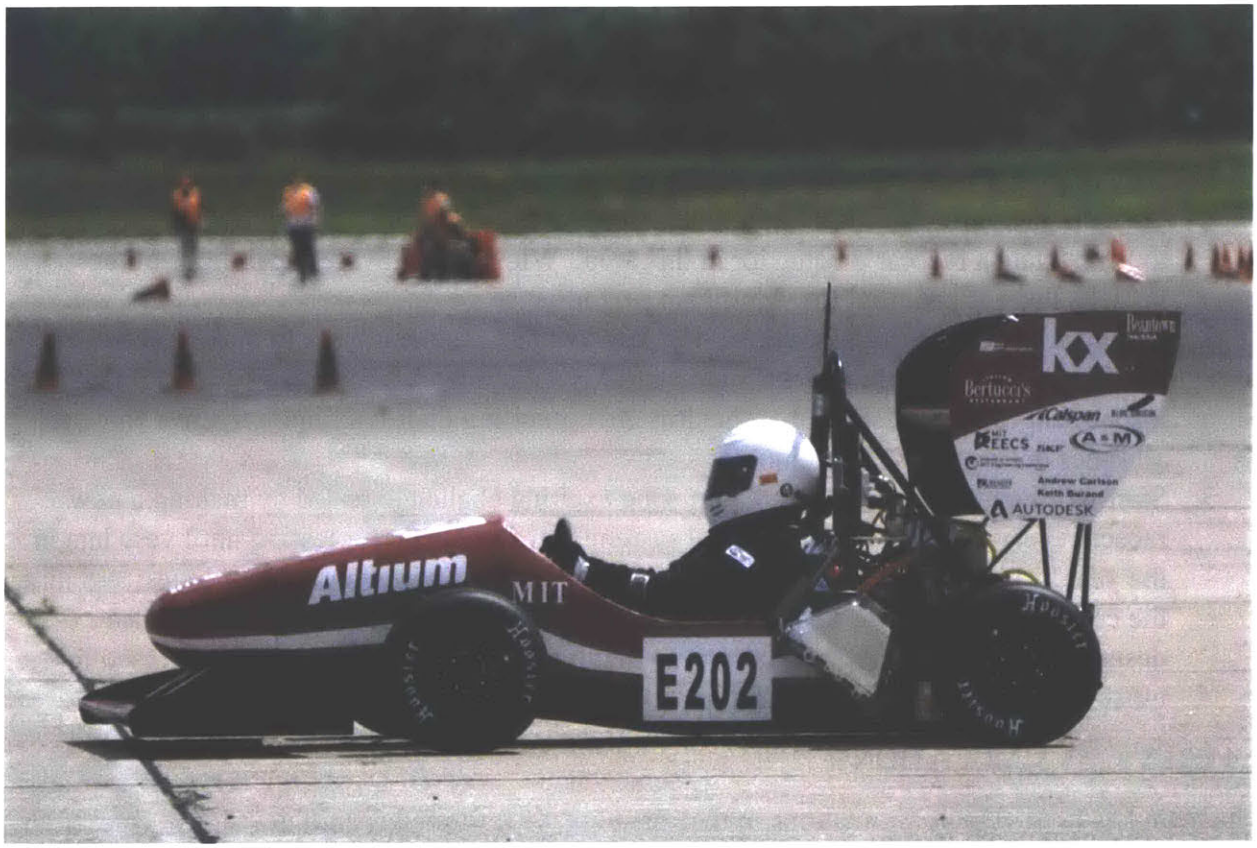

Figure 1: The Model Year 2018 vehicle in the endurance race at Lincoln, NE.

Metric	MY18	MY19 (predicted)
0-75m time	4.12 s	3.7 s
Top speed	40 m/s (90 mph)	40 m/s (90 mph)
Weight	230 kg	250 kg
Tires	18 x 6" Hoosier	18 x 6" Hoosier
Aero	Front wing, active rear wing	Front wing, active rear wing
Rear powertrain	Single Emrax 228 (axial flux) Limited slip differential @rear axle	Single Emrax 228 (axial flux) Limited slip differential @rear axle
Front powertrain (per wheel)	None	DHX Hawk 40 Planetary gearbox

Table 1: Specifications for MY18 and MY19.

1.3 OBJECTIVES

There were three main motivations for this project.

1. Learn more about straight line motion. Two of the most important performance metrics for a racecar are straight line acceleration and steady state cornering acceleration. The less complex of the two is straight line acceleration, so the first objective of the project is to investigate a racecar's behavior in this area. With the hardware capabilities of the vehicle, there is potential for significant performance gains.
2. Design a launch control algorithm. The key to a fast acceleration is to make full use of the tires' tractive capability. Launch control is a method to automatically control the torque sent to the wheels so that the tires are outputting their maximum tractive force at all times during the acceleration.
3. Start writing a full vehicle simulation. One exciting challenge posed by making a new racecar every year is that the physical system is not available for testing until very late in the year. This motivated the development of a vehicle simulation in order to test and tune the controllers before the physical testbed was functional. The simulation was also designed to have a versatile structure, so that it could be easily expanded to be used for other dynamic situations in the future.

1.4 PREVIOUS SIMULATIONS

The team has used vehicle simulations in the past to make informed design decisions. One notable simulation is a Matlab acceleration simulation, written in 2016. This simulation was accurate enough to make high level architecture decisions, but is not fine enough nor properly structured to be used to test control algorithms. The high-level structure is shown in Figure 2. It worked well for design decisions, but the fundamental reason why it cannot be used to tune controllers is that there is no separation of the control system and the physical system. It was necessary to develop a new, real-time, simulation that accurately represented the signals sent and received by the vehicle control unit. It was also important that this new simulation have a versatile structure that would make it easy to add refining functions as more assumptions are addressed.

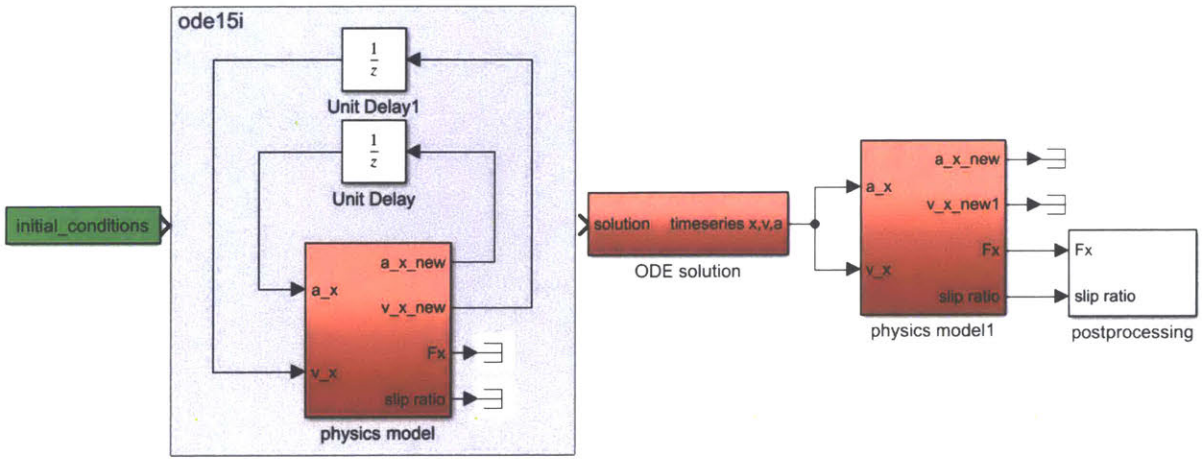

Figure 2: Block diagram of previously used acceleration simulation, written in Matlab in 2016.

2. DESIGN OF SIMULATION

2.1 COORDINATE SYSTEM

The coordinate system used in this simulation is SAE standard (Figure 3). Positive X is front, positive Y is right, and positive Z is down. Roll, pitch, and yaw are rotational movements about the X, Y, and Z axes respectively. The origin was selected to be at the intersection of the ground plane, the X-Z midplane of the car, and the plane intersecting the rear axle.

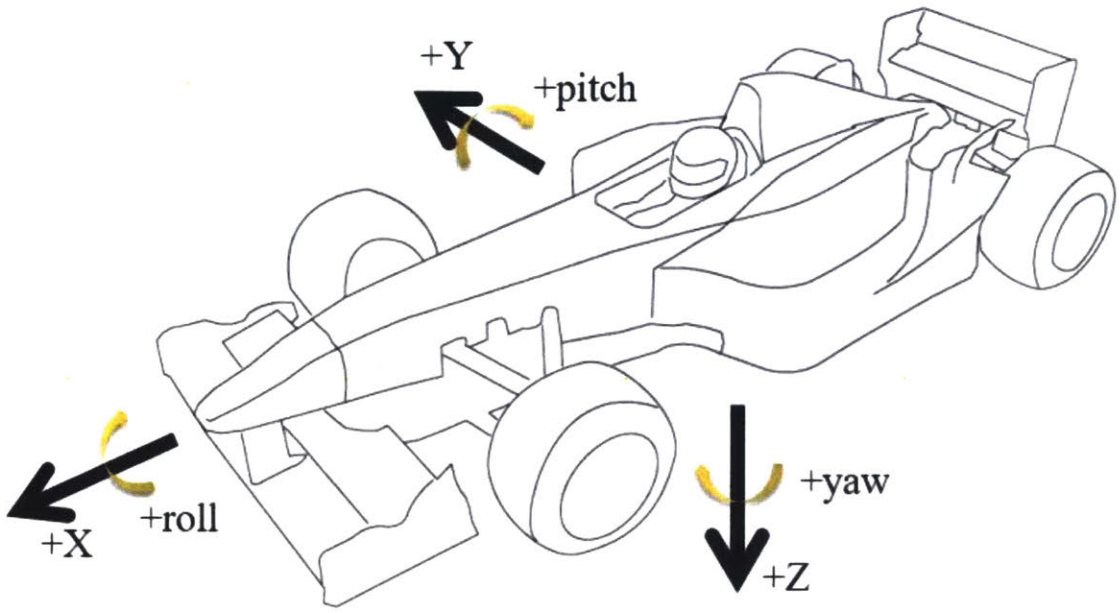

Figure 3: SAE Standard coordinate system.

2.2 ASSUMPTIONS

The assumptions made in this specific iteration of the simulation are as follows:

1. Straight line movement only
2. Single mass weight transfer model, i.e. the sprung mass, which is the portion of the car supported by the springs, is much heavier than the unsprung mass.
3. Driver is not being a feedback controller
4. No tire pressure effects
5. Tires are operating in optimal temperature range
6. No pitch effects on aero
7. The ground is flat

2.3 HIGH LEVEL STRUCTURE

The simulation is written in Simulink and Matlab. The major blocks that make up the simulation are the driver block, software block, and vehicle block (Figure 4).

The driver is a pure output block. The driver's only communication to the vehicle control unit (VCU) is through the steering wheel, the throttle pedal, and the brake pedal. These are passed to the VCU as a number between 0 to 1.

The software block represents the VCU, which computes the torque command. The inputs are sensor readings and the driver commands. The output is four torque commands to the wheels.

For a rear wheel drive car, the front torque commands are set to zero. Any torque command modifiers are contained in this block, including controllers and limits.

The physical system then responds to the torque commands, and the response is sent back to the VCU as sensor readings. The vehicle block contains the physics model and time step increment.

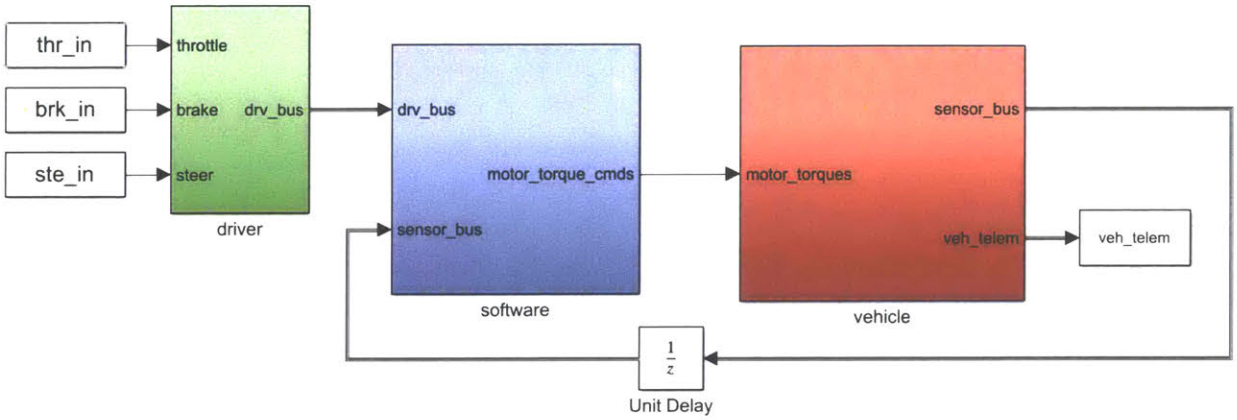

Figure 4: High level block diagram, showing the three main systems and main feedback loop.

2.4 SUBSYSTEM MODELING

The simplest model for a vehicle has a point mass m experiencing an acceleration a caused by the sum of external forces acting on the system. For this simplistic system, the external forces consist of the friction forces outputted by the tire. One tire's friction force is equal to its coefficient of friction mu multiplied by the normal force. The acceleration is easily computed from the friction forces, in the following equations.

$$F_{front} = \mu N_{front} \tag{1}$$

$$F_{rear} = \mu N_{rear} \tag{2}$$

$$a = \frac{1}{m}(F_{front} + F_{rear}) \tag{3}$$

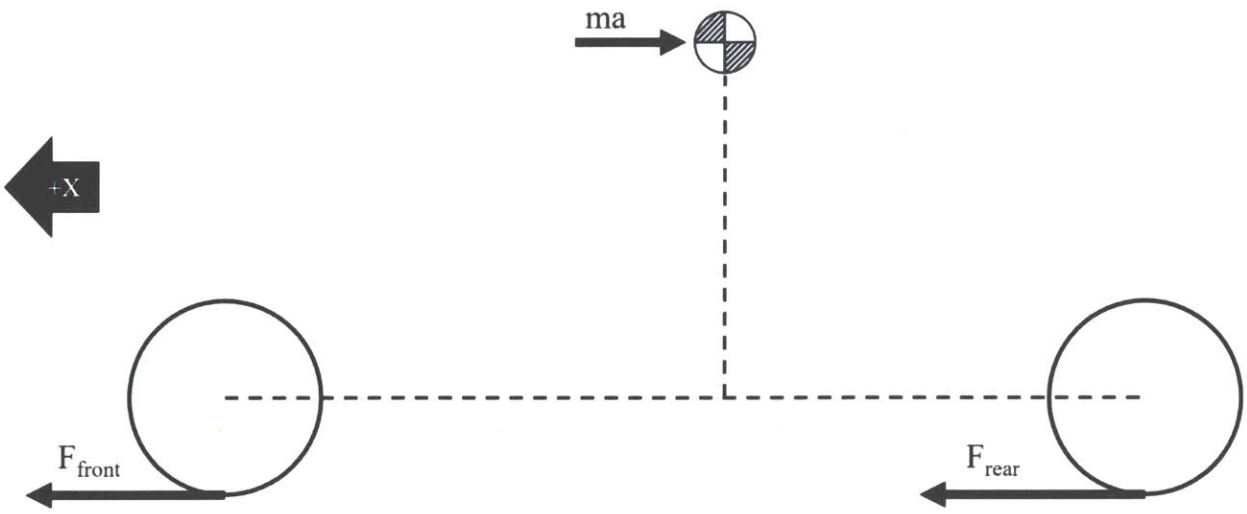

Figure 5: The free body diagram for the simplest vehicle model.

2.4.1 TIRES

The first and most important refinement to the simple model is the tire characteristics. Typically in vehicle dynamics, a tire is modeled using a semi-empirical function. The goal of this function is to describe the forces and moments output by the tire given a set of operating conditions. These forces and moments are described at the center of the contact patch, which is the flat spot where the tire touches the ground. A frequently used semi-empirical model is the Pacejka Magic Formula [1]. In this model, a large number of coefficients is fitted to a set of force and moment data gathered from a purpose-built tire testing machine. In the 2006 edition of this model, the tractive forces and moments depend on the normal load, slip angle, slip ratio, inclination angle, pressure, and temperature. There is much digging that can be done into the specifics of tire modeling, but for the purposes of a full vehicle simulation, the tire can be treated as a black box described by the following function.

$$\left(F_x\,,\,F_y\,,\,M_z\right) = fn(N\,,\,\kappa\,,\,\alpha\,,\,\gamma,\,P,\,T) \tag{4}$$

$$where$$
$$N = normal\ load$$
$$\kappa = slip\ ratio$$
$$\alpha = slip\ angle$$
$$\gamma = inclination\ angle$$
$$P = pressure$$
$$T = temperature$$

The parameters with greatest effect on longitudinal motion are the normal load and slip ratio. The most unexpected property of the tire is that the coefficient of friction, mu, is not constant

(Figure 6). It actually decreases with increasing normal force. This is referred to as "saturation" or "load sensitivity". This phenomenon stems from the construction and chemistry of the tire rubberitself, which vehicle dynamicists want nothing to do with.

But understanding tire saturation is essential for a good high-level view of vehicle design. Because of the decreasing coefficient of friction, the shape of the friction force vs normal force curve is concave down. This means that the total friction force available for any pair of tires is maximized when the normal forces are equal. This is illustrated in Figure 7. This fact drives much of the car's design. For instance, the entire reason shock absorbers exist is to quickly smooth out oscillations in normal force, and to return the car to a state where the driven tires have as equal a load distribution as possible.

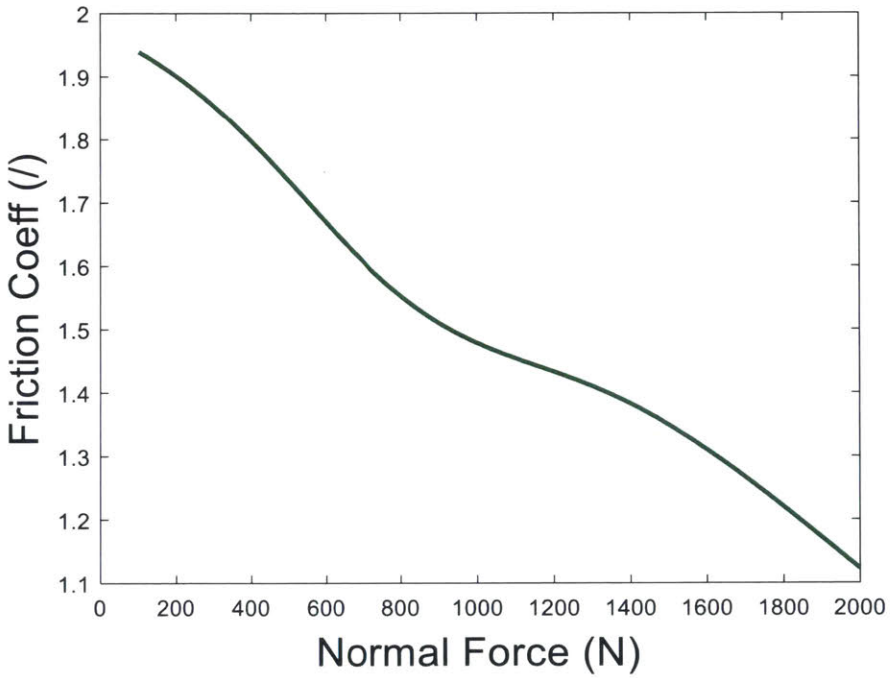

Figure 6: The dependence of the coefficient of friction on normal load. This plot was made with a slip ratio of 0.05. The tire used was the 18x6in Hoosier.

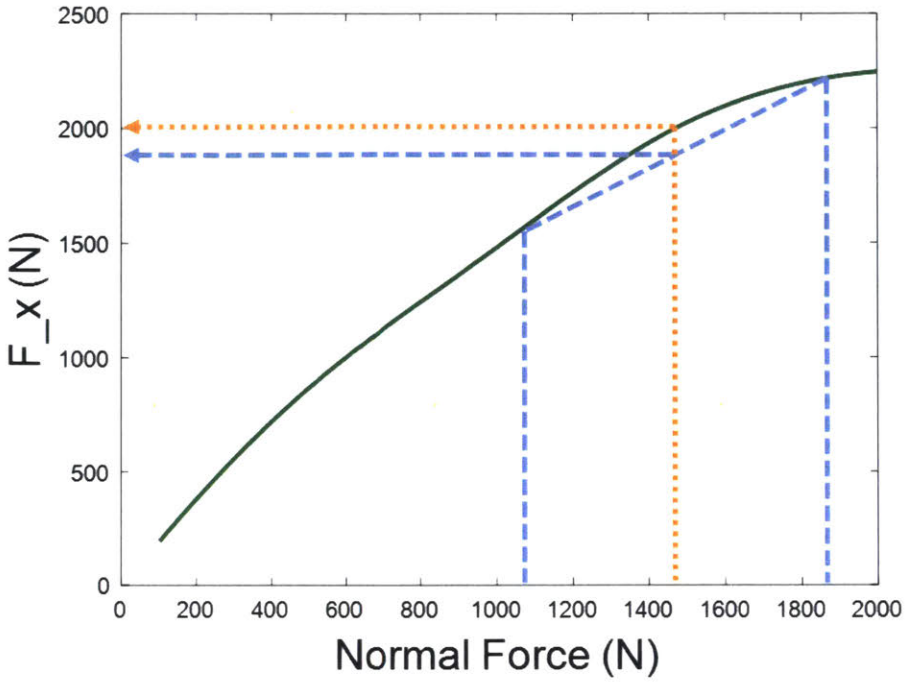

Figure 7: Longitudinal friction force vs normal force. The average friction force available for a pair of tires occurs when the tires are evenly loaded. Blue shows the average friction force for an unevenly loaded pair. Red shows a higher friction force from an evenly loaded pair.

Slip ratio describes the difference between the rotational speed of the tire and the linear speed of the tire. Intuitively, it is related to the deformation of the tire sidewall as torque is applied to the hub. An extreme example slip ratio on a dragster rear tire is shown in Figure 8 for better visualization.

Figure 8: Dragsters' tire wrinkles are an example of extremely high slip ratio.

This deformation of the sidewall, and the contact patch itself, is what allows the tire to output a friction force that launches the vehicle forward, or stops the vehicle in the braking case. Slip ratio is defined as follows.

$$\kappa = \begin{cases} \dfrac{R\Omega - v}{|v|}, & |\mathrm{v}| > \mathrm{v}_{\mathrm{th}} \\ \dfrac{2(\mathrm{R}\Omega - \mathrm{v})}{\mathrm{v}_{\mathrm{th}} + \dfrac{\mathrm{v}^2}{\mathrm{v}_{\mathrm{th}}}}, & |\mathrm{v}| \leq \mathrm{v}_{\mathrm{th}} \end{cases} \tag{5}$$

$\Omega =$ wheel rotational speed

$v = wheel\ center\ speed$

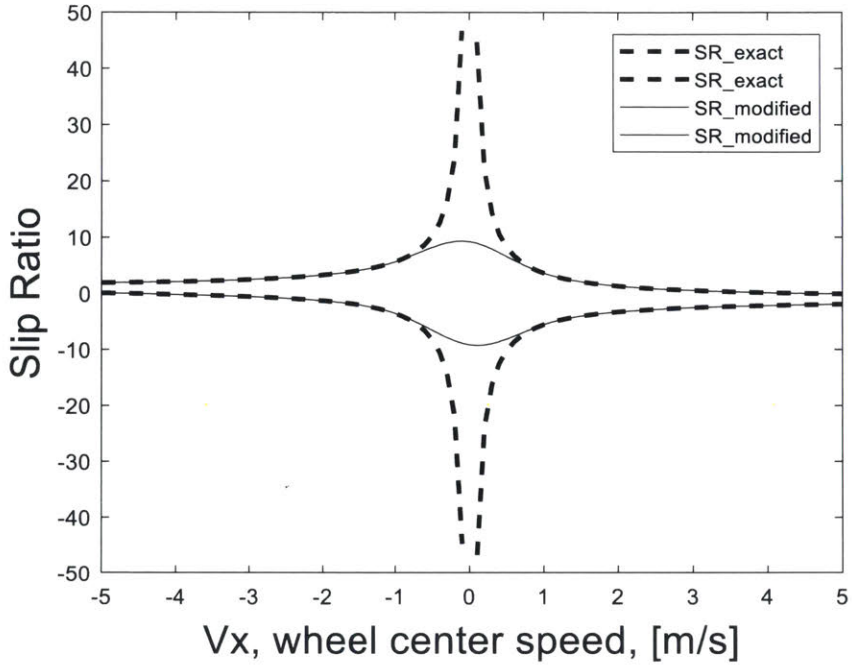

Figure 9: Illustration of how the low-speed slip ratio function differs from the exact slip ratio.

In order to prevent slip ratio from becoming infinite at low speeds, a modified function is used below a threshold speed, which allows slip ratio to be continuous as the linear speed crosses zero [2]. The effects of the modification are illustrated in Figure 9, where the modified function is plotted against the exact, discontinuous function.

In this simulation, slip ratio must be calculated from the rotational dynamics of the tire since it cannot be deduced from driver inputs alone. The inputs to the slip ratio calculation are angular velocity and linear velocity of the wheel center. In the case of longitudinal motion, the linear speed of all four wheels is equal to the x speed of the vehicle CG. The angular velocity of each wheel is obtained from the rotational dynamics of the tire. The sum of torques is the sum of the previous time step's friction force and the torque input from the motor. The angular acceleration is then integrated to obtain the angular velocity, which can finally be input into the slip ratio equation. The equation of motion is as follows, referencing the free body diagram in Figure 10.

$$T_{in} - F_x R = J\alpha \tag{6}$$

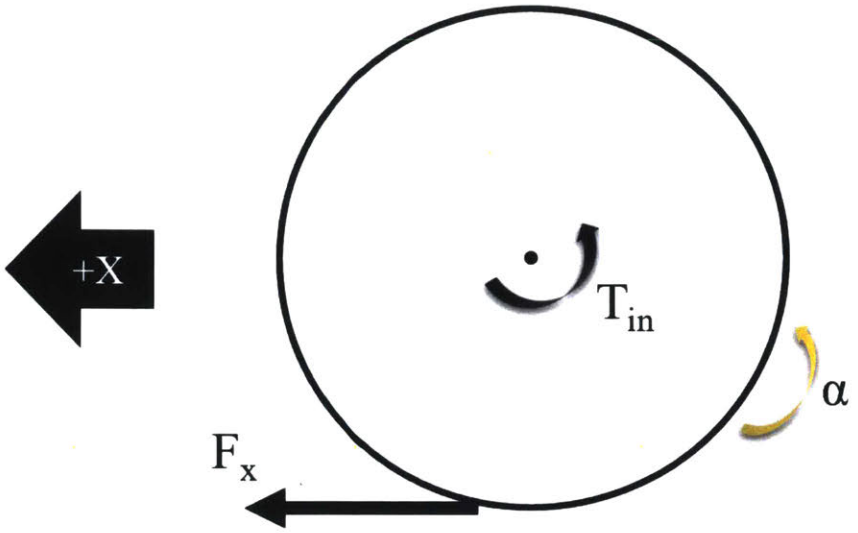

Figure 10: Free body diagram of tire rotational dynamics.

The semi-empirical model is stored in a lookup table with inputs slip ratio and normal load, and output Fx, friction force in the X direction. The table values were created by fitting the Pacejka 2006 model coefficients to a set of tire data provided by the FSAE Tire Test Consortium [3]. The fitting process is outside the scope of this paper. The resulting lookup table values are plotted in Figure 11. The breakpoint ranges were chosen to be the range of the test data available, and the extrapolation method holds the final table value constant for queries outside the breakpoint range.

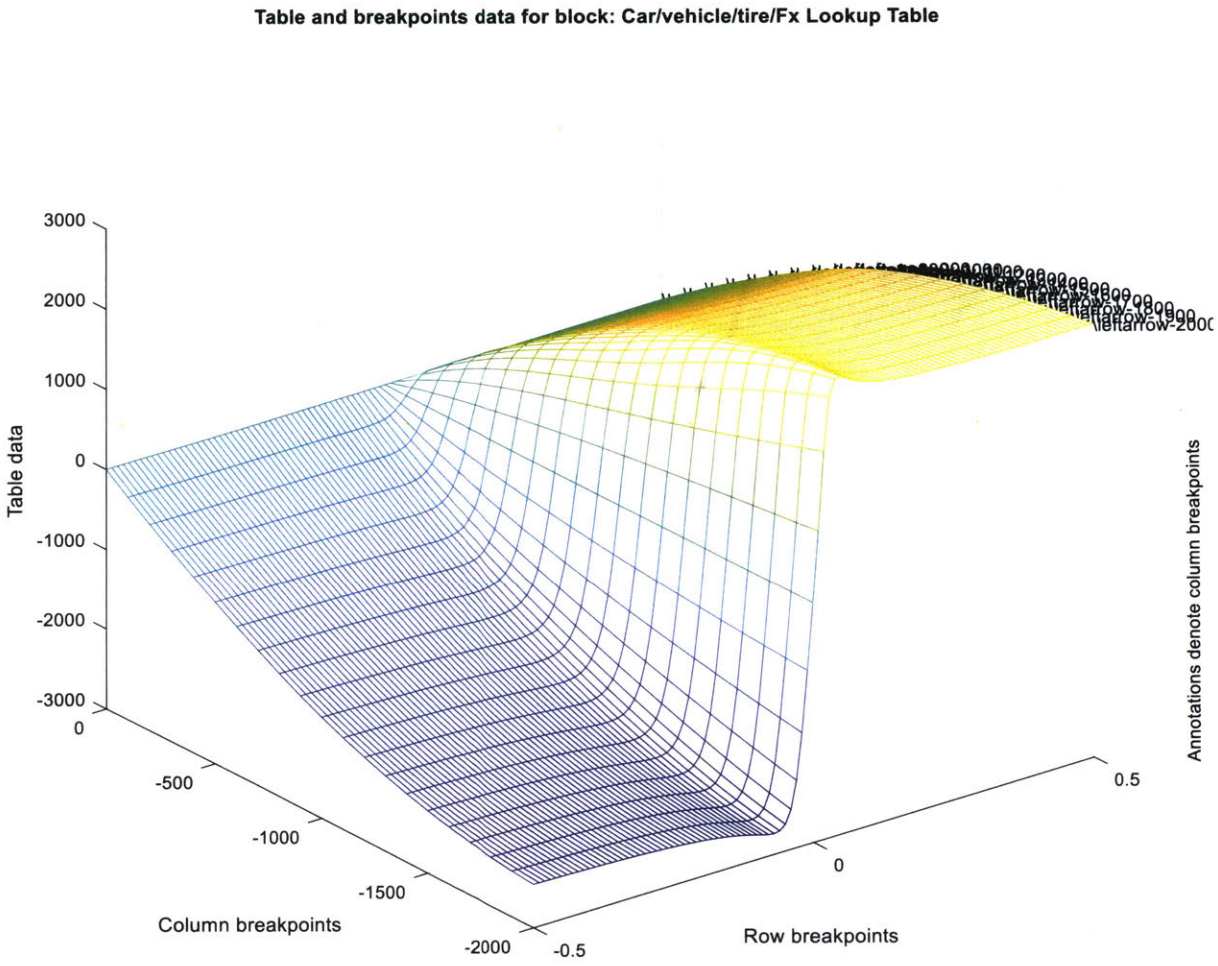

Figure 11: The lookup table stored in the simulation. The row breakpoints are slip angles, the column breakpoints are normal loads. The table data is Fx, longitudinal friction force.

Another layer of complexity that can be added to the tire model is rolling resistance. A constant rolling resistance coefficient model is used, where the rolling resistance force is directly proportional to the normal force on the wheel [4]. The coefficient is calculated as in the following equations. The velocity input is only needed to smooth the transition as the wheel changes direction. The resulting rolling resistance force is applied at the wheel center and so does not affect the tire rotational dynamics. It is subtracted from the final calculation of the force that the tire applies to the chassis.

$$F_{rr} = NC_{rr} \tag{7}$$

$$C_{rr} = C_{rr0}\, tanh\left(\frac{4v}{v_{th}}\right) \tag{8}$$

$$where$$
$$N = normal\ load$$
$$v = wheel\ center\ velocity$$
$$v_{th} = threshhold\ velocity = 1$$

2.4.2 NORMAL FORCE FUNCTIONS

Because the tire friction force output is so sensitive to normal load, it is imperative that it is computed accurately. The main contributions to normal load on the four tires are static load, inertial load, and aero load, assuming straight line motion only.

Static load is simply what one would read from a set of car scales. This can be calculated from the total vehicle mass and the location of the center of gravity.

Inertial load is the change in normal loads if the vehicle is accelerating. From D'Alembert's principle, the X acceleration on the vehicle results in an inertial force equal to the product of the car's mass and X acceleration. From the free body diagram in Figure 12, we can find the change in normal force at the front tires due to the body acceleration. The change in normal force at the rear tires must be equal and opposite to that of the front tires in order to maintain equilibrium in the Z direction.

The third contribution is aero load. The aero package on the vehicle usually consists of a front wing, a rear wing, and an optional diffuser. Each element provides a lift and drag force, both acting at the element's center of pressure. However, the aero's effect on the entire car can be simplified greatly by adding these together and treating them as one unit. The aero effect can be reduced to two forces and one moment: a lift force ("downforce" in racing terms), a drag force, and a torque. Thus, its effect in the XZ plane is fully described. We can arbitrarily choose the point at which the forces act; here they are described as acting at the origin (rear contact patch) in order to simplify the resulting calculation of the wheel loads. The three aero coefficients required to describe the contribution are C_LA, C_DA, and $C_{Xm}A$, which were found by the aero team via flow simulations.

Finally, the three contributions are added to obtain the total normal force at the front and rear wheels. Note that these refer to the sum of each pair of wheels, and in actual implementation, they are divided by two to split into four individual tires.

$$N_{front} = mg\frac{b}{WB} - ma_{cg}\frac{H_{cg}}{WB} + \frac{T_{aero}}{WB} \tag{9}$$

$$N_{rear} = mg\frac{a}{WB} + ma_{cg}\frac{H_{cg}}{WB} + (F_{lift} - \frac{T_{aero}}{WB}) \tag{10}$$

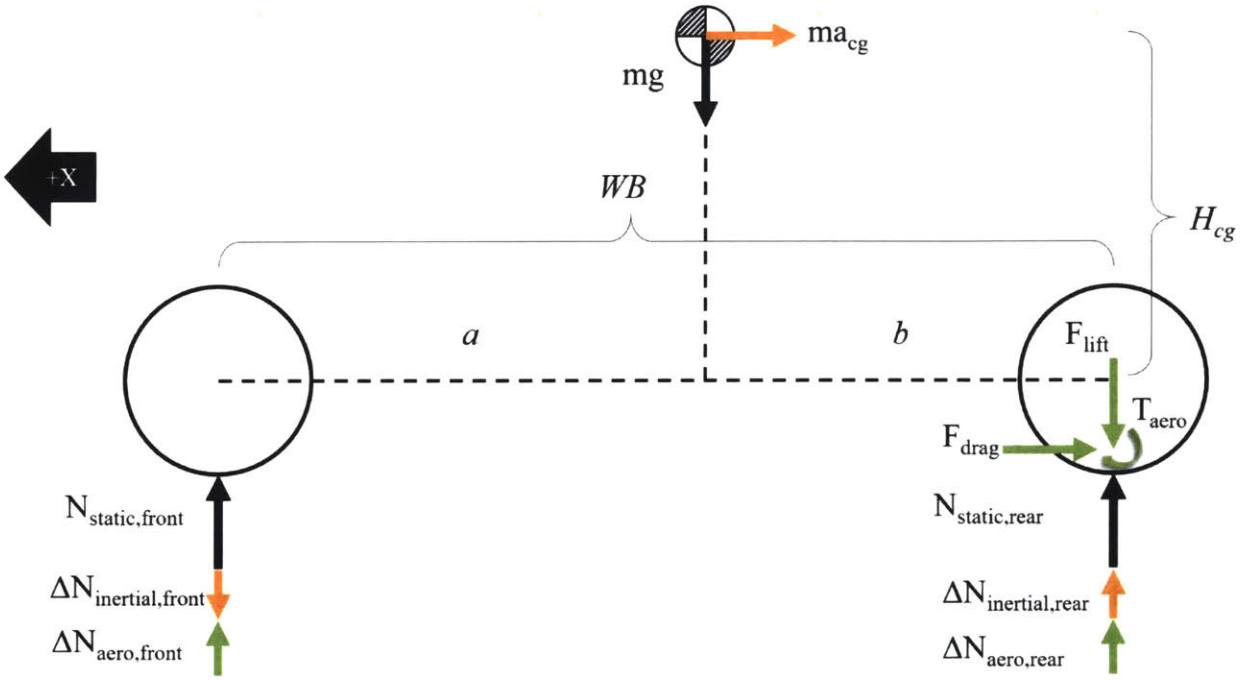

Figure 12: Free body diagram to find total normal force on front and rear wheels. The three contributions are static, inertial, and aero.

2.4.3 PITCH DYNAMICS

So far, we are assuming the normal force transfers instantaneously from the ground to the chassis, as if the car had rigid links instead of dampers. In reality, the dampers result in a delay before reaching the steady state value, and effectively smooth the large normal force peak at the beginning of the launch. Since we are only considering longitudinal motion at this point, we can use pitch stiffness and pitch damping, instead of modeling individual corner shocks, as a simplified way to incorporate dampers and springs into the model.

Consider a simplified vehicle model where the front and rear suspension is represented by a spring and dashpot. The pitch center, the point about which the vehicle pivots instantaneously, is located on the ground midway between the front and rear contact patches. This is due to suspension geometry alone; the control arms' inboard pickup points are parallel to the X axis. If the geometry incorporated anti-squat or anti-dive, the pitch center would be in a different location. The pitch stiffness and pitch damping can be found as follows, referencing the diagram in Figure 13. Note that the motion ratio, defined as the ratio of shock travel to wheel travel, is not taken into account, because it is very close to 1.

$$k_{pitch} = \frac{WB^2}{4}(k_f + k_r) \tag{11}$$

$$b_{pitch} = \frac{WB^2}{4}(b_f + b_r) \tag{12}$$

$$where$$
$$k_{f,r} = front, rear\ ride\ (spring) stiffness$$
$$b_{f,r} = front, rear\ damping\ coefficient$$

Because the pitch system is well-approximated as a linear second order system, we can write the transfer function from torque about the Y axis to the pitch angle. The torque T about the Y axis is equal to the body acceleration multiplied by the moment arm of the height of the CG.

$$\frac{\theta_{pitch}(s)}{T_y(s)} = \frac{1}{Js^2 + b_{pitch}s + k_{pitch}} \tag{13}$$

Because the instantaneous contribution from the springs is already included in the inertial load, we only need to calculate the contribution from the dampers, i.e. the transient contribution. Pitch angle is differentiated to pitch angular speed, put through a geometric transformation, then multiplied by the front or rear damping coefficient to obtain the transient contribution.

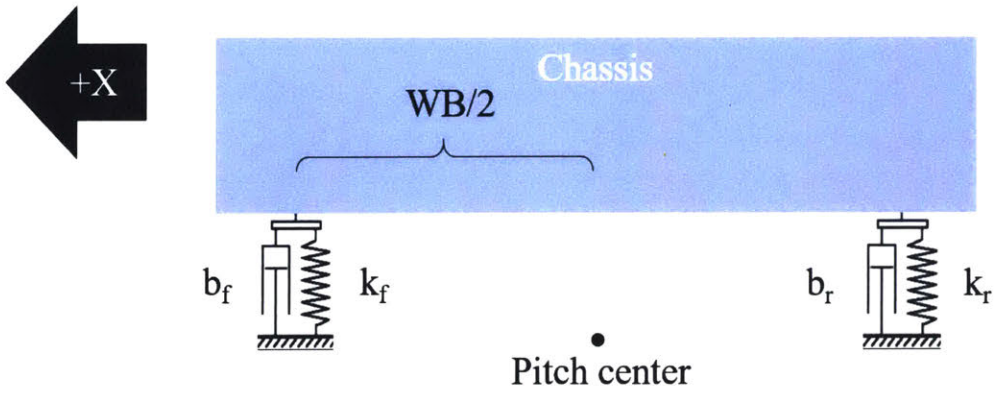

Figure 13: Simple model as a chassis supported by a spring-damper at each axle. The pitch center is midway between the two axles.

2.4.4 POWERTRAIN

The powertrain may further limit the car's performance if the motor cannot provide the requested torque, even if the tires have the tractive capability. The powertrain function block is modeled as a simple limiter. The torque speed curve for the chosen motor is stored in a lookup table. For each time step, the function looks up the motor velocity, equivalent to the wheel velocity divided by the gear ratio, to find the maximum torque the motor is able to produce, and caps the torque request. To calculate the power-speed curve, the torque-speed curve was simply multiplied by the speed. The power-speed curve for an Emrax 228 motor, which was used in the rear powertrain on the MY18 car, is shown in Figure 14a. This curve was extracted from current and speed measurements throughout the endurance event at the 2018 competition. Note that field weakening was used in this session, and is evident in the sloping torque dropoff at high speeds. The power-speed curve for a Hawk 40 motor, which is used in each front wheel on the MY19 car, is shown in Figure 14b. This curve was obtained from the manufacturer-provided datasheet, and verified with a stall torque and no-load speed test. The Hawk's power-speed profile assumes no field weakening.

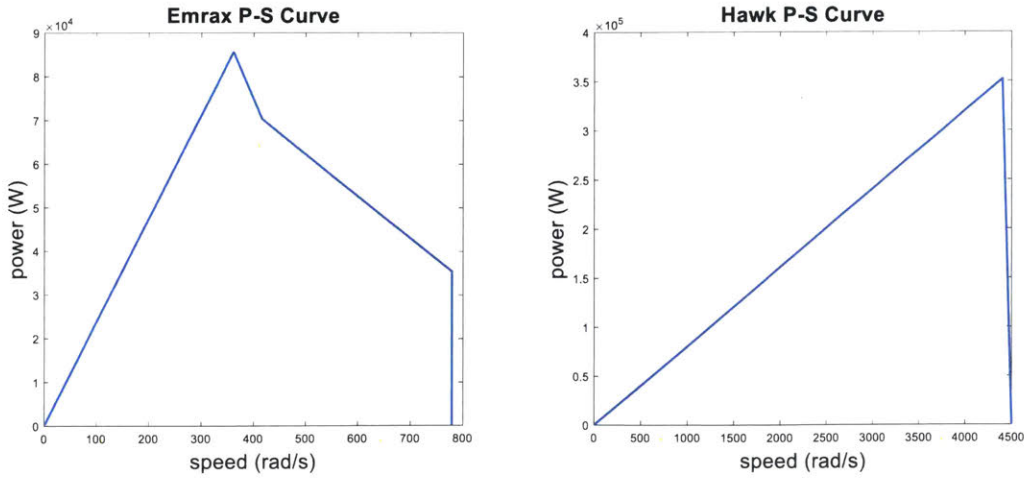

Figure 14: Power-speed curves for the two motors used in the simulation. The Emrax (a) is an axial flux surface permanent magnet motor, while the Hawk (b) is a radial flux SPM.

2.5 SIMULATION OUTPUTS

The output of the simulation is a collection of timeseries. The most essential information is acceleration, velocity, and position of the center of gravity. Power is also an important metric to look at when assessing performance. Plotting all the contributions to power together gives a good visual for what components are dominating the vehicle movement. Slip ratio is also an important metric to monitor as it is the one thing we can control.

An example of a typical acceleration run without any power or powertrain limits is shown in Figure 15. The effects of drag can be seen in the acceleration's steady decrease. Figure 16 shows the signals relevant to the tires. The effect of downforce can be seen clearly in the increasing average Fz.

Figure 17 and 18 show the results from a run with the powertrain limit. The effects are clearly seen at around 2 seconds when the acceleration drops off sharply, as the motor can no longer provide the requested torque.

To demonstrate the versatility of this simulation, an acceleration – braking run is plotted in Figure 19. A sinusoidal torque input is also plotted in Figure 20. Modeling these scenarios would not have been possible using the 2016 acceleration simulation.

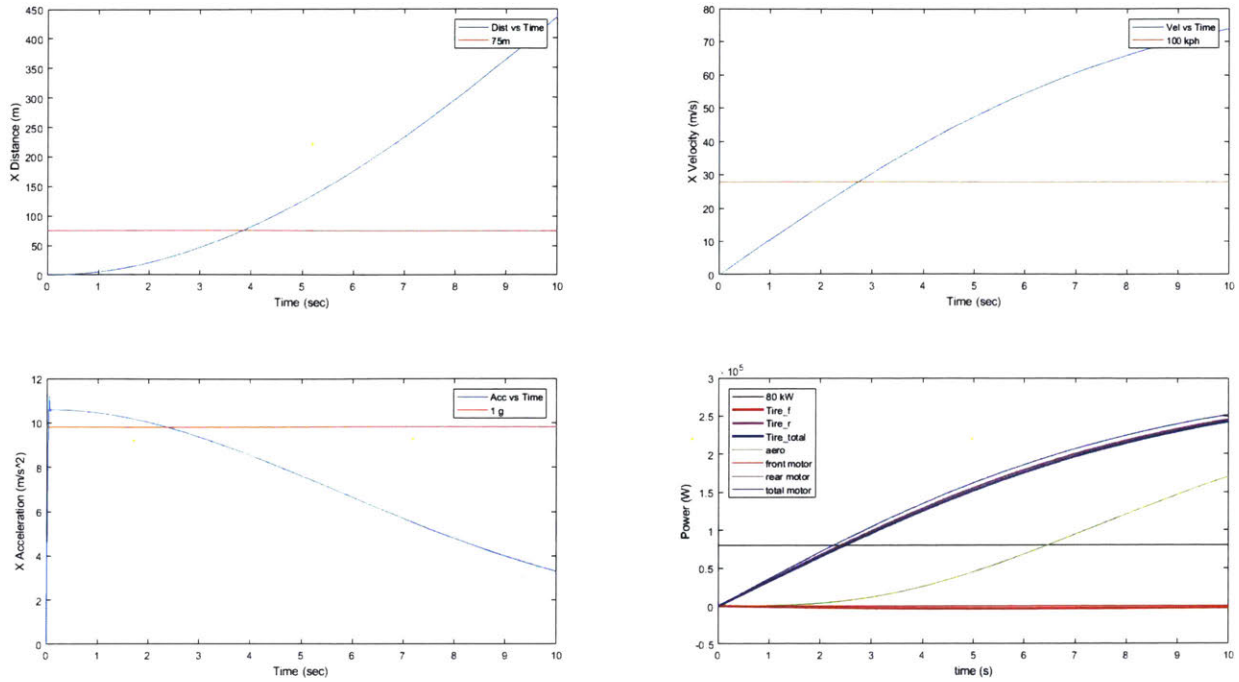

Figure 15: Sim output for launch with no powertrain limit. Driver input is a constant maximum torque request. Clockwise from top left: position, velocity, power, acceleration.

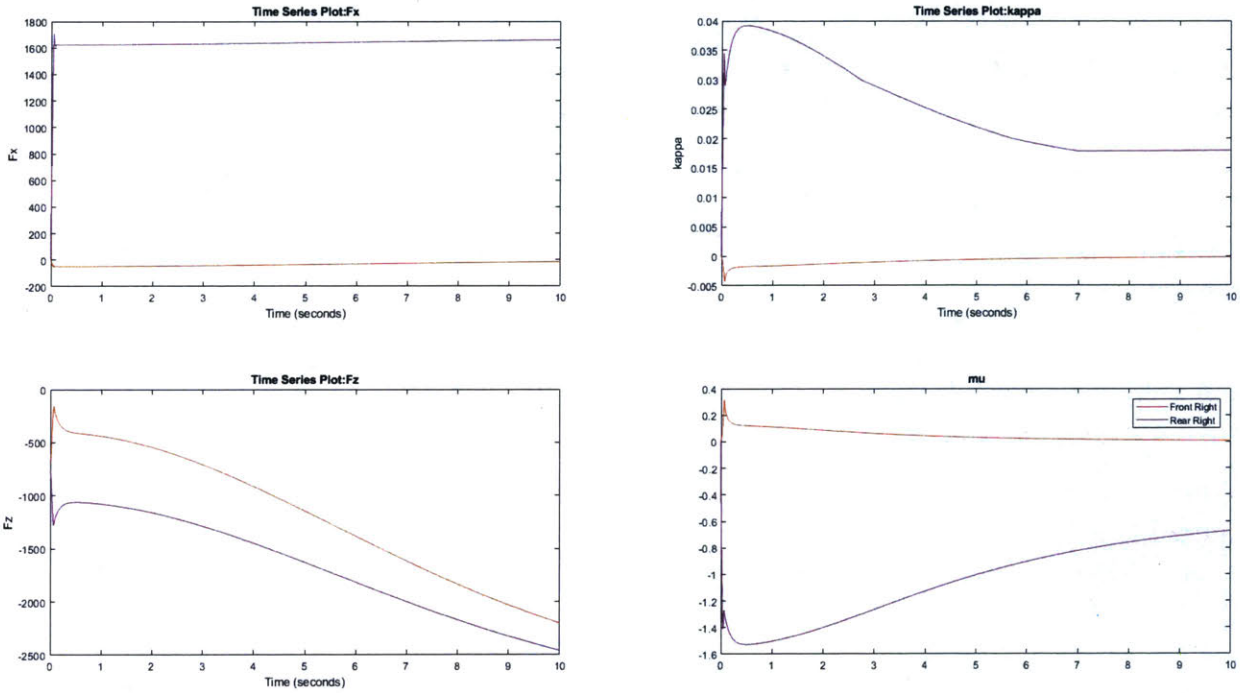

Figure 16: Tire telemetry output for launch with no powertrain limit. Driver input is a constant maximum torque request. Clockwise from top left: Fx longitudinal force, slip ratio, coefficient of friction mu, Fz normal force. Legend: red is one front wheel, purple is one rear wheel.

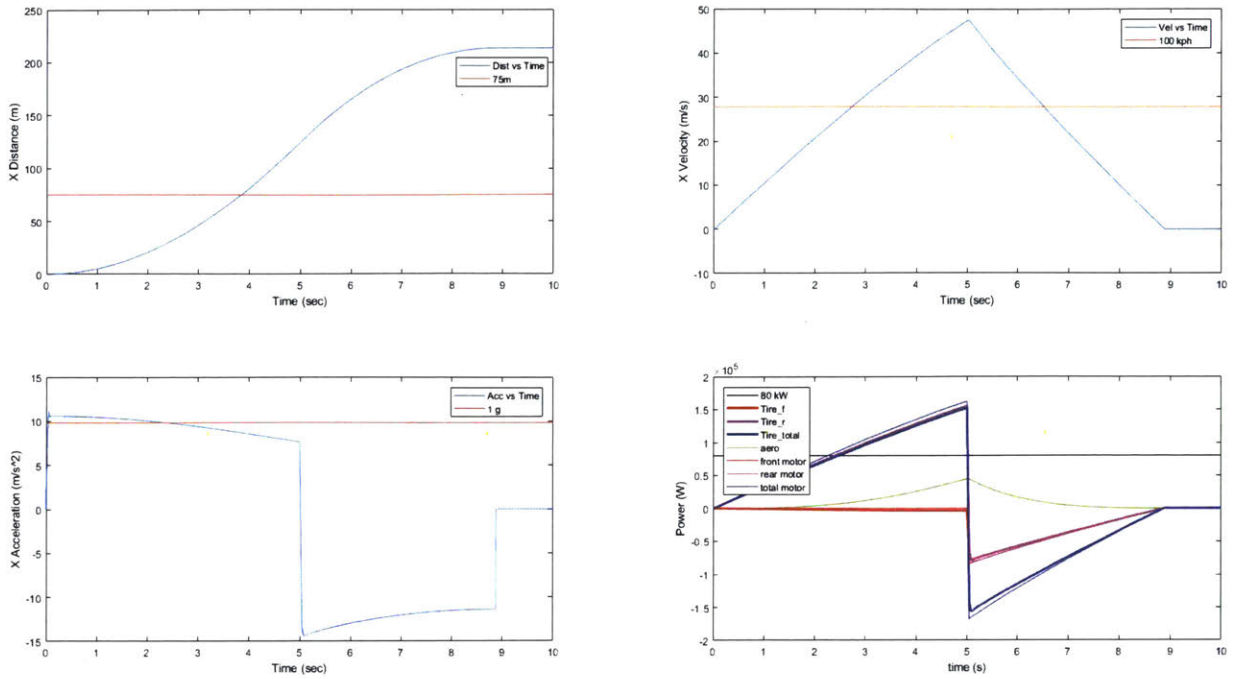

Figure 19: Sim output for an acceleration-braking run, without powertrain limit, to demonstrate versatility.

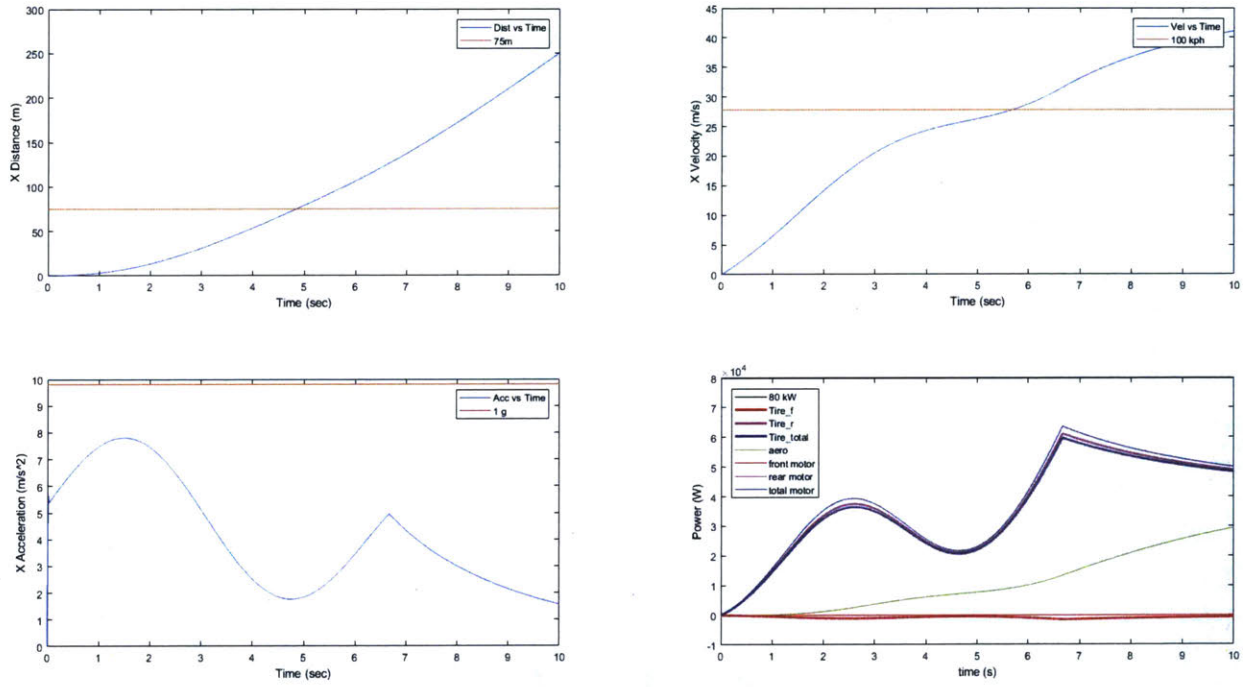

Figure 20: Sim output for a sinusoidal torque command, with powertrain limit.

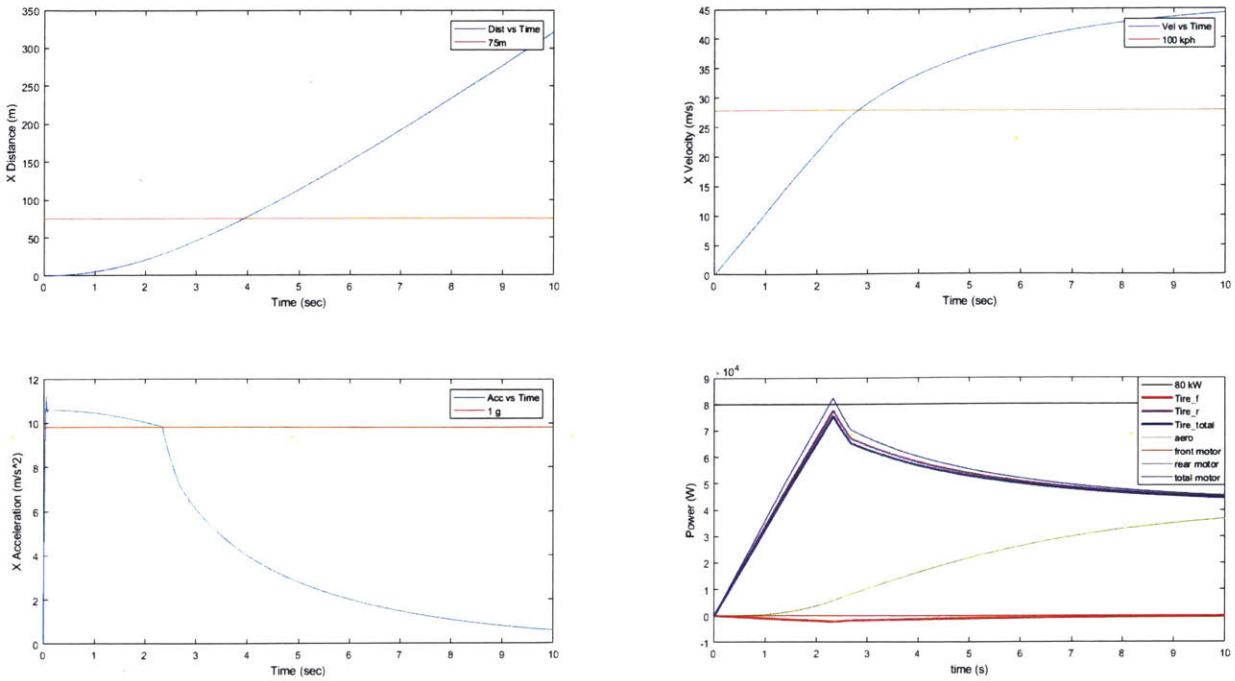

Figure 17: Sim output for launch with powertrain limit. Driver input is a constant maximum torque request.

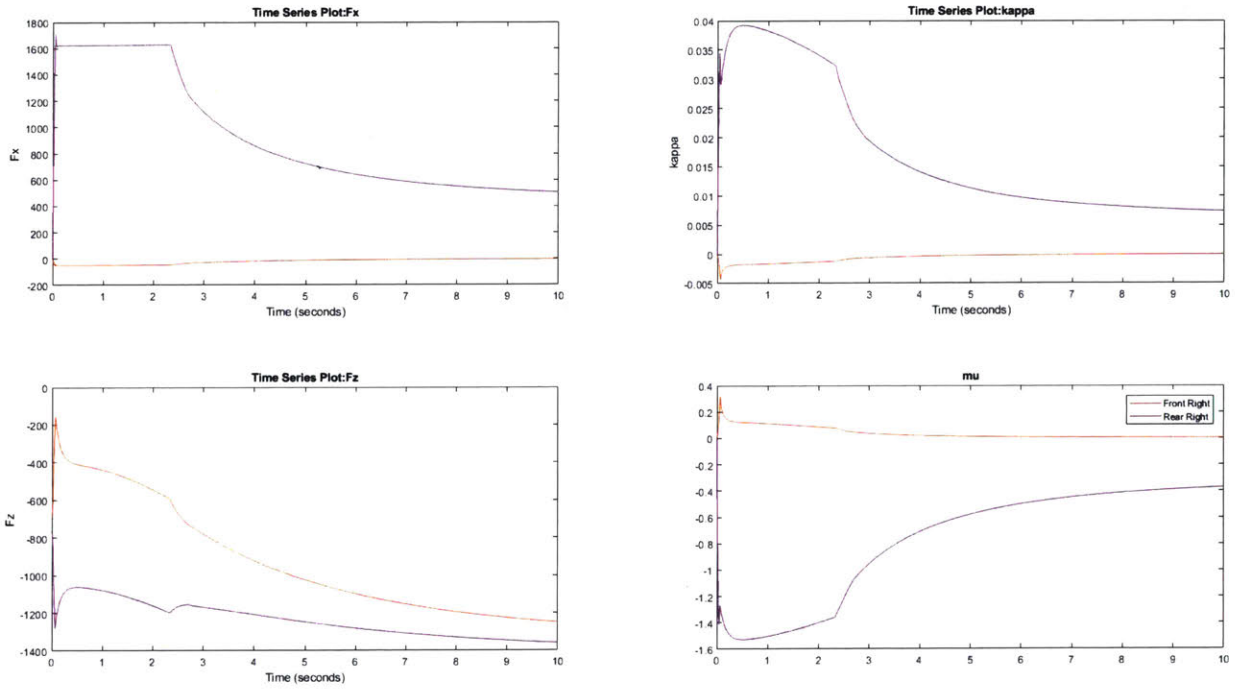

Figure 18: Tire telemetry for launch with powertrain limit. Driver input is a constant maximum torque request.

2.6 CORRELATION WITH TEST DATA

Telemetry from an actual acceleration run was used to verify the accuracy of the simulation. The data was taken from a run performed at Palmer Motorsports Park on September 9, 2018, with the MY18 car. This run was done at the end of the testing trip, so it is fair to assume the tires were operating at temperature. However it is important to note that the tires used on this trip had been used many times before, and had likely lost a lot of tractive capability. It is also important to note that the length of track the run was performed on is not completely flat.

First, the relevant metrics were computed. The car's position, velocity, and acceleration, as well as the slip ratio, are good indications of accuracy, and relatively easy to extract from the telemetry. Since the vehicle does not have all the sensors that the simulation has, some approximations were made to compensate. Front wheel angular speed was measured with a Hall effect sensor fixed to the wheel hub (Figure 21). Note that this sensor gives very poor readings at low speed, but is quite accurate at high speeds. Motor feedback speed from the Emrax's resolver was used for the rear wheel angular speed. Both signals was quite noisy, so a moving average filter with a window of 0.2s was implemented, introducing a small delay.

In order to compute slip ratio, the linear speed of the wheels are required. Although the car was instrumented with an inertial measurement unit (IMU), the signal from the IMU was extremely noisy and required an accurate dimensional measurement from the mounting point to the center of gravity of the car, which was not obtainable. Instead, we can assume that the front wheels had a slip ratio of zero during the run, linear wheel speed was approximated by multiplying the front wheel angular speed by the radius of the tire. In reality, there is a small negative slip ratio at the front tires due to rolling resistance, brake drag, and rotational inertia, but it is much smaller than the rear slip ratio. The front wheel angular speed was also used to approximate the speed of the entire car. The car speed was then integrated to find position, and differentiated to find acceleration, which needed to be filtered further.

To simulate the run as accurately as possible, the throttle pedal input was extracted and inputted directly into the simulation. Brake pressure data for both the front and rear hydraulic lines was also readily available, and inputted into the simulation as well. Finally, a mechanical efficiency factor of 0.8 was added between the powertrain and the wheels in order to better match the data.

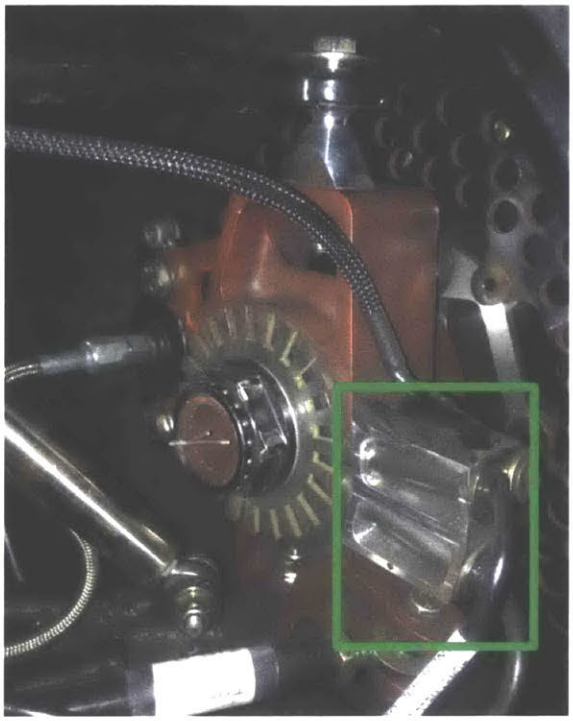

Figure 21: Front wheel speed sensor. The Hall effect sensor (green) reads the square toothed wheel, which rotates with the hub.

Figure 22 shows the simulation output along with the data. The first thing to note is that the shapes of the signals match extremely well. The powertrain limit is seen clearly in the acceleration and slip ratio data. The simulation is fairly accurate until about 2.8s, at which point the deviation in velocity is significant. Notice that the powertrain begins to limit at around 2.8s. This indicates the torque-speed curve is an overestimate, even though it was extracted from the large endurance event dataset. One possible explanation for the discrepancy is if the motor parameters for field weakening had been changed from the endurance event. Unfortunately this information is difficult to trace. Another possible explanation is inaccurate aero coefficients. Aero effects are proportional to the square of velocity, and the effects of an incorrect drag coefficient would be seen only at higher velocities.

Another notable discrepancy is the offset in the slip ratio data (Figure 23, top right). The method of using front wheel speed instead of true linear wheel speed actually underestimates the linear velocity, and overestimates the slip ratio, so this is not the cause of the offset. A more likely explanation is again the large discrepancy in the powertrain characteristics. A lower torque limit would result in the car not being able to produce high enough slip ratios to match the simulation. Similarly the aero inaccuracies would also result in this slip ratio discrepancy.

Despite this, the simulation can still be used confidently to learn about vehicle behavior and sensitivities because the accurate shapes of the signals indicates the different subsystems are interacting in the correct manner.

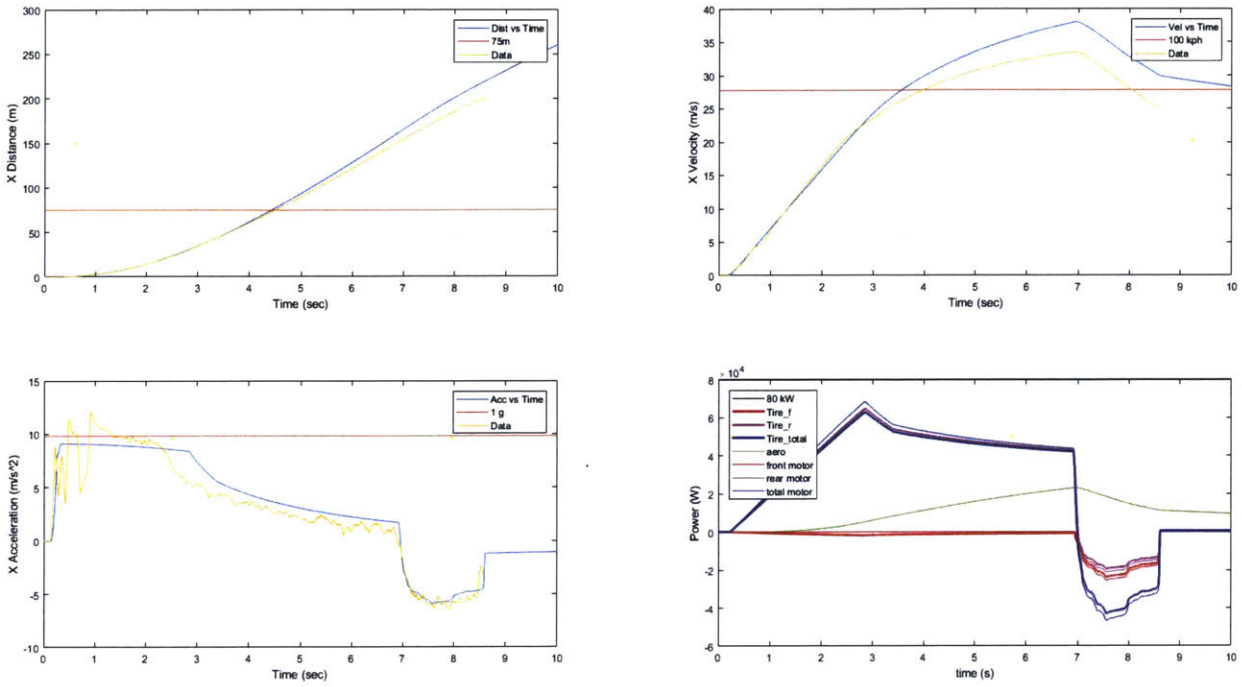

Figure 22: Sim output plotted with data (yellow) extracted from an accel run.

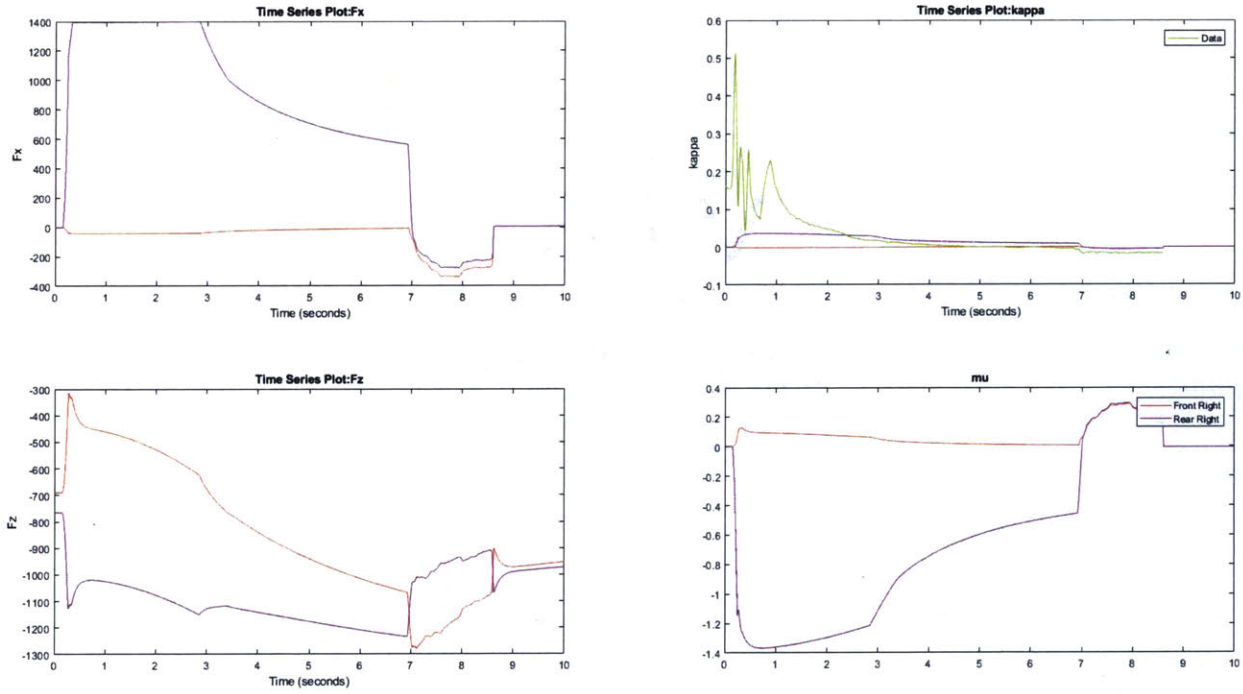

Figure 23: Tire output plotted with measured slip ratio (green) from an accel run.

3. INVESTIGATION OF LINEAR MOTION

3.1 IDEAL CONTROLLER

The second objective of the project is to use the simulation to learn about the vehicle's linear motion. In order to isolate the physical system from the control system, an "ideal" controller was developed. This controller measures the slip ratio and normal force, and uses the tire lookup table in order to find the maximum possible friction force at the current time step. Then it back-calculates the torque required to produce that force, and replaces the driver input with that torque. This controller is impossible to implement in actuality because we will never have perfect tire data, especially because it is so dependent on external conditions. However, it works well for showing us what the maximum possible performance of the vehicle is.

Figure 24 shows a typical output for the ideal controller in rear wheel drive, with a powertrain limit. For comparison, Figure 25 shows the output for a constant torque command. Note that the slip ratio is maintained at a higher value for longer than the human-controlled run. It is interesting to note that there is only 4.5 hundredths of time difference between the ideal and the constant controller. This seems to indicate there are diminishing returns for designing a launch controller.

The ideal controller is also useful for assessing a four wheel drive car's performance, which would usually be very dependent on control algorithms. Figure 26 shows the output from the ideal four wheel drive run, and Figure 27 shows the output from a constant torque command run. There is a time difference of 0.2s, which further emphasizes the importance of a well-designed algorithm to split the torque between the front and rear axles.

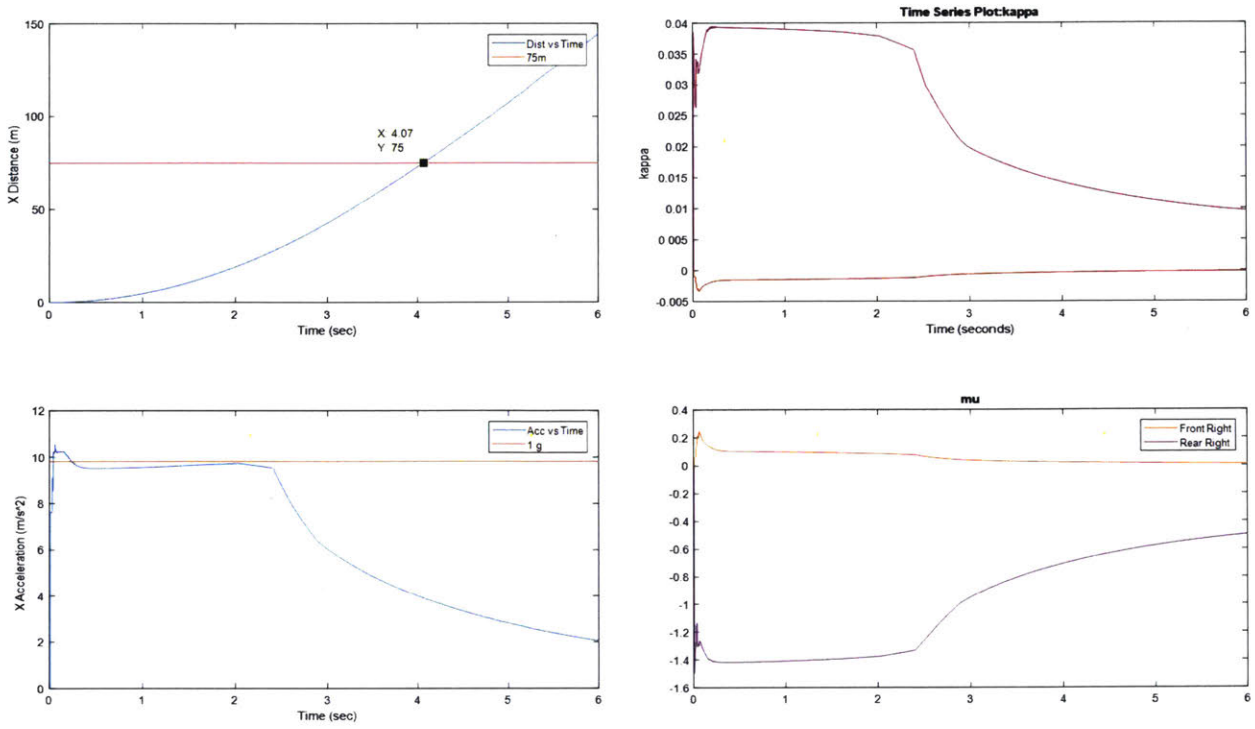

Figure 24: Ideal controller, rear wheel drive, with powertrain limit.

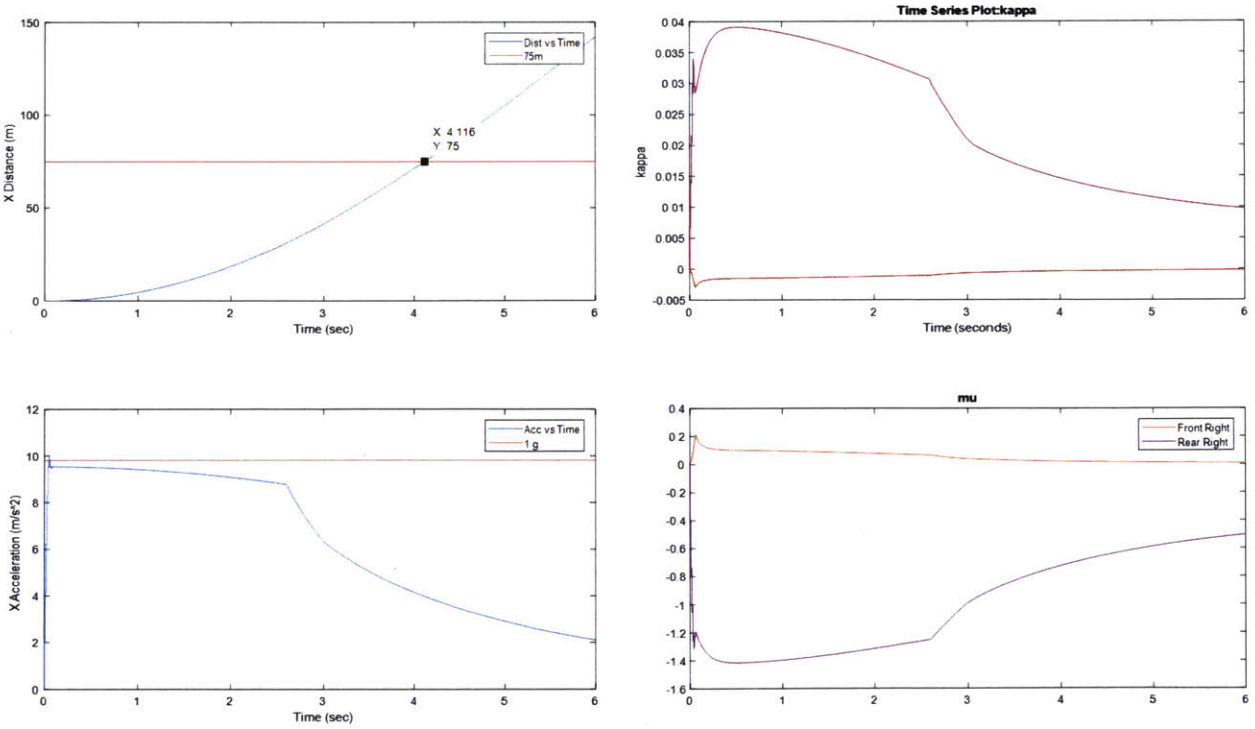

Figure 25: Constant torque command, rear wheel drive, with powertrain limit.

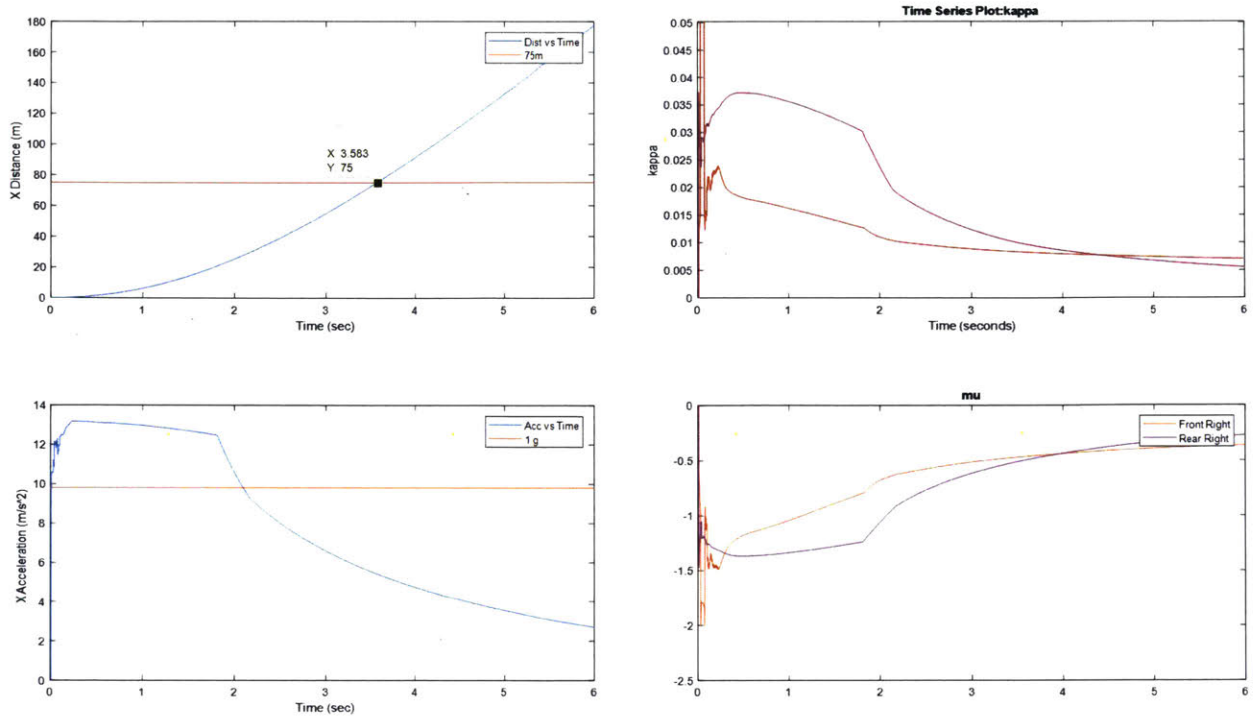

Figure 26: Ideal controller, four wheel drive, with powertrain limit. 0-75m time is 3.583s.

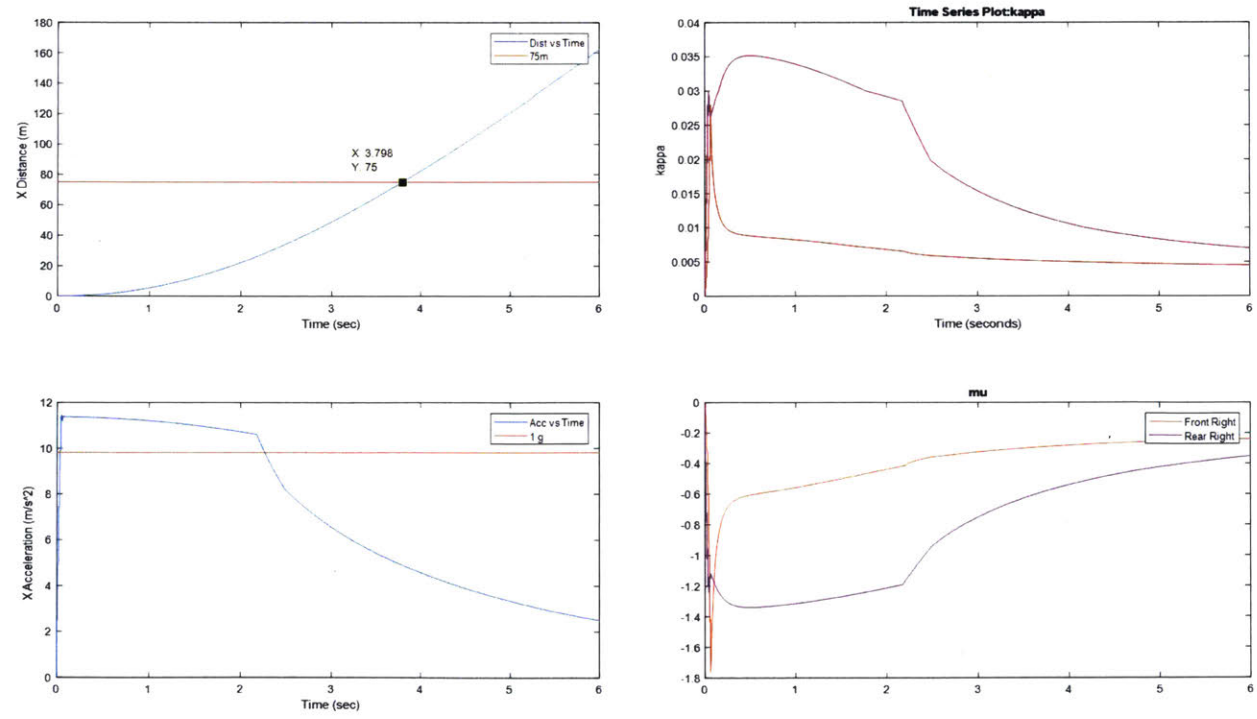

Figure 27: Constant torque command, four wheel drive, with powertrain limit. 0-75m time is 3.796s.

3.2 ROAD SURFACE COEFFICIENT

Perhaps the most important factor is one that is nearly impossible to measure. Within the tire model, there is a correction factor *lambda_mu*, which is put in place to account for differences between the surface on which the tire was tested, which is similar to 120 grit sandpaper, and the surface on which the tire is currently operating, which clearly does not provide as much traction as sandpaper. A rule of thumb, delivered directly from the people who oversee the tire test, that is commonly used for Formula SAE tires operating on a racetrack is lambda_mu = 2/3. This factor changes drastically depending on the surface roughness, how much dust is on the track, how much rubber is left on the track from previous cars, the temperature, the humidity, just to name a few. The tire's dependence on difficult-to-predict factors has driven the focus of the controls team to develop a controller that can accommodate a range of lambda_mu values. Figure 28 shows the dependence of 0-75m time on lambda_mu. This study was done on a rear wheel drive vehicle, powertrain-limited, with the ideal controller.

The 0-75m time changes by almost an entire second depending on the track surface quality. This makes sense, as an acceleration run on a cold day with old tires is noticeably slower than a run on a hot sunny day with brand new tires. It asymptotes abruptly because the motor begins to severely limit torque as the track surface is able to provide more traction. This study illustrates the extent of the challenge posed to the team. It is especially difficult for teams based in the northeast, like MIT, because oftentimes the only time we can run in conditions similar to hot, dry Nebraska concrete is the day before competition.

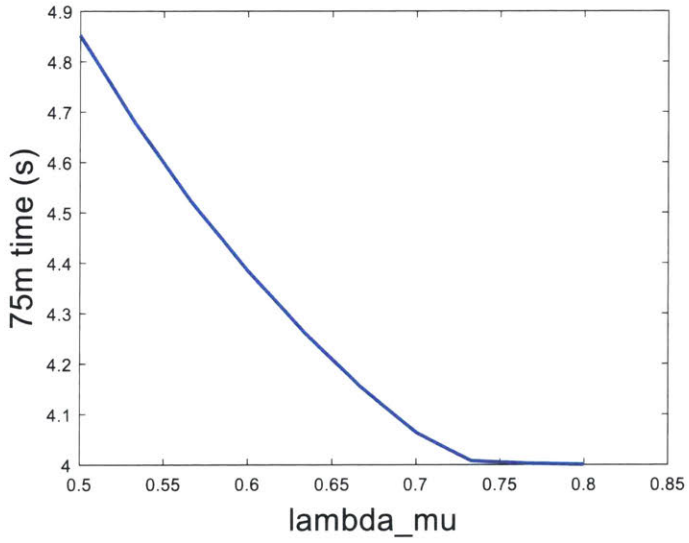

Figure 28: Dependence of 0-75m time on road surface coefficient.

4. CONTROLLERS

4.1 TRACTION LIMIT AND CONSTANT CONTROLLER

The simulation was used to conduct a study of the traction limit. This was motivated by the need to either predict or detect when the tire has lost traction, without knowing the road surface coefficient. Two lambda_mu values were chosen to represent the possible range of values on any given dry day. For each lambda_mu value, the simulation was run in rear wheel drive mode with a constant torque command multiplied by a factor from 0 to 1. This factor was swept through a narrow range. Then, the slip ratio timeseries was plotted to find the maximum torque command that did not cause the tire to slip. 0-75m time was found and plotted for the sweep. The results are summarized in Table 2.

The result for the high lambda_mu, 0.6, is displayed in Figure 29. It is obvious which runs saw a loss of traction, as the slip ratio rises to a large number. The loss of traction is also evident in the 0-75m times plotted against the torque factor in Figure 30. As soon as there is wheelspin, the times suffer by about 0.1s, which is significant. This is one of the reasons why wheelspin, or loss of traction, is not desirable. Furthermore, although the effect is not detectable in the straight line simulation, a high slip ratio results in a significant decrease in lateral tractive capability, which makes it easier for the car to become unstable and "spin out", even if there is no steering wheel input.

The results for the low lambda_mu value shows similar behavior (Figures 31, 32). However, the maximum torque before wheelspin differs greatly for the two lambda_mu values, as does the maximum slip ratio before wheelspin. This proves, unfortunately, that there is no magic measurable number that indicates the edge of the traction limit, and that there is no one torque command that will result in a reasonable 0-75m time for a range of lambda_mu values.

It is not a viable launch control scheme to use a hardcoded constant torque command. It is difficult to choose the torque command value, since it varies so much with different road surface coefficients.

Metric	**Lambda_mu = 0.6**	**Lambda_mu = 0.475**
Best constant torque 0-75m time	4.152 s	4.7 s
Best constant torque factor	90.4 %	68.2 %
Maximum peak slip ratio before wheelspin	0.081	0.071

Table 2: Results of traction limit study.

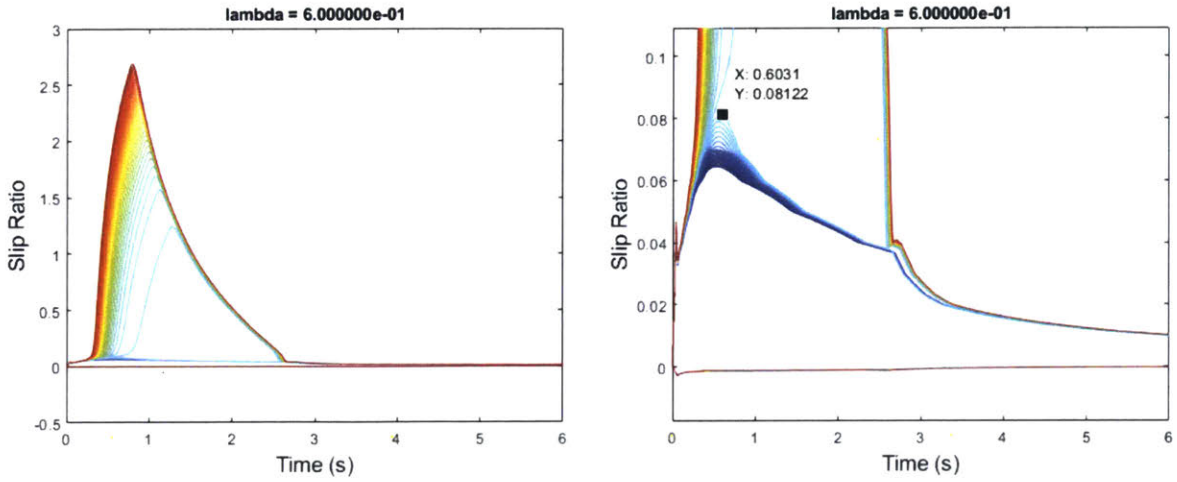

Figure 29: Slip ratio vs time plot for lambda_mu = 0.6 (high). The torque factor was swept from 89.5% to 92%. Right plot is a zoomed in view of the slip point.

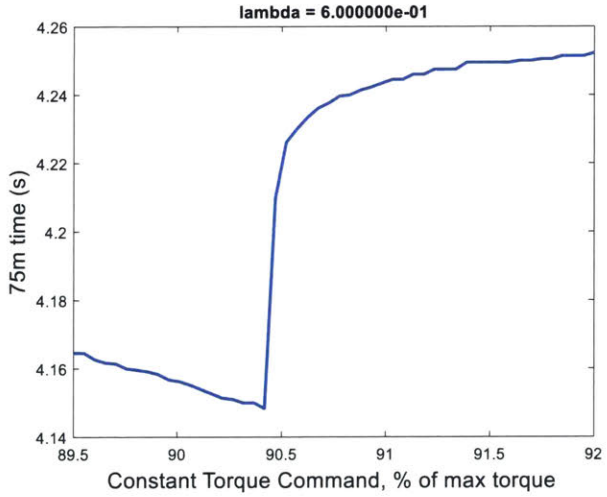

Figure 30: Dependence of 0-75m time on torque factor, for lambda_mu = 0.6. The global minimum occurs at 90.4%.

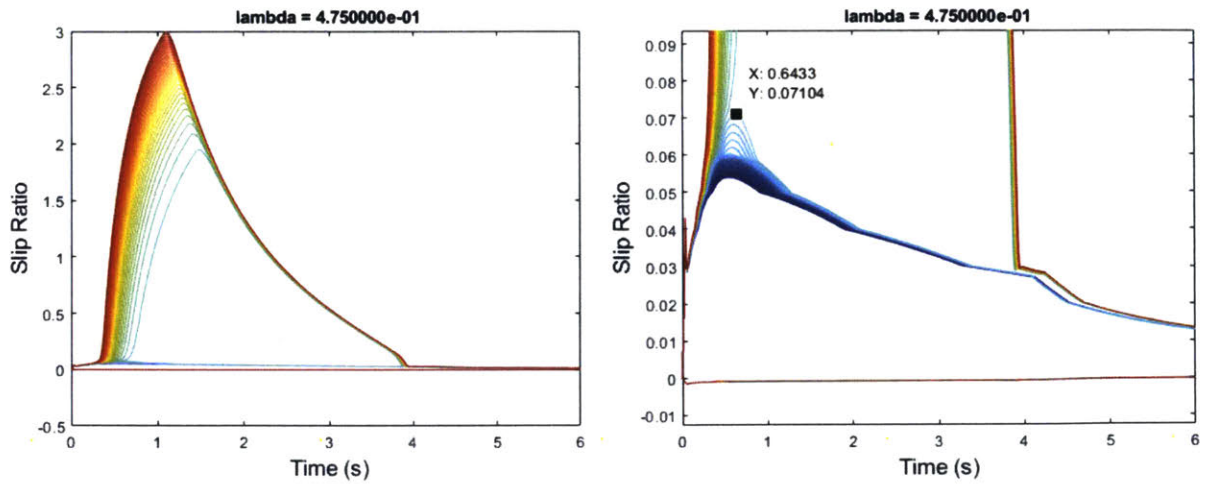

Figure 31: Slip ratio vs time plot for lambda_mu = 0.475 (low). The torque factor was swept from 67.6% to 69%. Right plot is a zoomed in view of the slip point.

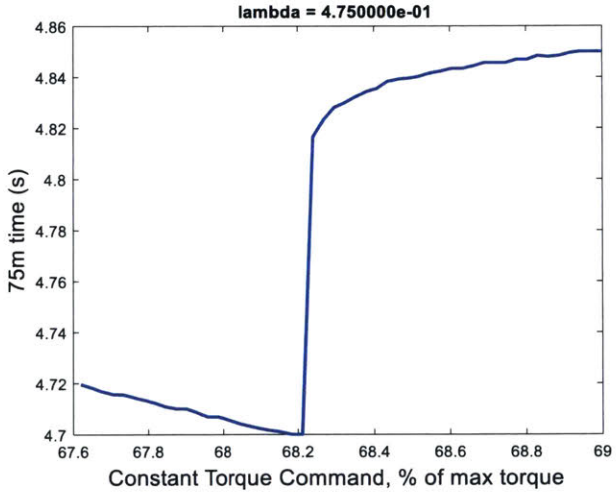

Figure 32: Dependence of 0-75m time on torque factor, for lambda_mu = 0.475. The global minimum occurs at 68.2%.

4.2 RAMPED CONTROLLER

A ramping controller was tested after the constant controller. The motivation came from an intuition that it is easiest to lose traction in the first few milliseconds of the run. A lambda_mu value of 0.6 was chosen for this study because the vehicle is traction limited in this condition. The initial ramp time, illustrated in Figure 33, was swept from 0.1s to 2s. The simulation was run in rear wheel drive mode, with powertrain limit.

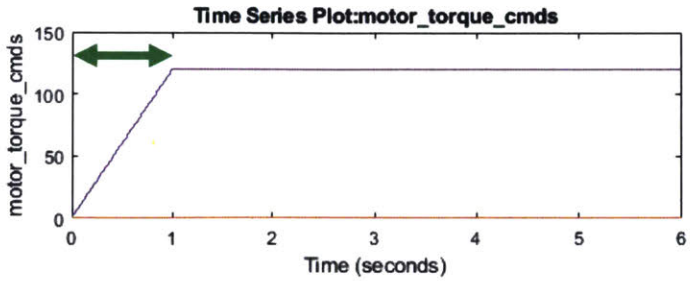

Figure 33: Definition of ramp time is seconds needed to reach full torque command (green).

The results are plotted in Figure 34. Counterintuitively, increasing ramp time did not prevent wheelspin; rather it just delayed it. From these results, the ramped controller is not a viable launch control scheme. However, a simple test can be conducted on the vehicle to verify these results, because the inverter on MY18 has a built in ramp function.

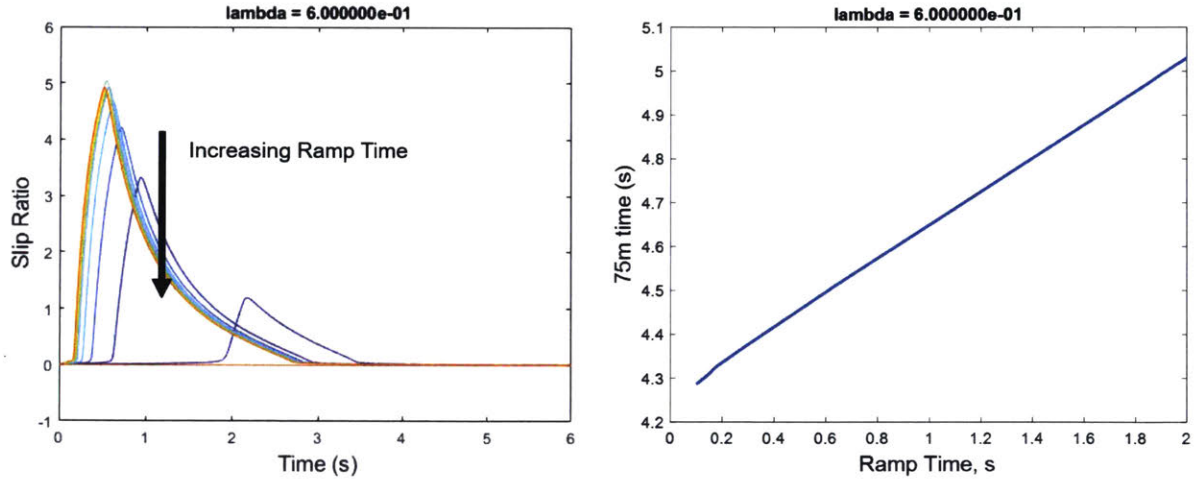

Figure 34: Results of ramp controller study, for lambda_mu = 0.6.

4.3 FOUR-WHEEL DRIVE LAUNCH CONTROLLER

A basic launch control scheme was developed for the four wheel drive configuration. As with the rear-wheel drive car, the simplest scheme is to command constant torque throughout the run. However, this method is a particularly poor choice for four-wheel drive, for several reasons. First, it is slow, as discussed in section 2.7.1. Second, it compromises stability. A constant split between the front and rear axles does not react to changes in normal load, and thus does not react to a potential decrease in tractive capability of the lightly loaded tire (front tires in a launch scenario). This makes it very easy for the driver to over-command the lightly loaded tires and lose stability in that axle. The driver would have to command very conservatively to prevent loss of traction at either axle.

The next step up from a constant torque command is to make it proportional to the normal force on its tire. This is feasible on the car because compression load cells can be installed on the

pushrods to directly measure normal load. This easily takes into account the changes in normal load. Essentially this controller is using a tire model that assumes constant coefficient of friction. Obviously this is not true, but the controller may be good enough even with this assumption.

To implement the controller, the normal loads are fed back to the software block, then multiplied by a gain, referred to as *Fz torque gain*, to transform it into a torque command. The gain is negative due to the positive Z axis pointing down; a higher absolute value of Fz torque gain is a more aggressive command. An example of the behavior of this controller is shown in Figures 35 and 36. This simulation was run in four-wheel drive mode with powertrain limit, lambda_mu = 0.6, Fz torque gain = -0.35.

A study was then conducted to preliminarily assess the viability of this method in various conditions. A range of lambda_mu was swept through. Because we are only assessing the viability of the proportional method, a range of gains was also swept through. Currently there is no good way of choosing the best gain for any given lambda_mu, so instead, we can simply look at a range.

The results are shown in Figure 37. The color plot indicates 0-75m time. The red line marks the edge of traction; the 0-75m times to the bottom and left of the line level off because wheelspin occurs. The definition of wheelspin for a run is if any value of slip ratio is greater than 1. From this preliminary study, this proportional method seems promising. It can produce a blistering 0-75m time of 3.62s at a conservative lambda_mu value of 0.55, which is already lower than the FSAE rule of thumb of 2/3. It can produce stable results even at low values of lambda_mu. This simple scheme could also improve braking drastically. Normal hydraulic braking systems have a brake bias bar that provides a constant ratio of front braking force to rear braking force. With four-wheel drive and regenerative braking, the brake bias can be dynamic and react to changes in normal load. Provided the motors are able to provide all the requested negative torque, the dynamic brake bias can significantly increase braking performance. The remaining problem is to figure out how to choose the gain value, which again returns to the problem of lambda_mu.

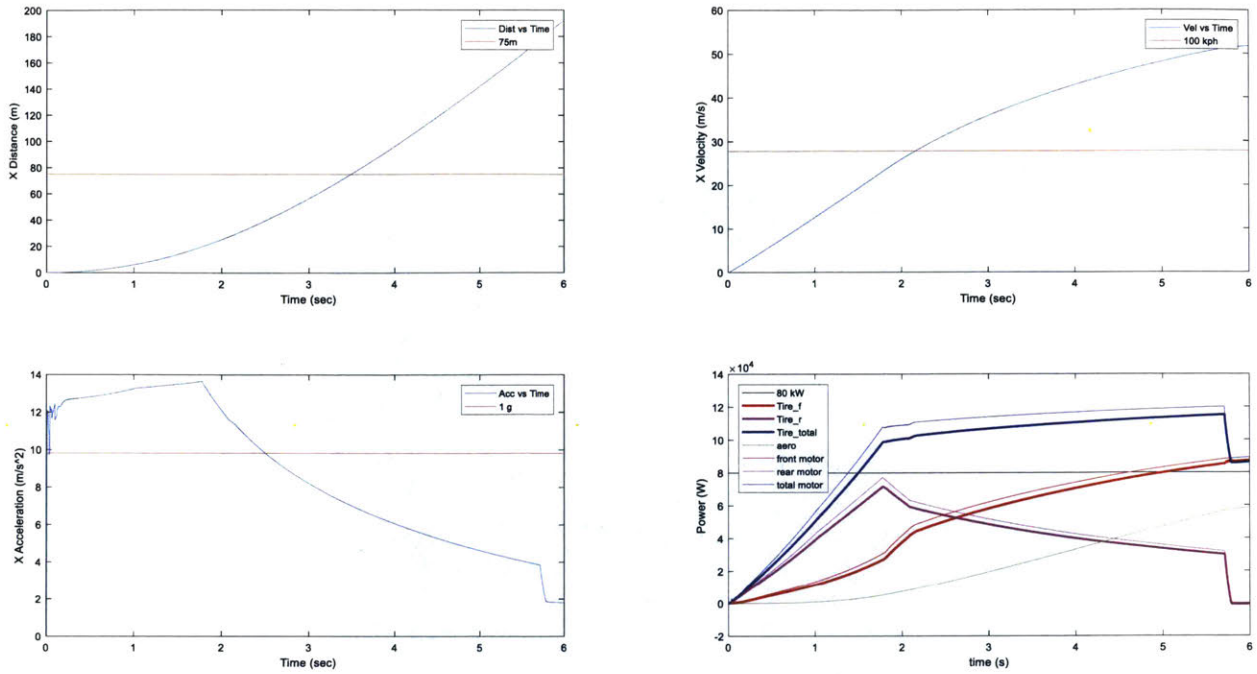

Figure 35: Sim output for the normal load proportional controller. Clockwise from top left: position, velocity, power, acceleration.

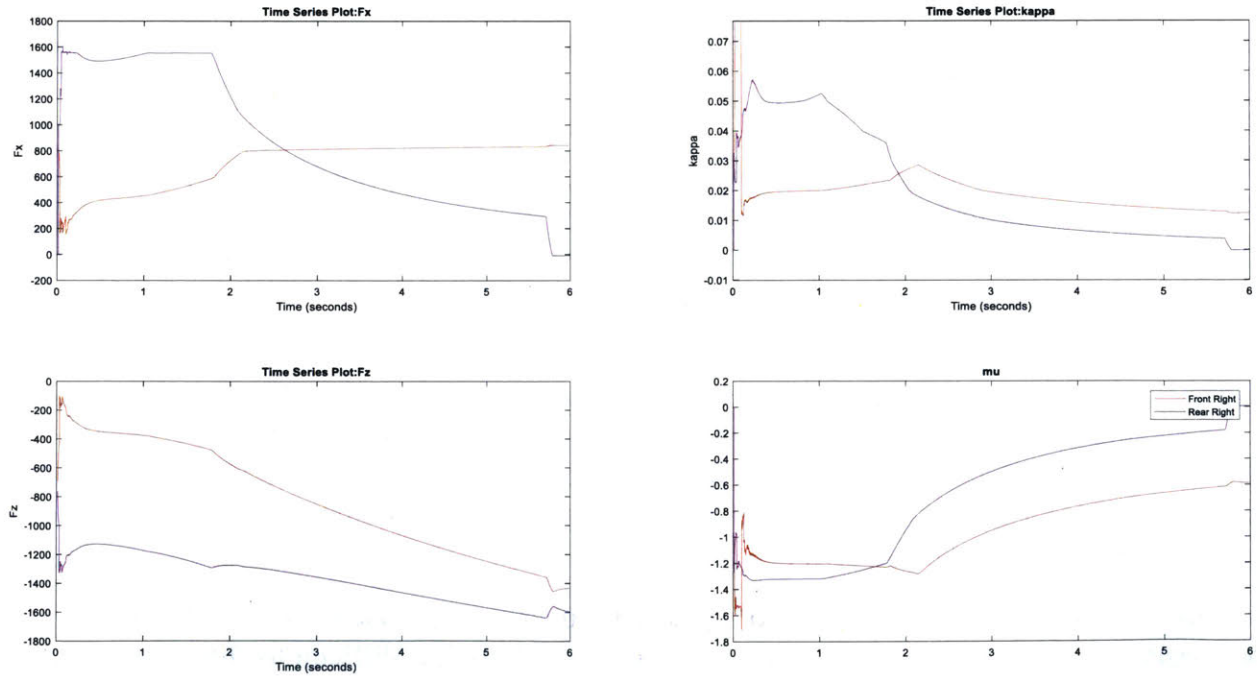

Figure 36: Tire output for the normal load proportional controller. Note the bizarre slip ratio output, a combination of powertrain limit, normal load changes, and wheel inertia. Clockwise from top left: Fx, slip ratio, coefficient of friction, Fz.

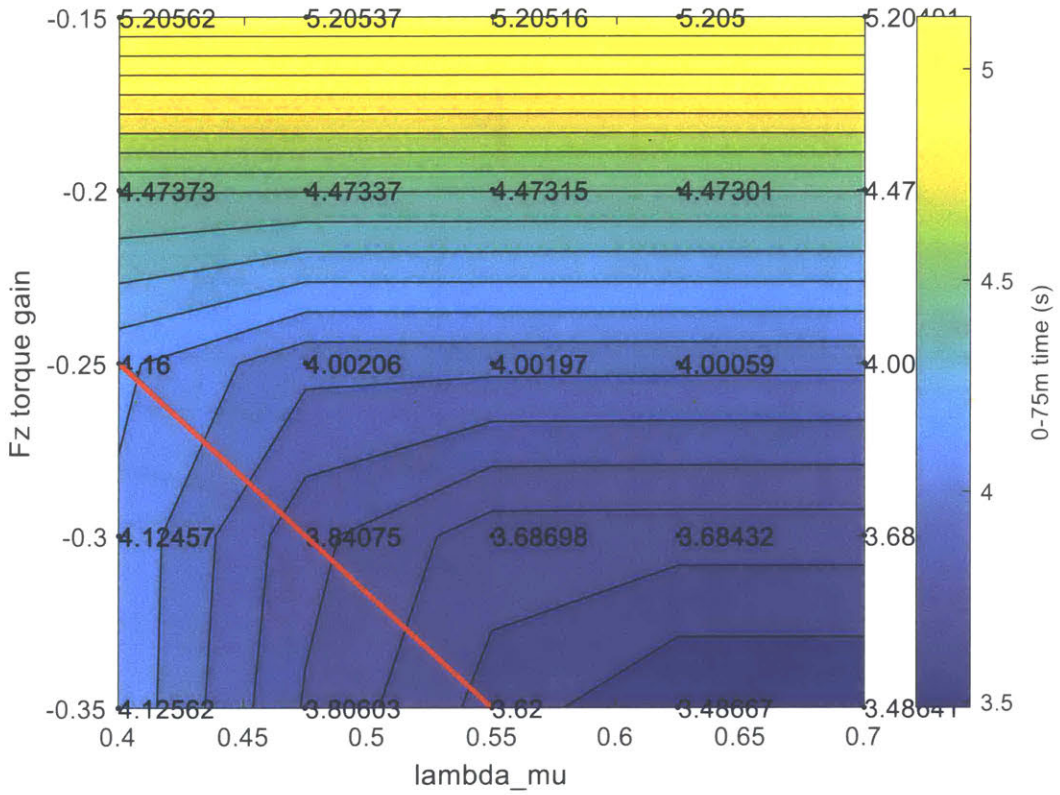

Figure 37: Results of normal load proportional controller study

5. CONCLUSION

A versatile vehicle simulation was written in Simulink for the purpose of investigating vehicle motion and designing and tuning controllers without the need for a physical testbed. The scope of this paper was restricted to straight line motion. Several launch control schemes were tested for both rear wheel drive and four wheel drive.

The simulation structure lends itself to a few purposes. Being written in a visual coding language, it is easier for a newcomer to understand the underlying physics. Adding subsystems is also more straightforward. Furthermore its structure is designed to be accurate to the actual vehicle, with a software block that is separate from the physics model. This makes it useful for testing control algorithms. It is also useful for conducting sensitivity studies as all the variables are modifiable from an external script.

From the results of the investigation into launch controllers, a few conclusions can be drawn. The constant torque command results in good performance but it is difficult to choose the correct torque command. The ramped torque command suffers from the same problem and seems to be even slower. For four wheel drive, a normal load proportional controller produced promising results, proving to be fast and stable given the correct gain value is chosen. However, the issue of lambda_mu, the road surface coefficient, is still outstanding. The ultimate challenge is to design an algorithm that is agnostic or insensitive to external conditions.

5.1 FUTURE WORK

One of the main purposes of this project was to begin the MIT FSAE team's venture into full vehicle simulation. Many paths can be taken from this point; however there are a few things that are appropriate to be done in the short term to improve the simulation.

1. Conduct tests to verify the simulation. The data available for the correlation described earlier was not of the best quality. Instrumenting all four wheels with Hall effect sensors, load cells on the pushrods, and carefully recording test conditions will provide useful data for correlation. It is also recommended to prepare different scenarios such as an acceleration-braking run or a coast down test. The constants that are most in need of tuning are the mechanical efficiency, rolling resistance, aero coefficients, and motor torque-speed curves.
2. Add noise and offset to the feedback signals. Noisy or inaccurate signals pose a large problem for implementing control algorithms on the physical system. The Simulink structure makes it easy to add noise. This would potentially cut down further on time required with the physical testbed. It could also help inform sensor selection with a more concrete requirement for accuracy.
3. Add complexity to the simulation. The next major subsystem to add for longitudinal motion is the battery, and with it comes voltage and current. A battery is an extremely complex system to model, but it is necessary to be able to test any power limiting algorithms. After the battery, add lateral capability. Lateral tire modeling is actually easier than longitudinal tire modeling, but is still a difficult task as it adds two degrees of freedom to the overall model.

6. APPENDIX: CODE DETAIL

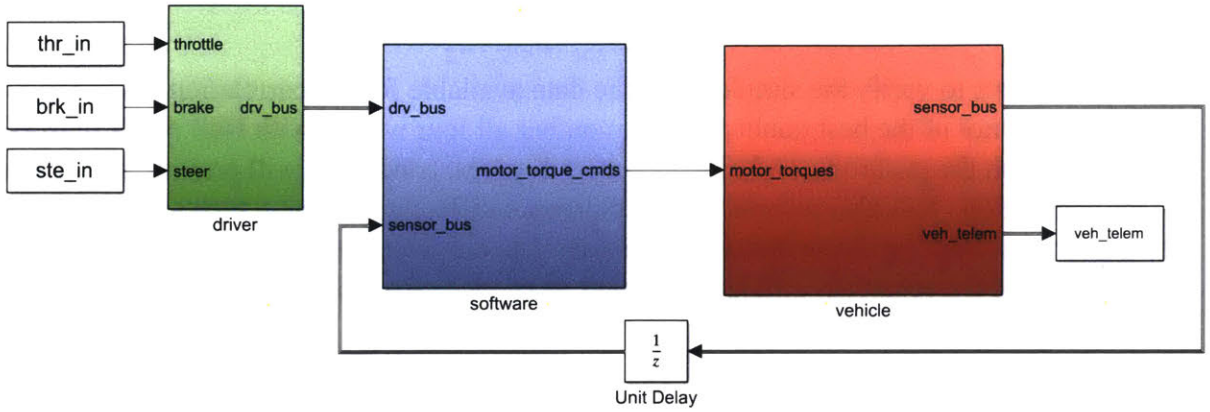

Figure 38: Top level

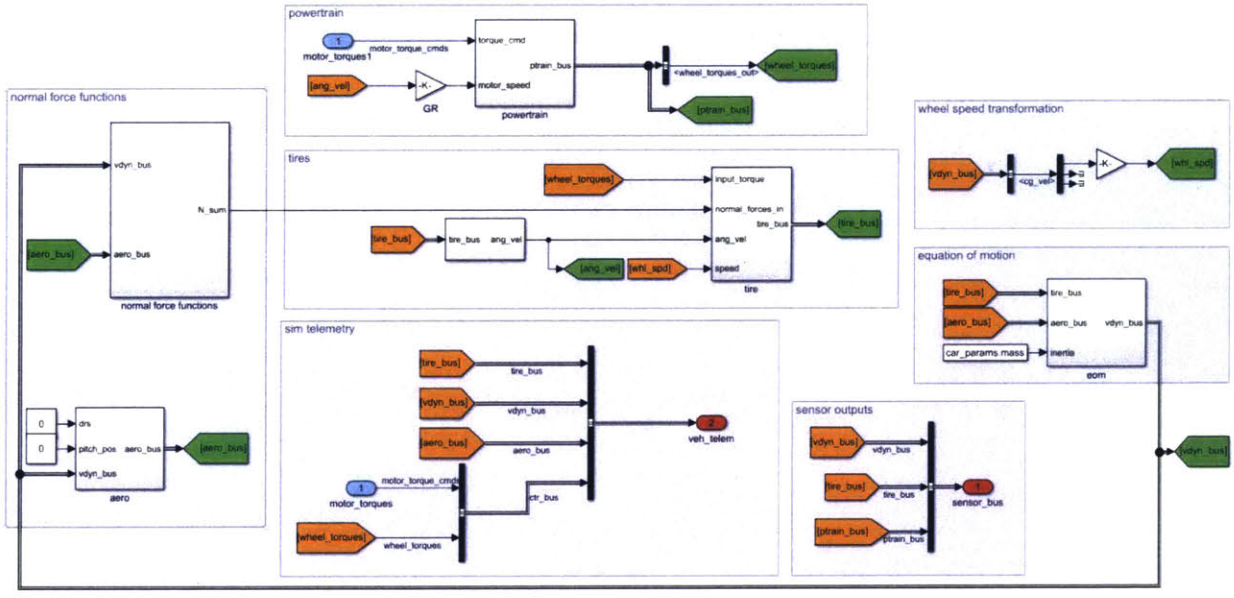

Figure 39: Inside the vehicle block

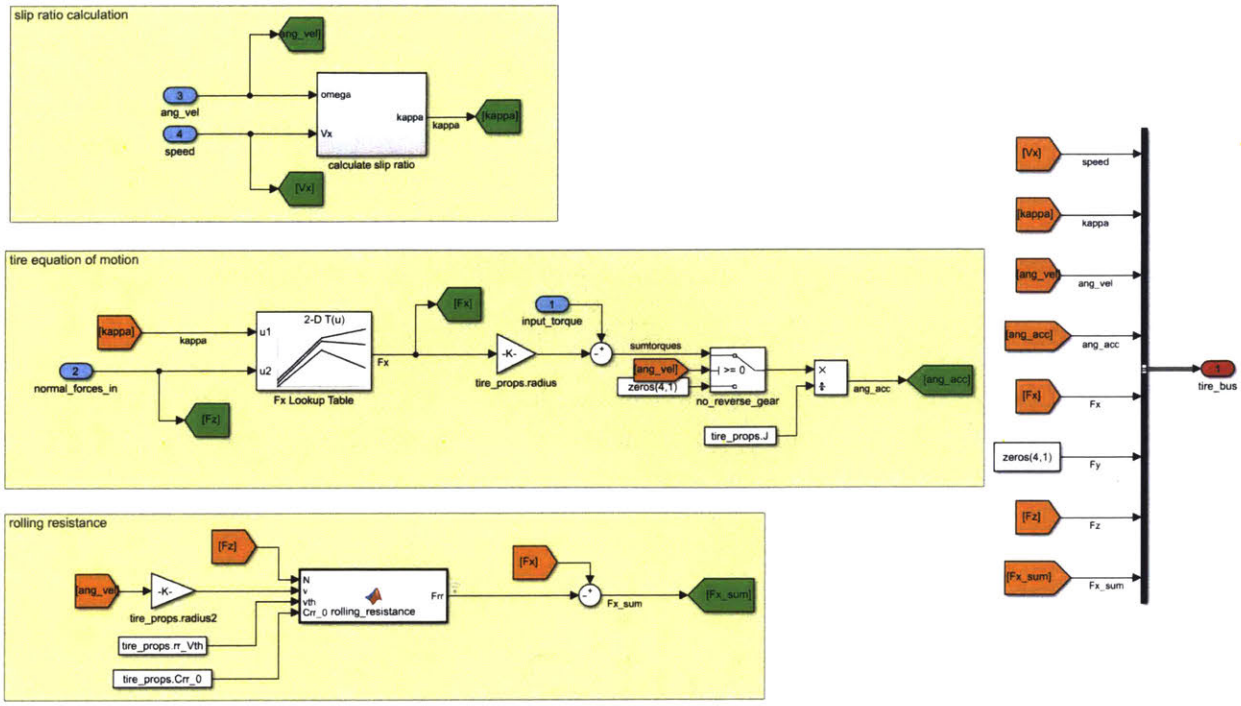

Figure 40: Inside the tire block

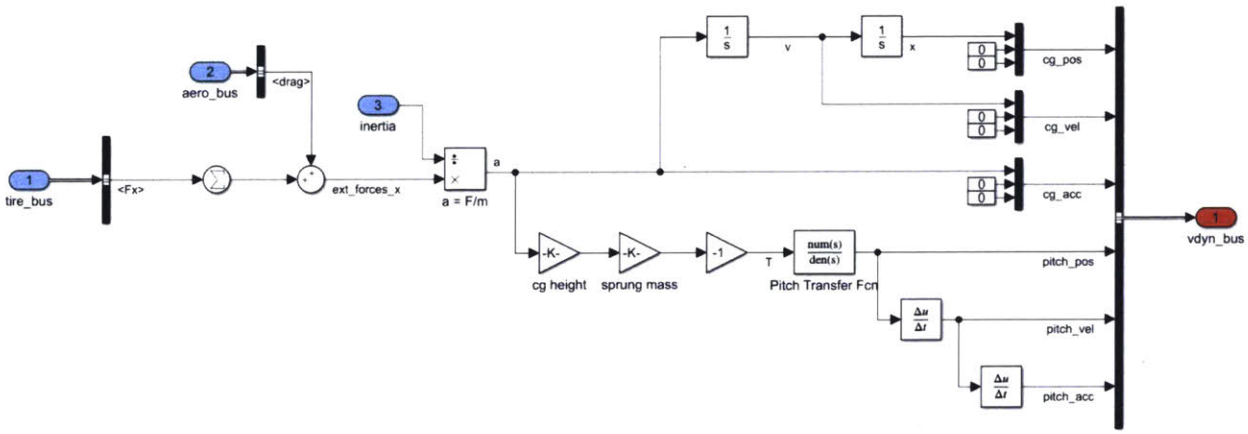

Figure 41: Equation of motion

7. REFERENCES

1. Pacejka, H.B., 2006, *Tire and Vehicle Dynamics*. 2nd Edition, Butterworth-Heinemann, Oxford.
2. "Tire (Magic Formula)," 2019, Mathworks, from https://www.mathworks.com/help/physmod/sdl/ref/tiremagicformula.html
3. "Optimum Tire Help File Version 1.1," Optimum G, Denver, CO.
4. " Model rolling resistance," 2019, Mathworks, from https://www.mathworks.com/help/physmod/sdl/ref/rollingresistance.html
5. Milliken, D. L., and Milliken, W.F., 1995, *Race Car Vehicle Dynamics*, Society of Automotive Engineers (SAE), Warrendale, MA.