

2011

Comparison of Three Degree of Freedom and Six Degree of Freedom Motion Bases Utilizing Classical Washout Algorithms

Christopher Daven Larsen
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Larsen, Christopher Daven, "Comparison of Three Degree of Freedom and Six Degree of Freedom Motion Bases Utilizing Classical Washout Algorithms" (2011). *Graduate Theses and Dissertations*. 10139.
<https://lib.dr.iastate.edu/etd/10139>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Comparison of three degree of freedom and six degree of freedom motion bases
utilizing classical washout algorithms**

by

Christopher Daven Larsen

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Mechanical Engineering

Program of Study Committee:
Greg Luecke, Major Professor
Pranav Shrotriya
Thomas Rudolphi

Iowa State University

Ames, Iowa

2011

Copyright © Christopher Daven Larsen, 2011. All rights reserved.

Table of Contents

List of Figures	iii
Nomenclature	v
Abstract	ix
Chapter 1: Introduction	1
Chapter 2: Vehicle Dynamics Simulator	9
Chapter 3: Washout Algorithm	21
Chapter 4: Tuning	26
Chapter 5: Results	35
Chapter 6: Future Work	55
Chapter 7: Conclusion	58
Bibliography	61
Appendix A: Vehicle Dynamics Matlab Code	63
Appendix B: Vehicle Vector Matlab Code	73
Appendix C: Motion Base Parameter Matlab Code	75
Appendix D: Washout Algorithm Simulink Code	76

List of Figures

Figure 1: Three-Degree of Freedom Motion Base	1
Figure 2: Six-Degree of Freedom Motion Base	2
Figure 3: NADS-1 Motion Base	3
Figure 4: NADS-1 Movement Constraints	4
Figure 5: Hydraulic Actuators and Car Cab Inside NADS-1 Dome	5
Figure 6: Driver View NADS-1	5
Figure 7: NADS MiniSim	6
Figure 8: SAE Vehicle Coordinate System	9
Figure 9: SAE Tire Force and Moment Axis System	11
Figure 10: Washout Algorithm	22
Figure 11: High Pass Filter	22
Figure 12: Low Pass Filter	23
Figure 13 : Tilt Coordination about Y-Axis	24
Figure 14: Tilt Coordination about X-Axis	24
Figure 15: Tuning Selected Maneuver	28
Figure 16: Tune Driver Selected Problem	30
Figure 17: Tuning Selected Coefficient, Driver Problem	31
Figure 18: Tune Limiting Problem	32
Figure 19: Tuning Selected Coefficient, Limiting Case	33
Figure 20: Limerock Vehicle CG Path	35
Figure 21: Longitudinal Velocity vs Time	36
Figure 22: Longitudinal Acceleration vs Time	37
Figure 23: Pitch Angle vs Time	38
Figure 24: Steering Angle vs Time	39
Figure 25: Lateral Acceleration vs Time	40
Figure 26: Plot of X and Y Accelerations on CG Path	41
Figure 27: Lateral Velocity vs Time	42
Figure 28: Speed vs Time	43
Figure 29: Yaw Rate vs Time	44
Figure 30: Heading Angle vs Time	45
Figure 31: Roll Angle vs Time	46
Figure 32: Lateral Accelerations of Simulation, 3DOF, and 6DOF Motion Base	47
Figure 33: Lateral Accelerations of Turn 1	49
Figure 34: Root Mean Square Error Turn 1	51
Figure 35: Longitudinal Accelerations of Simulation, 3DOF, and 6DOF Motion Base	52
Figure 36: Longitudinal Acceleration Braking, Coast, and Entering Turn 1	53
Figure 37: Longitudinal Acceleration Braking, Coast, and Entering Turn 1 Close Up	53

Figure 38: Root Mean Square Error Braking, Releasing Brake, and Entering Turn 1	54
Figure 39: Dynamometer Printout	55

Nomenclature

1. l = wheelbase
2. t_f = front track
3. t_r = rear track
4. a = longitudinal distance from front wheels to cg
5. b = longitudinal distance from rear wheels to cg
6. cg = center of gravity height
7. c = damping coefficient rolling
8. C = damping coefficient under applied load
9. w_f = weight front axel
10. w_r = weight rear axel
11. w_{fl} = weight front left tire
12. w_{fr} = weight front right tire
13. w_{rl} = weight rear left tire
14. w_{rr} = weight rear right tire
15. m = mass of vehicle
16. I_{zz} = moment of inertia about z-axis
17. I_{xx} = moment of inertia about x-axis
18. I_{yy} = moment of inertia about y-axis
19. C_{ar} = cornering stiffness rear tires
20. C_{af} = cornering stiffness front tires
21. μ = coefficient of friction
22. g = gravity
23. v = lateral velocity
24. r = yaw velocity about z-axis
25. x = x coordinate position
26. y = y coordinate position
27. ψ = yaw/heading angle

28. ψ_d = yaw/heading angle (degrees)
29. $\dot{\psi}$ = angular velocity about z-axis
30. u = longitudinal velocity
31. ϕ = roll angle
32. ϕ_d = roll angle (degrees)
33. $\dot{\phi}$ = angular velocity about x-axis
34. θ = pitch angle
35. θ_d = pitch angle (degrees)
36. $\dot{\theta}$ = angular velocity about y-axis
37. K = spring rate
38. \bar{p} = Longitudinal weight distribution factor
39. $\bar{\phi}$ = roll gain
40. δ_c = compliance steer
41. Q = ratio of steer to roll gain
42. t_{span} = time counter
43. k = understeer gradient
44. r_{turn} = turn radius
45. \bar{K} = spring stiffness
46. $\bar{\bar{K}}$ = load shift proportionality constant
47. f_x = total longitudinal force
48. Δt = change in time
49. BF_f = brake force front axel
50. BF_r = brake force rear axel
51. BF_{fl} = brake force front left tire
52. BF_{fr} = brake force front right tire
53. BF_{rl} = brake force rear left tire
54. BF_{rr} = brake force rear right tire
55. BF_{fl}/N = ratio brake force front left tire to normal force

56. $\frac{BF_{fr}}{N}$ = ratio brake force front right tire to normal force

57. $\frac{BF_{rl}}{N}$ = ratio brake force rear left tire to normal force

58. $\frac{BF_{rr}}{N}$ = ratio brake force rear right tire to normal force

59. δ = steer angle

60. δ_r = rear steer angle

61. θ_{fl} = pitch angle front left tire

62. θ_{fr} = pitch angle front right tire

63. θ_{rl} = pitch angle rear left tire

64. θ_{rr} = pitch angle rear right tire

65. α_{fl} = front left tire slip angle

66. α_{fr} = front right tire slip angle

67. α_{rl} = rear left tire slip angle

68. α_{rr} = rear right tire slip angle

69. f_{yr} = lateral force rear tire

70. f_{yrl} = lateral force rear left tire

71. f_{yrr} = lateral force rear right tire

72. f_{yfl} = lateral force front left tire

73. f_{yfr} = lateral force front right tire

74. ΔN_{rl} = lateral load shift rear left tire

75. ΔN_{rr} = lateral load shift rear right tire

76. ΔN_{fl} = lateral load shift front left tire

77. ΔN_{fr} = lateral load shift front right tire

78. ΔN_r = lateral load shift rear axel

79. ΔN_f = lateral load shift front axel
80. X_{fl} = x-position front left tire
81. X_{fr} = x-position front right tire
82. X_{rl} = x-position rear left tire
83. X_{rr} = x-position rear right tire
84. Y_{fl} = y-position front left tire
85. Y_{fr} = y-position front right tire
86. Y_{rl} = y-position rear left tire
87. Y_{rr} = y-position rear right tire
88. F_{xlr} = Longitudinal force left rear tire
89. F_{xlf} = Longitudinal force left front tire
90. F_{xrf} = Longitudinal force right front tire
91. F_{xrr} = Longitudinal force right rear tire
92. F_{xw} = Longitudinal Force
93. F_{yrf} = Lateral force right front tire
94. F_{ylf} = Lateral force left front tire
95. z = z-position
96. a_x = longitudinal acceleration
97. a_y = lateral acceleration
98. a_z = vertical acceleration
99. β = side slip angle
100. v_z = vertical velocity
101. S = speed
102. h = center of gravity height

Abstract

Vehicle Simulations have many practical applications ranging from leisure use, public safety, civilian training, and commercial design applications. Vehicle simulations are commonly designed to work with three-degree of freedom or six-degree of freedom motion bases. Each system has pros and cons, which make them more suitable for specific applications.

These two systems differ drastically when it comes to cost of hardware, software, and complexity. The three degree of freedom system is limited to rotation about the x, y, and z axes. This means this system is limited to causing the sensation of acceleration through rotation about these three axis and from maintaining a tilt angle and using gravity. The tilt in this system is limited typically to a 45 degree angle, which prevents the system from creating a sensation of acceleration over 0.707g's. The six-degree of freedom system not only has the ability to rotate about the three axes, but can move laterally along all three axes. The ability of this system to more accurately produce these forces would be a given, but is limited to the lateral distance available for movement about each of the three axes. This gives the system the ability to produce additional acceleration, which theoretically allows the system to change accelerations more freely.

Tuning of each system is accomplished by placing limits on the system's ability to rotate and move laterally to insure that the simulation does not request accelerations beyond the capabilities of the motion base. This will prevent the system from creating undesired sensations, which should not be present. These limits unfortunately will not prevent all undesired sensations. Altering control parameters contained in the washout algorithms is another method of tuning, which can attempt to prevent or limit the magnitude of these sensations.

The purpose of this study is to compare the two systems ability to replicate the forces or "feel" of actually riding in the vehicle. This comparison will show whether or not the extra

cost and effort in designing a more complex system is warranted for the motion bases applications.

Chapter 1: Introduction

Driving simulators without a doubt are more realistic when motion cuing is introduced. Adding the sensation of tilt and acceleration gives the driver the sensation of actually riding in the vehicle that is being driven. This is accomplished by using a motion base. Motion bases are available in multiple configurations, but the two of interest are a three-degree of freedom system and a six degree of freedom system.

The three-degree of freedom system has the ability to rotate about the x, y, and z-axes, which are commonly referred to as pitch, roll, and yaw in vehicle dynamics. This is done by using a motion base which consists of three actuators and servo motors, such as the one shown in Figure 1. The linear actuators have the ability to extend, which when properly controlled allows the motion base to rotate about all three axes simultaneously.



Figure 1: Three-Degree of Freedom Motion Base

Figure reprinted from [Servos](http://servos.com/Motion_Bases/Three_Axis_Seat_files/3%20axis%20without%20seat.jpg). 15 April 2011 <http://servos.com/Motion_Bases/Three_Axis_Seat_files/3%20axis%20without%20seat.jpg>.

The more complex six-degree of freedom systems have the ability to move laterally about all three axes and rotate about the pitch, roll, and yaw axes, such as the one in Figure 2. Six-degree of freedom systems achieve this by utilizing six actuators and servo motors. Once again the linear actuators can extend independently and, when properly controlled give this motion base the ability of to rotate and translate about all three axes simultaneously.



Figure 2: Six-Degree of Freedom Motion Base

Figure reprinted from [Wikipedia](http://upload.wikimedia.org/wikipedia/en/7/77/Motion_platform_6_axis.JPG). 15 April 2011 < http://upload.wikimedia.org/wikipedia/en/7/77/Motion_platform_6_axis.JPG>.

Driving simulators are used for many different applications. One obvious application is for creating realistic real-time driving simulators. These can be used for various studies such as determine reaction times for steering, braking, and shifting. Databases could be made to determine how reaction times are affected in various situations such as age, alcohol consumption, and distracted driving. Simulators could also be used to train law enforcement, race car drivers, and new drivers. They can also be used to recreate real life situations for use in court cases.

One of the most advanced driving simulators is the National Advanced Driving Simulator (NADS-1) shown in Figure 3.



Figure 3: NADS-1 Motion Base

Figure reprinted from National Advanced Driving Simulator Overview. Iowa City: National Advanced Driving Simulator, The University of Iowa, 2008-2010.

This thirteen degree of freedom simulator is located at University of Iowa's Oakdale Research Park Campus. The simulator consists of a 360 degree dome with an entire car cab inside the dome. The dome has the ability to rotate about the z-axis by 330 degrees in both the positive and negative directions. This is accomplished by utilizing a "yaw ring assembly", which is attached to a hydraulic six-degree of freedom hexapod motion base. In addition to this the whole base is mounted onto two belt driven beams enabling the whole base to move laterally 64 feet in both the x and y axes simultaneously. This enables the NADS-1 as seen in Figure 4, to produce lateral and longitudinal accelerations, which fixed six-degree of freedom hexapod would be unable to produce.



Figure 4: NADS-1 Movement Constraints

Figure reprinted from [National Advanced Driving Simulator Overview](#). Iowa City: National Advanced Driving Simulator, The University of Iowa, 2008-2010.

Inside the dome the car cab is mounted to the base by using four hydraulic actuators located in the wheel well, which independently move to produce the feeling of vibrations when driving over the road, as seen in Figure 5. This would also enable the car to be positioned in the proper roll and pitch angles, while the screen remains stationary producing the video to show the correct yaw angle. Since the car is enclosed in the dome this would prevent the driver from being able to visually perceive that the dome is moving preventing false visual motion cues. The 360 degree photo-realistic screen shown in Figure 6, gives the driver a view in all directions when driving in the simulator. The fact that an actual car is in the dome lets the driver use the actual brakes, gas, and steering wheel to control the car. The driver can even use the rear and side view mirrors to see the screen behind, which would show the virtual environment behind the car. In addition to the field of view an audio system produces sounds to imitate wind, tire, engine, and other vehicles sounds



Figure 5: Hydraulic Actuators and Car Cab Inside NADS-1 Dome

Figure reprinted from [National Advanced Driving Simulator Overview](#). Iowa City: National Advanced Driving Simulator, The University of Iowa, 2008-2010.



Figure 6: Driver View NADS-1

Figure reprinted from [Gadget Crave](#). 15 April 2011 <<http://gadgetcrave.frsrcrave.netdna-cdn.com/wp-content/uploads/2009/07/NADS-1-Driving-Simulator-4.jpg>>.

Considering the fact that it costs approximately eighty million dollars to develop the NADS-1 simulator and its lack of portability it would not be feasible for commercial or personal applications. The NADS-1 is the most ideal simulator for use, but for commercial and personal applications a low cost portable simulator would be ideal. The NADS MiniSim, shown in Figure 7, has multiple configurations including a desktop application and a quarter cab application. These systems can include real car steering wheels, brake pedal, gas pedal, and gear shifters. This system can also be coupled with a motion base.



Figure 7: NADS MiniSim

Figure reprinted from [National Advanced Driving Simulator Overview](#). Iowa City: National Advanced Driving Simulator, The University of Iowa, 2008-2010.

The purpose of this paper is to compare the accuracy of a three-degree of freedom motion base to a six-degree of freedom motion base. The washout algorithms are utilized for the six-

degree of freedom motion base and filters the acceleration by using a low pass filter for the acceleration, which will be handled by tilt coordination. This will leave only steady state accelerations, which are easily created by tilting the motion base and using a component gravity to simulate the sensation. The high pass filter leaves only high frequency accelerations, which are simulated more accurately by using lateral displacement. The three-degree of freedom motion base, which can only rotate must rely on rotation only. This means washout algorithms are inapplicable for this motion base, which must rely on tilt coordination in order to use a component of gravity to simulate any accelerations.

Chapter 2 will discuss the development of a six-degree of freedom vehicle simulator. This program was created in Matlab and the core of the program consists of ten differential equations, which are simultaneously solved in a loop of a specified time period to yield the pitch, roll, yaw, x-position, y-position, longitudinal velocity, lateral velocity, angular velocity about the roll axis, angular velocity about the pitch axis, and heading angle. Once these values are calculated the accelerations, tire forces, and other vehicle dynamic values can be determined.

Chapter 3 will discuss the functionality of the washout algorithms and how it is implemented by using the accelerations output by the vehicle simulator. Use of tilt coordination to simulate accelerations is discussed. High pass and low pass filters are also examined.

Chapter 4 discusses washout algorithm tuning methods. The ideal tuning situations are described showing how driver input is necessary for a proper tune. No motion base was available so a method without a driver was discussed for use with the washout algorithms for the six-degree of freedom system.

Chapter 5 discusses the results and verification of the vehicle simulation as well as the washout algorithm. The outputs of the vehicle simulation are discussed. The outputs of the washout algorithms are also discussed. Calculation of the root mean square deviation and

peak error are shown. The error of the three-degree of freedom system and the six-degree of freedom are compared.

Chapter 6 discusses what additional research and development can be done with the motion base and vehicle simulator. Commercial and personal applications are proposed, and specific applications are discussed. Additional programming and hardware, which will be useful for these applications, are described as well.

Chapter 7 summarizes the research, program development, and concludes which system is best for specific situations.

Chapter 2: Vehicle Dynamics Simulator

The vehicle dynamics simulation is based on ten differential equations which are simultaneously solved. The program requires (δ , BF_f , BF_r , and u) the inputs of steer angle, brake force front, brake force rear, and longitudinal or forward velocity. These ten differential equations provide the parameters required to accurately describe the vehicles position, velocity and orientation in 3-D space as seen in Figure 8.

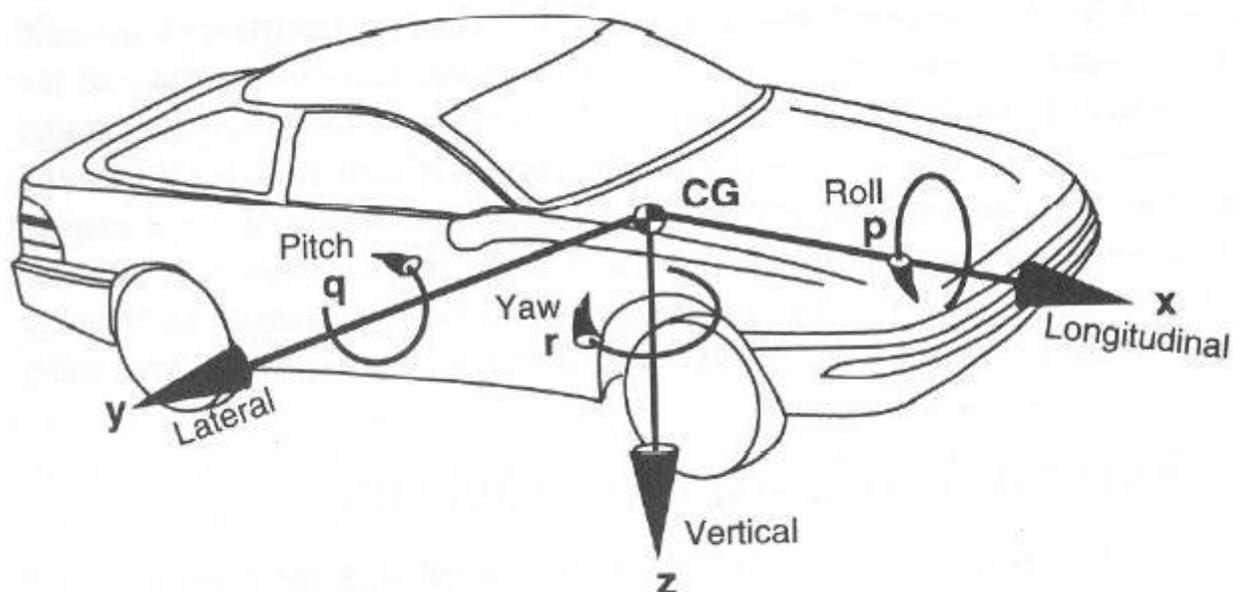


Figure 8: SAE Vehicle Coordinate System

Figure reprinted from Gillespie, Thomas D. *Fundamentals of Vehicle Dynamics*. Warrendale: Society of Automotive Engineers, 1992.

The outputs from the program can be used to describe the pitch, roll, yaw, x-position, y-position, and z-positions. When the program is run in a loop these values can be stored as a vector. These vectors can be used to create 3-D representation for use in simulations. These same vectors can again be used as inputs into classic washout algorithms.

The majority of the required vehicle parameters can be either found using references, measurements or through simple calculations. The basic parameters needed for the vehicle simulation are as follows: track, weight, mass, cornering stiffness and wheelbase. Using these parameters all the required constants can be calculated.

$$b = l \frac{w_r}{w}$$

$$k = \frac{w_f}{C_{\alpha f}} - \frac{w_r}{C_{\alpha r}}$$

Having already calculated all of the required vehicle parameters and constants, a loop will be created in order to create a vector to store all of variables which will represent the vehicles position, speed and orientation at any given time during the simulation. Having already calculated all of the required vehicle parameters and constants, a loop will be created in order to create a vector to store all of variables which will represent the vehicles position, speed and orientation at any given time during the simulation.

The dynamics of the vehicle can be broken down into four main components steering, braking, suspension, tires.

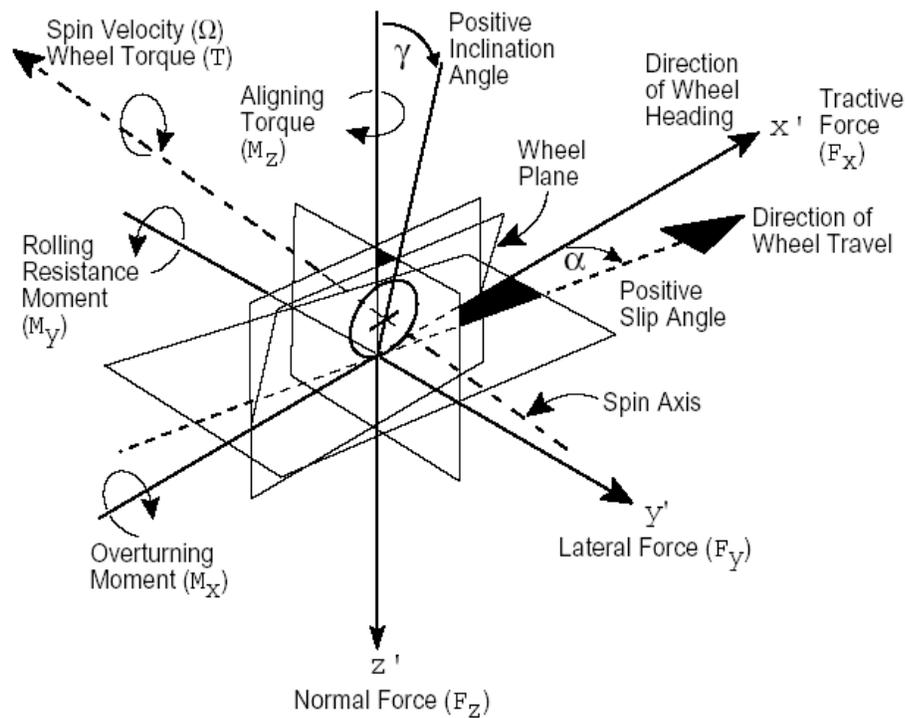


Figure 9: SAE Tire Force and Moment Axis System

Figure reprinted from [KXCAD](http://www.kxcad.net/MSOFT/Software/Adams_MD_R2/adams_tire/wwhelp/wwhimpl/common/html/wwhelp.htm?context=adams_tire&file=force_moment_89.html), 15 April 2011

<http://www.kxcad.net/MSOFT/Software/Adams_MD_R2/adams_tire/wwhelp/wwhimpl/common/html/wwhelp.htm?context=adams_tire&file=force_moment_89.html>.

Referring to the SAE tire force and moment axis system seen in Figure 9, it can be seen that there are three forces, which influence the movement of the tire. Assuming that the car is moving without sliding and a steering angle present all three forces will be present at that given time. The tractive force or longitudinal force will either be positive or negative depending on whether the car is accelerating or decelerating. Of course if the vehicle is coasting the traction force will be a small negative force due to rolling resistance.

Tire slip angle of each individual wheel can be calculating knowing the steering angle of each wheel (rear steering angle will be zero unless vehicle is equipped with rear steering), and the pitch angle of each individual tire. It can be assumed that no vehicle will have equal weight on each tire due to passengers having different weights, and the fact no vehicle has a

perfect weight distribution. This means that for the simulation each tire will have different forces applied at any given time.

To calculate tire slip angle of the front left tire the following equation is used. The equation can also be used to calculate the slip angle of the other front tire simply by using the associated pitch angle.

$$\alpha_{fl} = \theta_{fl} - \delta$$

In order to calculate the rear slip angle the rear roll steer will have to be calculated first.

$$\delta_r = Q\phi$$

This new steer angle will replace the original steer angle for the two rear tires. As before the associated pitch angle also needs to be used for the rear wheels.

In order to calculate all of the side slip angles the pitch of the tires must be calculated first. This equation involves longitudinal velocity, lateral velocity, track, yaw velocity, and the longitudinal distance from the cg to the front axle.

$$\theta_{fl} = \text{atan}\left(\frac{v - ar}{u + \frac{T}{2}r}\right)$$

There are slight variations in the equations for the other three tires, which take into account load transfer. Notice in the equation below, which shows if there is longitudinal load transfer the factor in the denominator is subtracted instead of added. This makes sense, if load was added to the front left tire it would have to be taken from the rear left tire.

$$\theta_{fr} = \text{atan}\left(\frac{v - ar}{u - \frac{T}{2}r}\right)$$

The same can be seen in equations for the rear tires, except the longitudinal distance from the cg to the rear axle is a different value.

$$\theta_{rl} = \text{atan}\left(\frac{v - br}{u + \frac{T}{2}r}\right)$$

$$\theta_{rr} = \text{atan}\left(\frac{v - br}{u - \frac{T}{2}r}\right)$$

Now that the slip angles are calculated the lateral force experienced can be calculated for each of the tires.

$$f_{yfl} = -C_{\alpha f} \alpha_{fl}$$

This simple equation can be used to calculate the lateral force of the other three tires simply by using the tires associated slip angle.

As mentioned before longitudinal load transfer is what causes the vehicle to pitch forward during braking and pitch backwards during accelerations. In order to calculate load shift caused by pitch the longitudinal load shift the following equation is used.

$$\Delta N_{fl} = -\bar{K} \theta_{fl}$$

This particular equation does have a constant that makes the load shift proportional to the pitch angle. In order to calculate this proportionality constant we assume that at a 1g steady state theta is equal to -0.1 degrees. We also know that if we have the total load transfer from the rear onto the front the following equation can be used.

$$\Delta N = W \frac{h}{l}$$

Rearrange the previous two equations we can solve for the constant.

$$\Delta N_{fl} = -\bar{K}\theta = W \frac{h}{l} \rightarrow \bar{K} = -\theta W \frac{h}{l}$$

This constant would be the same for each tire, but due to the variations the load shift would need to be calculated individually using its associated pitch angle.

Similarly the lateral load shift can be calculated. Lateral load shift is dependent on the roll angle. This is typically caused by the lateral force trying to counteract the centrifugal force during a turn. The lateral load shift will vary depending on the stiffness of the springs on the car.

$$\Delta N_f = \frac{-K\phi - c\dot{\phi}}{\frac{t_f}{2}}$$

In order to solve for the spring rate we assume a roll gain of negative 4 degrees per g.

$$K = \frac{wh}{\phi} + wh$$

Now that the lateral and longitudinal load shifts are known then we can calculate the weight on each tire at any given time during the simulation. The weight on each tire will change throughout the simulation due to weight transferring both laterally and longitudinally.

In order to calculate the loads on each of the tires on the front axle the following equations can be used.

$$w_{fl} = \frac{w_f}{2} + \Delta N_f \frac{\bar{p}}{2} + \frac{\Delta N_{fl}}{2}$$

The equation above can then be altered by subtracting in order to take the lateral load shift into account and by changing the longitudinal load shift value to the one of the associated tire.

$$w_{fr} = \frac{w_f}{2} - \Delta N_f \frac{\bar{p}}{2} + \frac{\Delta N_{fr}}{2}$$

Similarly the equation can be altered to take into account for the lateral load shift by changing \bar{p} to $(1-\bar{p})$ to make sure that the total weight distribution factor is one and setting the longitudinal load shift to the one associated with the rear left tire.

$$w_{rl} = \frac{w_r}{2} + \Delta N_r \frac{1 - \bar{p}}{2} - \frac{\Delta N_{rl}}{2}$$

Just as the equation was altered for the front right tire the lateral load shift is again subtracted and the longitudinal load shift is changed to the one associated with the rear right tire.

$$w_{rr} = \frac{w_r}{2} - \Delta N_r \frac{1 - \bar{p}}{2} - \frac{\Delta N_{rr}}{2}$$

Now that the weights are known on all four tires we need to determine whether or not the tires need to use the non-linear tire model. This is determined by the following inequality equation.

$$\frac{\mu w_{fl}}{2} < |f_{yfl}|$$

If the equation is true then the lateral force equation will change from the original equation mentioned earlier to the new equation below.

$$f_{yfl} = \frac{-\mu w_{fl} \text{sign}(\alpha_{fl})(1 - \mu w_{fl})}{4C_{\alpha f} |\tan(\alpha_{fl})|}$$

The inequality mentioned before can be used for the other three tires by changing the weight and lateral force to the values associated with the correct tire. Once the inequality is calculated then if the non-linear tire equation is needed it can be altered by changing the weight, tire cornering stiffness, and slip angles to the values associated with the correct tire.

Once the correct tire model is calculated the brake force can be allowable brake force can be calculated. Use of this inequality ensures that if more brake force is applied than is available on the road surface that the wheel locks up.

$$\frac{BF_{fl}}{N} > \mu$$

If the wheel does lock up then the brake force is equal to the coefficient of friction multiplied by the normal force or weight on that specific tire.

$$BF_{fl} = \mu w_{fl}$$

The inequality and brake equation above can be used for the other three tires by altering the brake force and weight to the values associated to the specific tire.

In order to take into account the fact that a tire can be sliding and turning at the same time the following inequality is utilized to insure that the brake force cannot be larger than the available force of μN .

$$BF_{fl} > \mu w_{fl} \cos \alpha_{fl}$$

If the inequality is true than the brake force and lateral force must be limited to the following values.

$$BF_{fl} = \mu w_{fl} \cos \alpha_{fl}$$

$$f_{yfl} = -\mu w_{fl} \sin \alpha_{fl}$$

Once again this inequality and the two subsequent equations can be used for the other three tires by changing the weight and slip angle to the values associated with that specific tire.

In the case that the tires have continued to rotate during heavy braking it is still possible that the combination of brake force and lateral force cannot be supported by the tires. This can be checked by using the following inequality.

$$BF_{fl}^2 + f_{yfl}^2 > (\mu w_{fl})^2$$

If this inequality is true than the brake force will remain equal to the value that is requested, but the lateral force will be limited to the value required for the tire to be able to support it.

$$f_{yfl} = -\sqrt{(\mu w_{fl})^2 - BF_{fl}^2} \text{sign}(\alpha_{fl})$$

This inequality and equation can be used for the other three tires by changing the weight, lateral force, brake force, and slip angle to be the value associated with the specific tire.

Now that the tire lateral and longitudinal forces are calculated they can be rotated into the vehicle coordinate system. The lateral forces can be calculated using the following equation.

$$F_{ylf} = f_{yfl} \cos(\delta) + f_{xlf} \sin(\delta)$$

$$F_{ylr} = f_{yfr} \cos(\delta) + f_{xrf} \sin(\delta)$$

Similarly the longitudinal forces can be calculated using the following equations.

$$F_{xlf} = f_{xlf} \cos(\delta) - f_{yfl} \sin(\delta)$$

$$F_{xlr} = f_{xlr} \cos(\delta) - f_{yrl} \sin(\delta)$$

Note that the longitudinal force on is simply equal to negative braking force for each of the individual tires.

Now that the longitudinal and lateral forces have been rotated into the vehicle coordinate system they can be summed together.

$$F_x = F_{xlf} + F_{xrf} + F_{xlr} + F_{xrr}$$

$$F_y = F_{ylf} + F_{yrf} + F_{ylr} + F_{yrr}$$

These sums can then be used to calculate the longitudinal and lateral accelerations.

$$a_x = \frac{F_x}{m \cdot g}$$

$$a_y = \frac{F_y}{m \cdot g}$$

Now that the required constants, lateral forces, and longitudinal forces are calculated the ten differential equations can be simultaneously solved. This is done using the Euler method as shown below.

$$y_{n+1} = y_n + f(x_n, y_n) d_t$$

The ten differential equations are simultaneously solved yielding the following:

Yaw Equation

$$\frac{dr}{dt} = \frac{\left[(F_{ylyf} + F_{yryf})a - (f_{yrr} + f_{yrl})b + (F_{xlyf} - F_{xryf} + F_{xlyr} - F_{xryr}) \left(\frac{5.2}{2} \right) \right]}{I_{zz}}$$

Lateral Velocity Equation

$$\frac{dv}{dt} = \frac{(F_{ylyf} + F_{yryf} + f_{yrl} + f_{yrr}) - ur - cg \left(- \left(K - (w_f + w_r) \right) (cg) \right) \phi + c\dot{\phi} + (F_{ylyf} + F_{yryf} + f_{yrl} + f_{yrr})}{I_{xx}}$$

Heading Angle

$$\frac{d\psi}{dt} = r$$

X Coordinate of Center of Gravity

$$\frac{dx}{dt} = u \cos \psi - v \sin \psi$$

Y Coordinate of Center of Gravity

$$\frac{dy}{dt} = v \cos \psi + u \sin \psi$$

Roll Angle

$$\frac{d\phi}{dt} = \dot{\phi}$$

Angular Velocity about Roll Axis

$$\frac{d\dot{\phi}}{dt} = \frac{- \left((K - (w_f + w_r)cg) \phi + c\dot{\phi} \right) + (F_{ylyf} + F_{yryf} + F_{yryr} + F_{yrl})cg}{I_{xx}}$$

Longitudinal Velocity

$$\frac{du}{dt} = \frac{F_x + vr}{m}$$

Pitch Angle

$$\frac{d\theta}{dt} = \dot{\theta}$$

Angular Velocity about Pitch Axis

$$\frac{d\dot{\theta}}{dt} = \frac{-(BF_f + BF_r)cg - \bar{k}\theta - C\dot{\theta}}{I_{yy}}$$

Once the ten differential equations are solved for they can be used to calculate the speed and side slip angles. The speed is simply calculated using the Pythagorean Theorem.

$$S = \sqrt{u^2 + v^2}$$

The side slip angle is then calculated.

$$\beta = \tan^{-1} \frac{v}{u}$$

Chapter 3: Washout Algorithm

Classic washout algorithms use a combination of high pass filters, low pass filters, gains, limits, sums, and sin functions, and transfer functions. The purpose of this algorithm is to calculate what forces a motion base can recreate when the system is constrained to a certain number of degrees of freedom and when the rates of rotation and lateral movement are limited due to human perception of movement. For six degree of freedom systems the acceleration is handled using both lateral and angular movement to replicate the true acceleration experienced. The lateral component is extracted by using a high pass filter, which eliminates steady state acceleration. Six degree of freedom motion bases can only move laterally in a limited space, which makes it impossible to create a sustained acceleration. Once the acceleration is filtered it will eliminate any low frequency accelerations, leaving only high frequency accelerations, which can be recreated using short quick lateral movements. This can be seen by referring to Figure 11, which shows two line graphs. The first line is the actual acceleration has several steady state accelerations. The second is the filtered acceleration, which shows all of the steady state accelerations as zero. The only acceleration remaining is of a high frequency. Next the acceleration is integrated twice to calculate the position. Once the data is filtered a closed control loop, which contains a combination of integration and gains as shown in Figure 10. All three off the control loops present in the Simulink code are equivalent to the following transfer function.

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The angular component is extracted by utilizing a low pass filter. This filter removes any high frequency accelerations, which cannot be accurately recreated using rotation. It can be seen that when filtered the high frequency acceleration or spikes in accelerations are filtered from the data as seen in Figure 12.

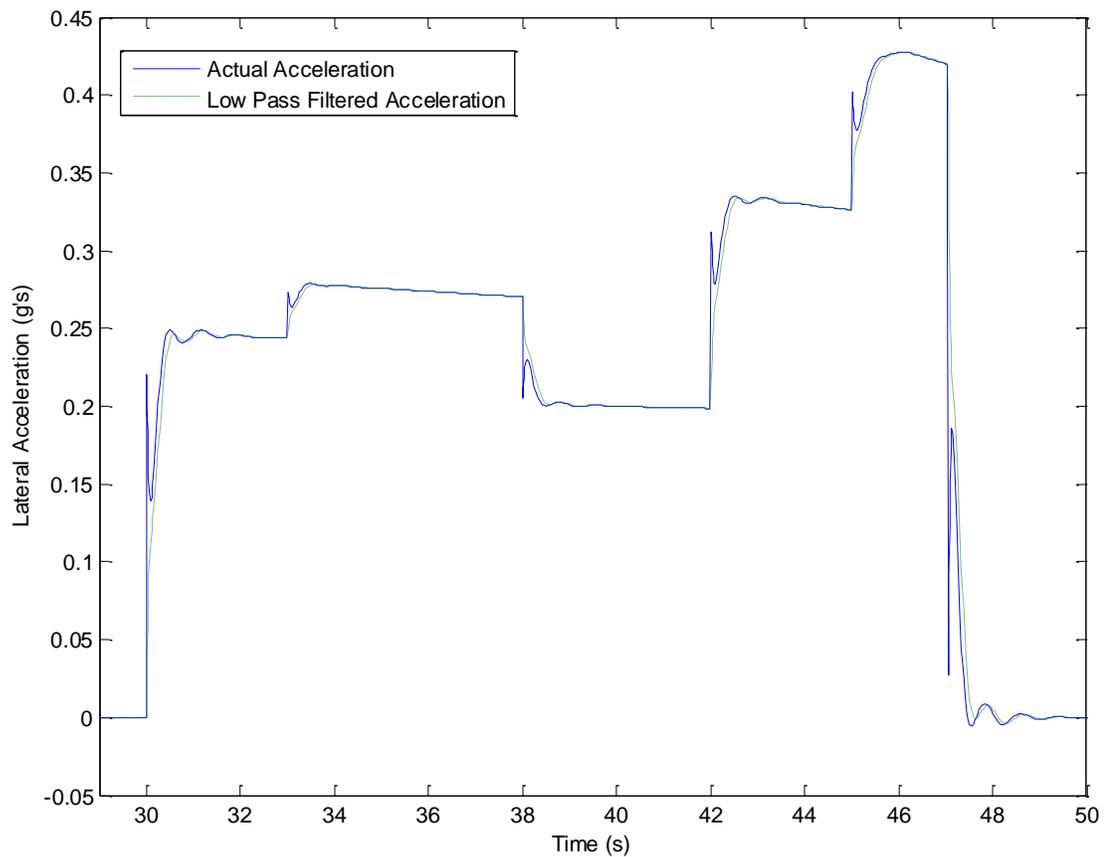


Figure 12: Low Pass Filter

Once again the first line is the actual acceleration, and the second line is the low pass filtered acceleration. Clearly all of the spikes in acceleration are eliminated leaving only low frequency acceleration and steady state acceleration. In order to replicate high frequency accelerations the motion base would have to rotate at a high rate, which potentially could be above the human threshold for rotation. Rotation can only use a component of gravity in order to produce force. This is done by moving the motion base to the proper angle to produce force in the desired direction as seen in Figure 13 and 14.

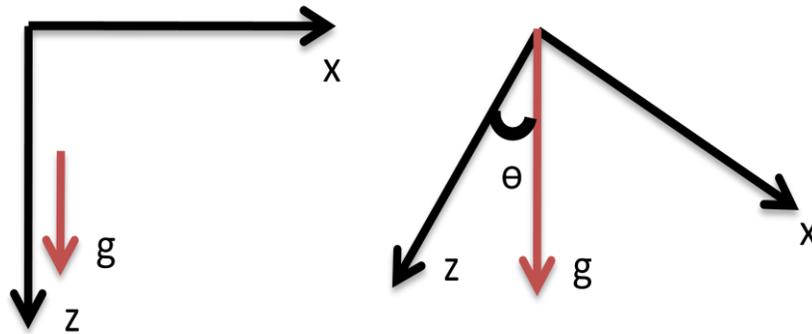


Figure 13 : Tilt Coordination about Y-Axis

The component of gravity, which would be felt in the x-direction can be solved by simple geometry.

$$\ddot{x} = g \sin \theta$$

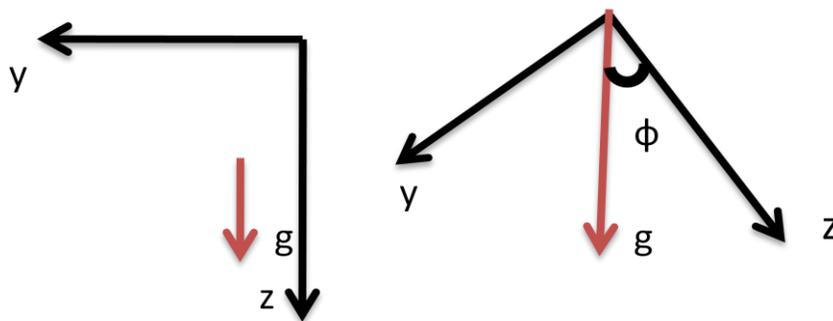


Figure 14: Tilt Coordination about X-Axis

Similarly the acceleration in the y-direction can be calculated.

$$\ddot{y} = g \sin \phi$$

Once the data is filtered the angle control loop applies acceleration and rate limits. The control loop uses same combination of gains, integrators as the previously mentioned controller, but has its own natural frequencies and damping. There also are an acceleration

limit and rate limit which was not present in the lateral controller. Upon exiting the control loop it is sent through a sin function to correct the direction of gravity so that it is applied to properly to the lateral or longitudinal accelerations.

A three degree of freedom system can only use rotation in order to recreate the force experienced. Since the base is unable to move laterally the data is not filtered using high and low pass filters. The motion base also can only produce a force, which is equivalent to gravity multiplied by the sine of the angle at which the motion base is rotated about the x and y axes. For example if the maximum rotation angle about a single axis is approximately forty-five degrees then the maximum force produced by the motion base would be $0.707 g$'s. The data is just sent through an arcsine and a gain function. The data is then sent into a control loop and afterwards is sent through a sine function once again to correct the direction of the gravitational acceleration so that it is properly applied to longitudinal and lateral accelerations.

Chapter 4: Tuning

Tuning is an important part of obtaining realistic feelings from the motion simulator. This is done by changing the motion base parameters in order to reduce the effects of false cues and more accurately replicate the acceleration profiles.

These false cues are defined as:

- A motion cue in the simulator that is in the opposite direction to that in the vehicle
- A motion cue in the simulator where none was expected
- A relatively high-frequency distortion of a sustained cue in the simulator for an expected sustained cue
- The most destructive cueing errors from a perceived fidelity point of view.

[Grant and Reid, 1997]

Specific methods on handling tuning can be addressed by determining the root cause of the particular cuing error. These particular methods are presented below:

Software or hardware limiting:

Limiting can cause large false cues caused by deceleration in the simulator, which is required to prevent damage to the motion system. Increasing natural frequency or damping for the high pass filter will reduce the angular displacement of the simulator. Increasing natural frequency of the high pass filter or decreasing the natural frequency of the low pass filter will reduce the simulator surge or sway displacement. Increasing high pass natural frequency, high pass damping, or increasing the order will reduce the heave displacement. Reducing the gain will reduce the displacement.

Return to Neutral

This false cue is attributed with the overshoot of the high pass filters to step inputs.

Increasing high pass damping can reduce the severity of the false cue for the angular and

heave channels. Reducing high pass natural frequency or increasing low pass natural frequency will reduce the severity of the false cue.

G-Tilt

Roll and pitch angles can lead to false cues from the x or y components of gravity. Increasing high pass natural frequency can reduce the severity of this type of cue. Reducing gain can also reduce the severity of this type of cue.

Tilt Coordination Angular Rate

The angular rate used to simulate sustained forces can also produce false cues. Reducing low pass natural frequency will reduce the severity of this type of false cue. Reduction of the tilt-rate limits can also reduce the severity of this cue. Reducing gain can also reduce this error.

Tilt-Coordination Remnant

When a sustained force is desired a steady-state pitch or roll angle is required. If this input ends abruptly it will cause the high pass specific force to initially cancel out the specific force associated with the tilt. This happens for a short time before the restricted displacement of the simulator prohibits translational acceleration. If the tilt is removed instantaneously a tilt-coordination angular rate false cue will occur or the remaining tilt will cause a sensation of acceleration called a tilt-coordination remnant false cue. Reducing high pass natural frequency or increasing low pass natural frequency will reduce the severity of this cue. Reducing the gain will also reduce the severity as well. If tilt rate limiting is significant, then increasing the limits of the tilt rates can reduce the severity.

Scaled or Missing Cues

Missing cues are considered extreme cases of scaled cues. Scaled or missing cues do not lead to the same reduction in perceived fidelity as false cues [Grant and Reid, 1997]. High pass filters can be too restrictive, leaving only high frequency initial cues, which can cause jerky motion but reduction of high pass natural frequency and high pass damping can reduce this.

If the low pass filter is too restrictive then increasing the low pass natural frequency can fix this problem. Raising the tilt rate limits can also help this problem. [Grant and Reid, 1997]

Grant and Reid also describe a procedure for tuning that follows a series of charts to determine the best fit coefficients. The process begins with selecting a specific maneuver to use for the purpose of tuning, for example the maneuver could be the first turn on the racetrack as described before. As shown in Figure 15 the maneuver is run through the simulation in order to determine whether or not the motion limit is reached by either hardware or software limits.

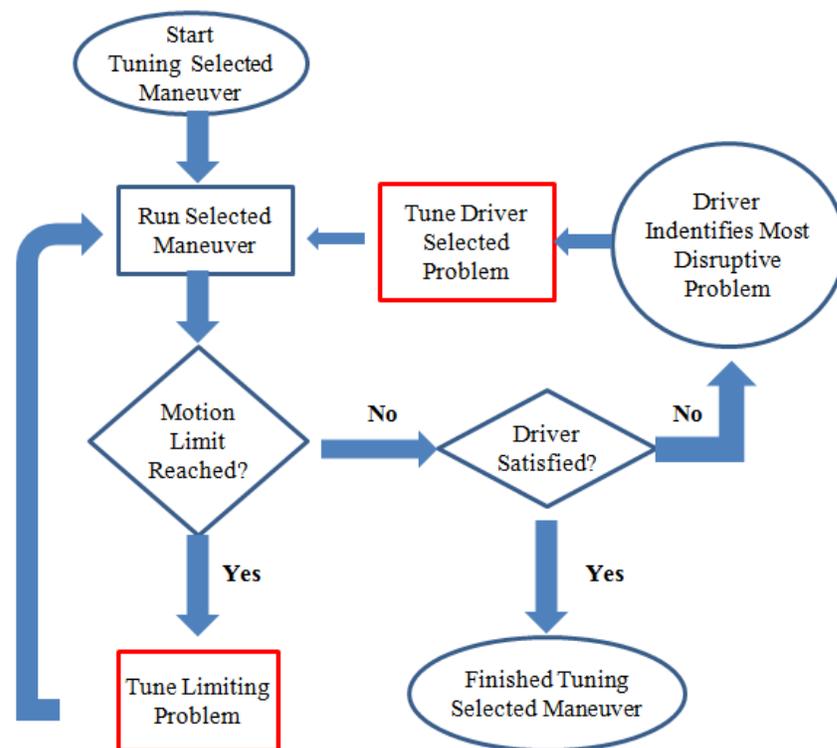


Figure 15: Tuning Selected Maneuver

Figure adapted from Grant, Peter R., and Lloyd D. Reid. "Protest: An Expert System for Tuning Simulator Washout Filters." *Journal of Aircraft* 34.2 (1997): 152-159.

If the limit is not reached then the driver should be questioned in order to determine if the feel of the simulation is satisfactory. If the driver feels satisfied then the tuning is complete, but if it is not satisfactory then the driver is asked to identify the "Most Disruptive Problem".

Once the problem is identified then the “Tune Driver Selected Problem” chart as shown in Figure 16 is started. However if the limit has been reached the driver would not have been questioned and the “Tune Limiting Problem” chart is the next step instead of “Tune Driver Selected Problem.”

Going back to having the motion limits not reached the “Tune Driver Selected Problem” is started as shown in Figure. When tuning the selected problem all the coefficients are considered as potential candidates for tuning as shown in the first block in Figure 16. The most likely coefficient is then adjusted by a large amount to insure that a significant change will show in the simulation. The simulation is ran and monitored to determine if a significant change is found in the simulation. If no significant change is found then the adjustment direction is reversed. The simulation is then sent back to make the adjustment and monitor the simulation. If the reversed adjustment does not show a significant change then the coefficient is deleted from the list of potential candidates. If a significant change did occur and the limit was exceeded then the adjustment is reversed and retested. If the limit was not exceeded then the result is compared with the original simulation. If the selected problem changes due to the adjustments then a new coefficient is picked for tuning. If the problem did not change then the adjustment is undone and if all of the coefficients have been deleted then the process starts over if the tuning hasn't been tested twice. If all of the coefficients had not been deleted then the tuning starts over again. Assuming that the problem had changed and the new coefficient has been picked then it is sent to “Tune Selected Coefficient”. Once the selected coefficient is tuned the problem tuning is monitored. The problem tuning either is stopped at this point or the coefficient is deleted and a new coefficient is picked at the beginning of the driver selected problem.

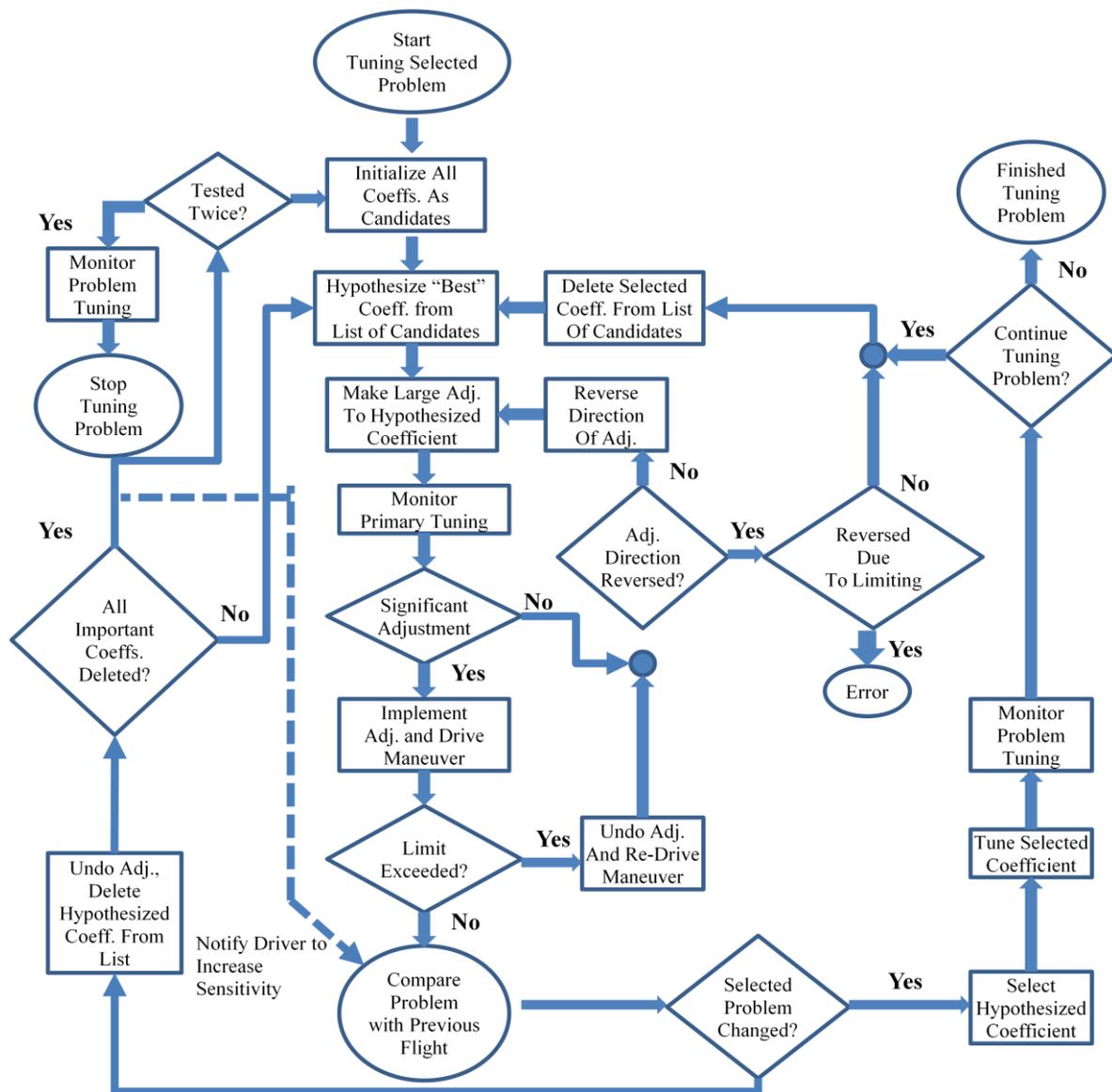


Figure 16: Tune Driver Selected Problem

Figure adapted from Grant, Peter R., and Lloyd D. Reid. "Protest: An Expert System for Tuning Simulator Washout Filters." *Journal of Aircraft* 34.2 (1997): 152-159.

Referring back to the point that the selected problem changed the "Tune Selected Coefficient" chart is reviewed as seen in Figure 17. The driver first describes the problem and its severity. The primary coefficient adjustment is calculated and the simulation is run. The simulation is then reviewed to determine whether or not tuning of the coefficient should continue. If tuning is stopped, then a second coefficient may be tuned. If it is decided not to

tune a second coefficient then the tuning coefficient is complete. If a second coefficient is tuned then first it is selected, calculated, and monitored. The primary coefficient is set and tuning is either completed or continued at this point. If the tuning continues the simulation is run in order to determine if limiting is present on the flight. If it is limited the primary coefficient is recalculated. If motion is not limited the secondary coefficient has not been adjusted then the driver compares the motion to the previous run. If it is the first time the motion got worse, and a secondary problem is found the driver is asked to identify the secondary problem and the chart is restarted. Had the coefficient already been adjusted then the chart restarts immediately.

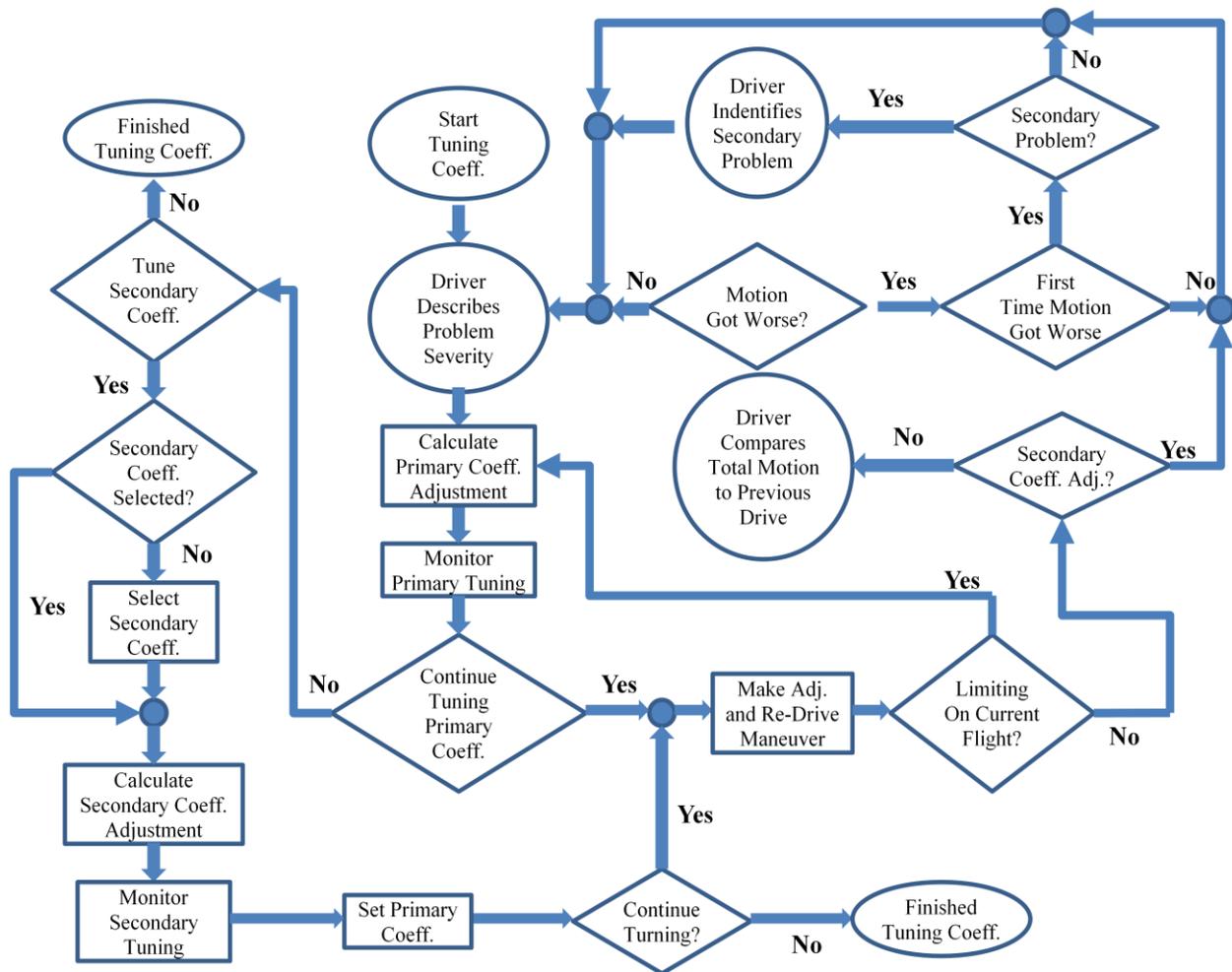


Figure 17: Tuning Selected Coefficient, Driver Problem

Figure adapted from Grant, Peter R., and Lloyd D. Reid. "Protest: An Expert System for Tuning Simulator Washout Filters." *Journal of Aircraft* 34.2 (1997): 152-159.

Going back to the first chart and assuming that the motion limit had been reached the tune limiting problem chart is reviewed. In this chart all of the coefficients start as potential candidates. First the best coefficient is decided and then the tune selected coefficient chart is started. Assuming this was already completed the simulation is ran and monitored. If the problem appears to be fixed the tuning limit problem is complete. If the problem needs more tuning then the coefficient is deleted and the process starts over. Had all of the coefficients been deleted an error should be present.

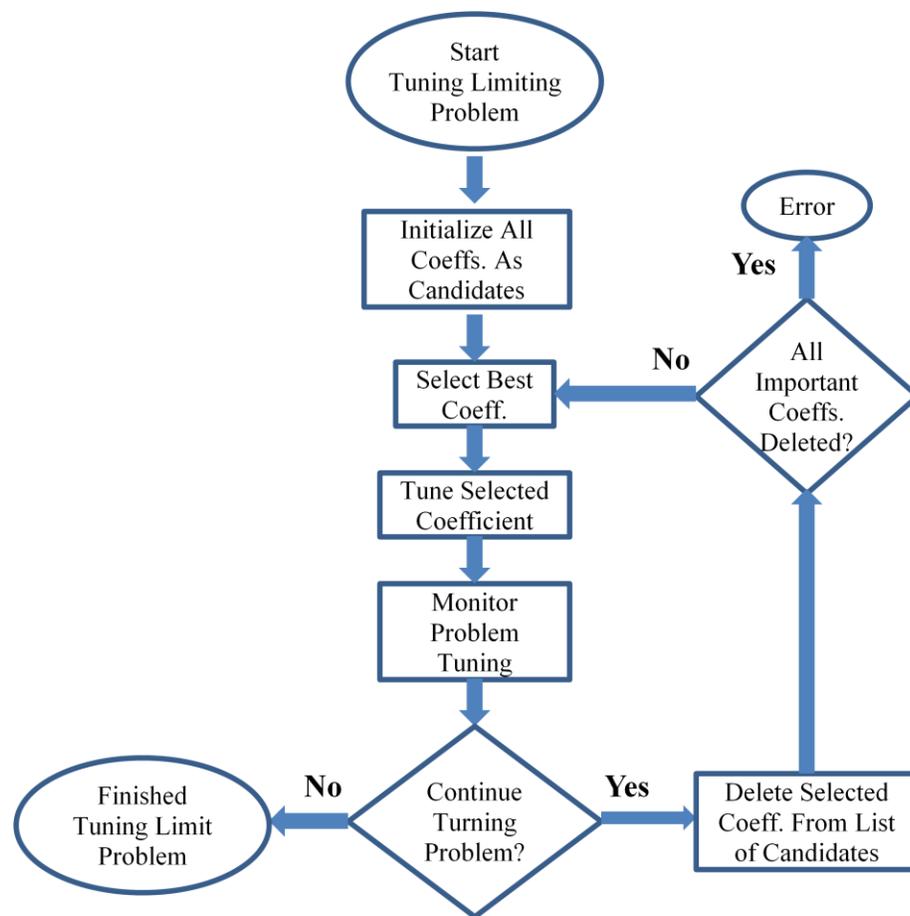


Figure 18: Tune Limiting Problem

Figure adapted from Grant, Peter R., and Lloyd D. Reid. "Protest: An Expert System for Tuning Simulator Washout Filters." *Journal of Aircraft* 34.2 (1997): 152-159.

The tune selected coefficient chart is now examined as seen in Figure 19, having been a requirement for the tune-limiting problem chart as seen in Figure 18. First the primary coefficient adjustment is calculated and then the simulation is ran and monitored. The decision to continue tuning the primary coefficient or to move on to the secondary coefficient is made. If further tuning is required then the primary coefficient is recalculated and the process starts over. If tuning on the primary coefficient was completed then the decision whether or not to tune a second coefficient is made. If the coefficient is to be tune and has not already been selected then it is selected at this time. The secondary coefficient adjustment is calculated and the simulation is ran. It is monitored and then the primary coefficient is set. The decision is then made whether or not to continue tuning or to stop.

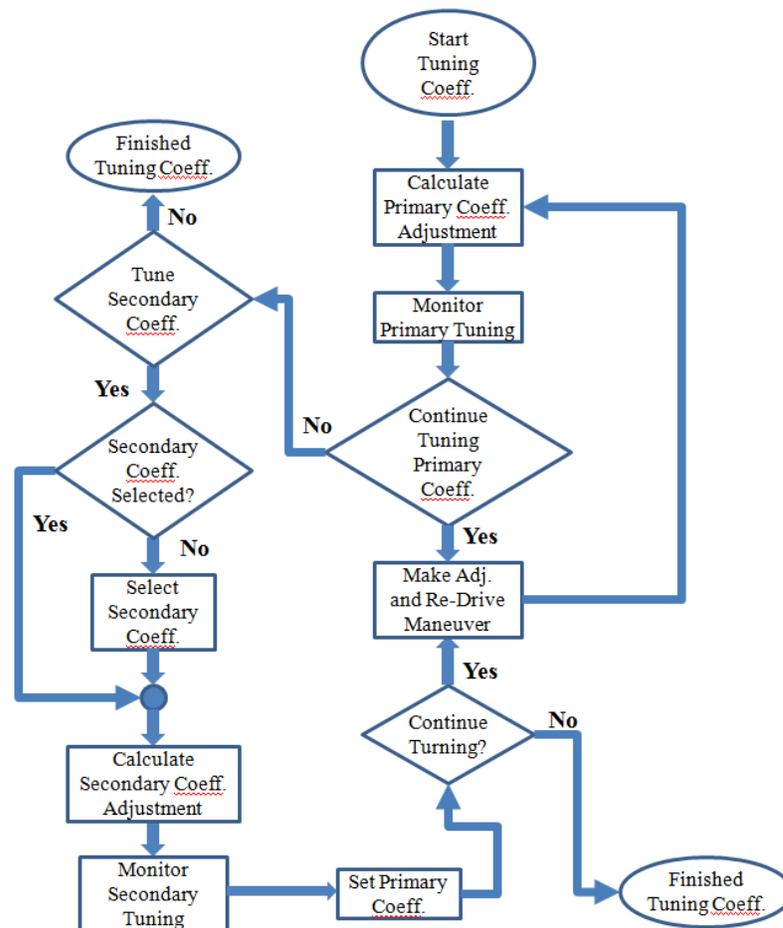


Figure 19: Tuning Selected Coefficient, Limiting Case

Figure adapted from Grant, Peter R., and Lloyd D. Reid. "Protest: An Expert System for Tuning Simulator Washout Filters." *Journal of Aircraft* 34.2 (1997): 152-159.

Tuning of the washout algorithm and motion base is mostly based on trial and error. Even with the structured approach and adjustments mentioned by Grant and Reid no one can make a perfect tune. Grant and Reid mention several methods which are attempted to model human reaction to motion sensed by the vestibular, proprioceptive and tactile systems [Grant and Reid, 1997] Most of the attempts either do not currently exist, have not been properly reviewed, or are not very well established. This requires tuning to rely on the people controlling the vehicle and using their responses to tune the system as shown in the approaches previously mentioned.

However in the case of the vehicle simulator, motion bases were not readily available therefore no driver input could be used. This forced the tuning to be done with simple trial and error methods. For the three degree of freedom system there is no application of washout algorithms so no tuning is required. Also the hardware and software limits were already set for this motion base so no limits needed to be placed. However in future cases limits could be placed on the software, therefore keeping the base from reaching its limits.

The six degree of freedom motion base does use the washout algorithm, but the hardware limits are preset. For use with the vehicle simulator the limits for the acceleration and rotation rates were made high enough so no adjustments were necessary. This left only five available coefficients for tuning. The coefficients are as follows natural frequency of the low pass filter, natural frequency of the high pass filter, time constant for the low pass filter, damping coefficient for the low pass filter, and damping coefficient for high pass filter. Trial and error was simply used to determine what effects the large adjustments made to the acceleration output from the six degree of freedom motion base. Once these adjustments were made smaller adjustments were made one at a time to get the acceleration match the magnitude of the crests and trough as close as possible. The lag was also attempted to be kept at a minimum if possible.

Chapter 5: Results

In order to show the abilities of the vehicle dynamics simulation the car was driven over a 1.5 mile racetrack located in Lakeville, CT as seen in Figure 20.

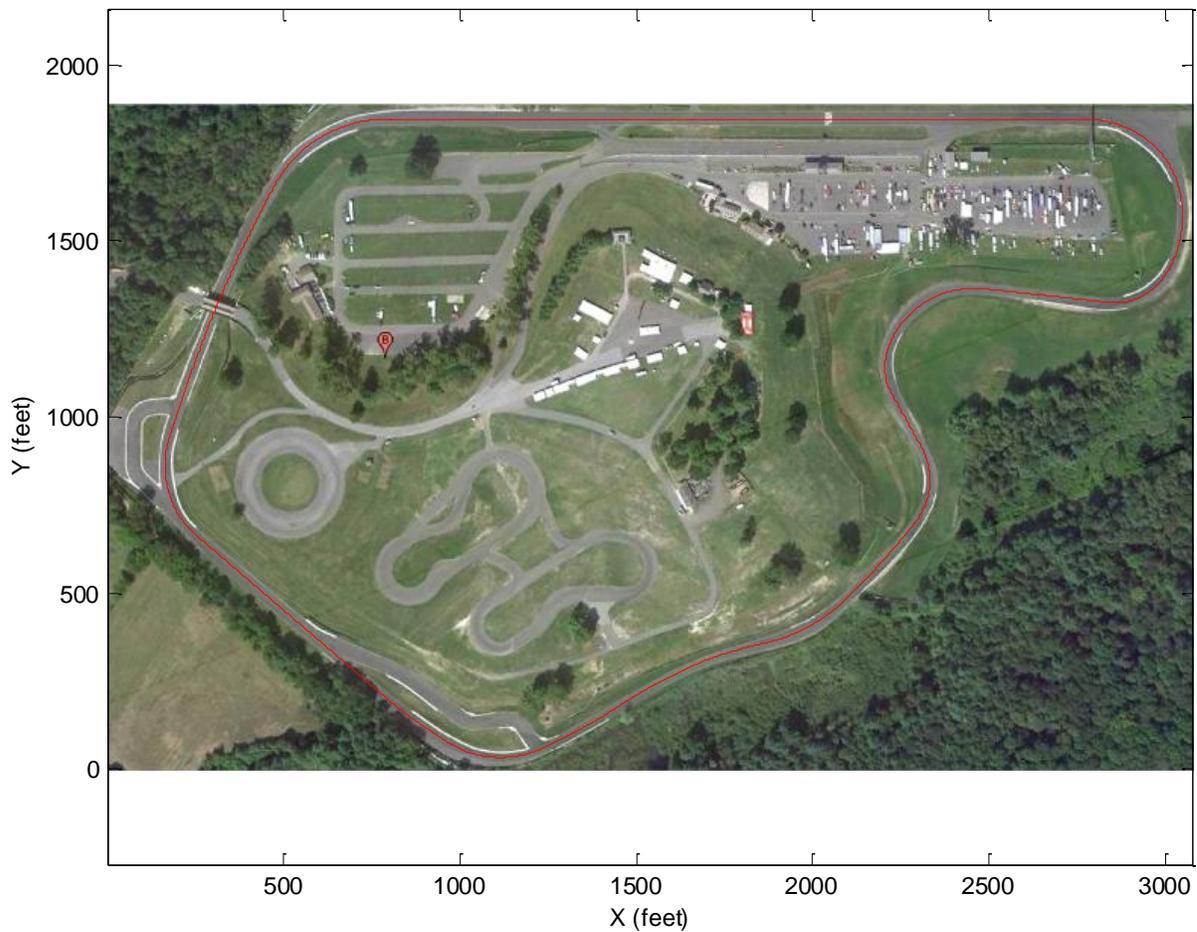


Figure 20: Limerock Vehicle CG Path

Figure adapted from [Servos](http://maps.google.com/maps?f=q&source=s_q&hl=en&geocode=&q=limerock+park&ll=37.0625,-95.677068&spn=38.911557,93.076172&ie=UTF8&hq=limerock+park&hnear=&ll=41.927921,-73.387492&spn=0.008541,0.030899&t=h&z=16), 15 April 2011 <
http://maps.google.com/maps?f=q&source=s_q&hl=en&geocode=&q=limerock+park&ll=37.0625,-95.677068&spn=38.911557,93.076172&ie=UTF8&hq=limerock+park&hnear=&ll=41.927921,-73.387492&spn=0.008541,0.030899&t=h&z=16>.

In order to make the vehicle follow the proper course assigned inputs were assigned rather than using normal operator inputs. The vehicle starts at the following coordinates (800, 1845) at a speed of 0 ft/s. The vehicle immediately begins accelerating at $0.22g$'s until it reaches a

speed of 107 ft/s. As the vehicle approaches the first turn it begins to brake at 0.125 g 's slowing to approximately 48 ft/s. The vehicle then proceeds to coast through the next four turns reaching the very bottom of the track. Once this point is reached the vehicle accelerates through the straight at approximately 0.22 g 's up to a speed of approximately 80 ft/s, and then proceeds to decelerate at approximately 0.14 g 's to a speed of approximately 55 ft/s. The vehicle once again coasts through the next two turns returning the starting position. The change in speed caused by the vehicle accelerating and decelerating can clearly be seen in Figure 21. The steady state speeds can also clearly be seen in the graph as well.

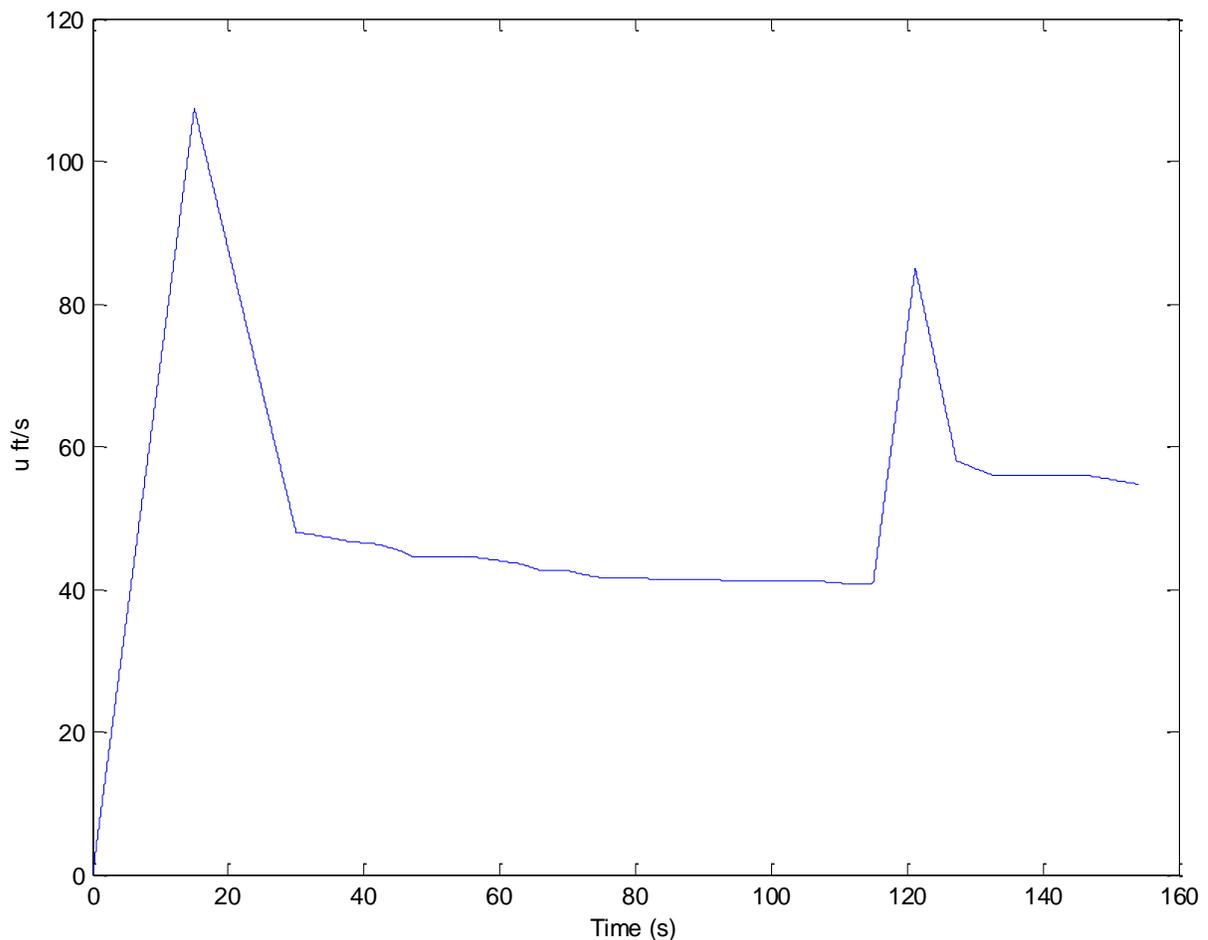


Figure 21: Longitudinal Velocity vs Time

The accelerations which cause to vehicle to speed up and slow down would all be introduced in the x-direction as seen in Figure 22. This causes the graph of longitudinal acceleration seen in Figure to have mostly steady state accelerations, with drastic changes due to the fact that throttle response was not modeled. Accelerations in this model are simply introduced by applying either a positive longitudinal force to the tires or a negative force to cause deceleration. If acceleration had been based on throttle response the acceleration would have much more fluctuation with only drastic changes caused by large braking forces.

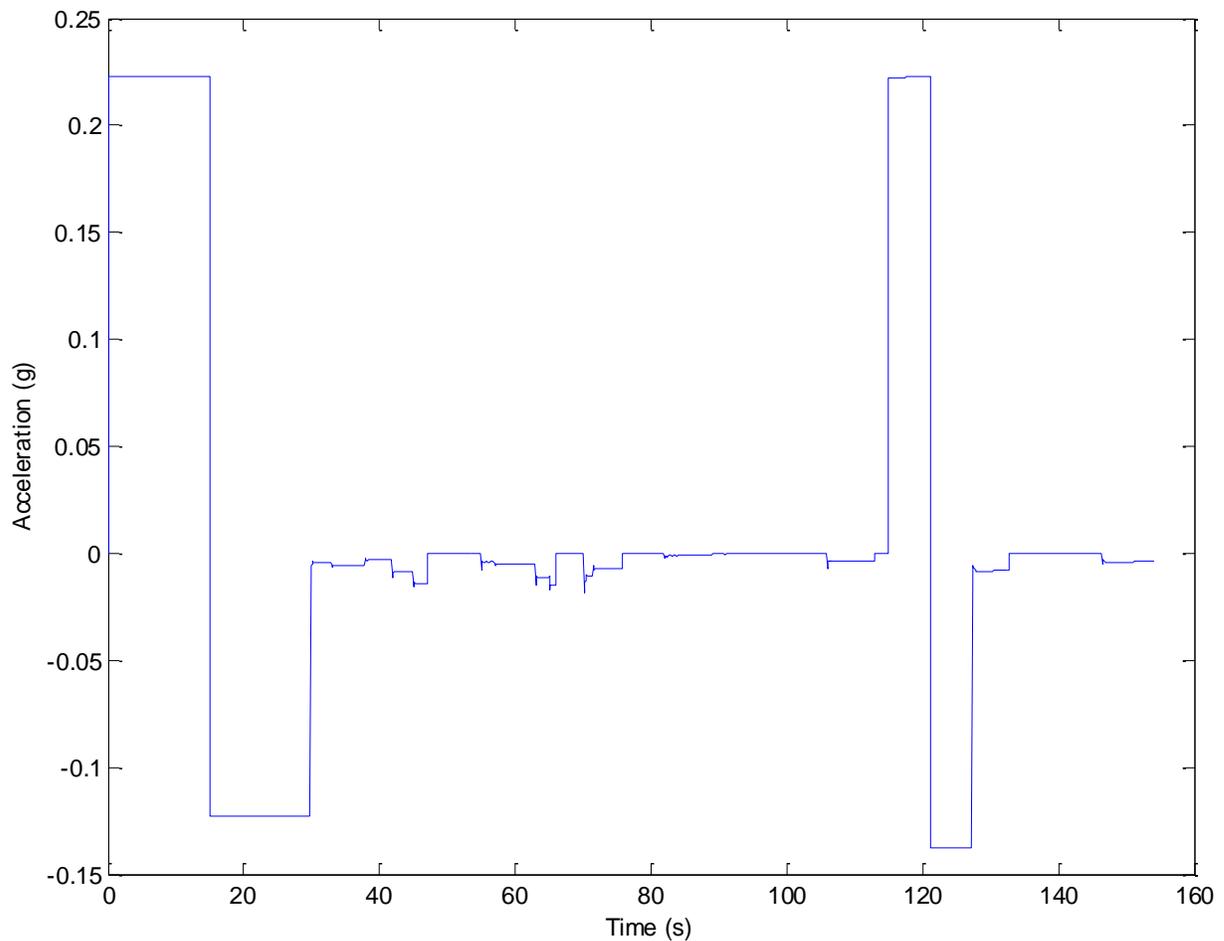


Figure 22: Longitudinal Acceleration vs Time

As the vehicle accelerates it would begin to pitch backwards which can be seen from Figure showing a theta as a value of approximately 1.25 degrees. Theta then settles to a steady state caused by the damping in the suspension. Theta reverses into the negative angle of approximately -0.75 degrees once the brakes are applied causing the vehicle to pitch forward. Once the vehicle starts decelerating the acceleration drops into the negative, and as soon as the brakes are released theta eventually reaches steady state of zero degrees as seen in Figure 22. This trend can clearly be seen throughout all of the acceleration, braking, and brake release maneuvers, which commonly happen in a track racing scenario. The same will happen in everyday normal driving cases, but to a less degree. It should be noted that these small angle motions are inconsequential when compared to the acceleration forces and the angles required to simulate them.

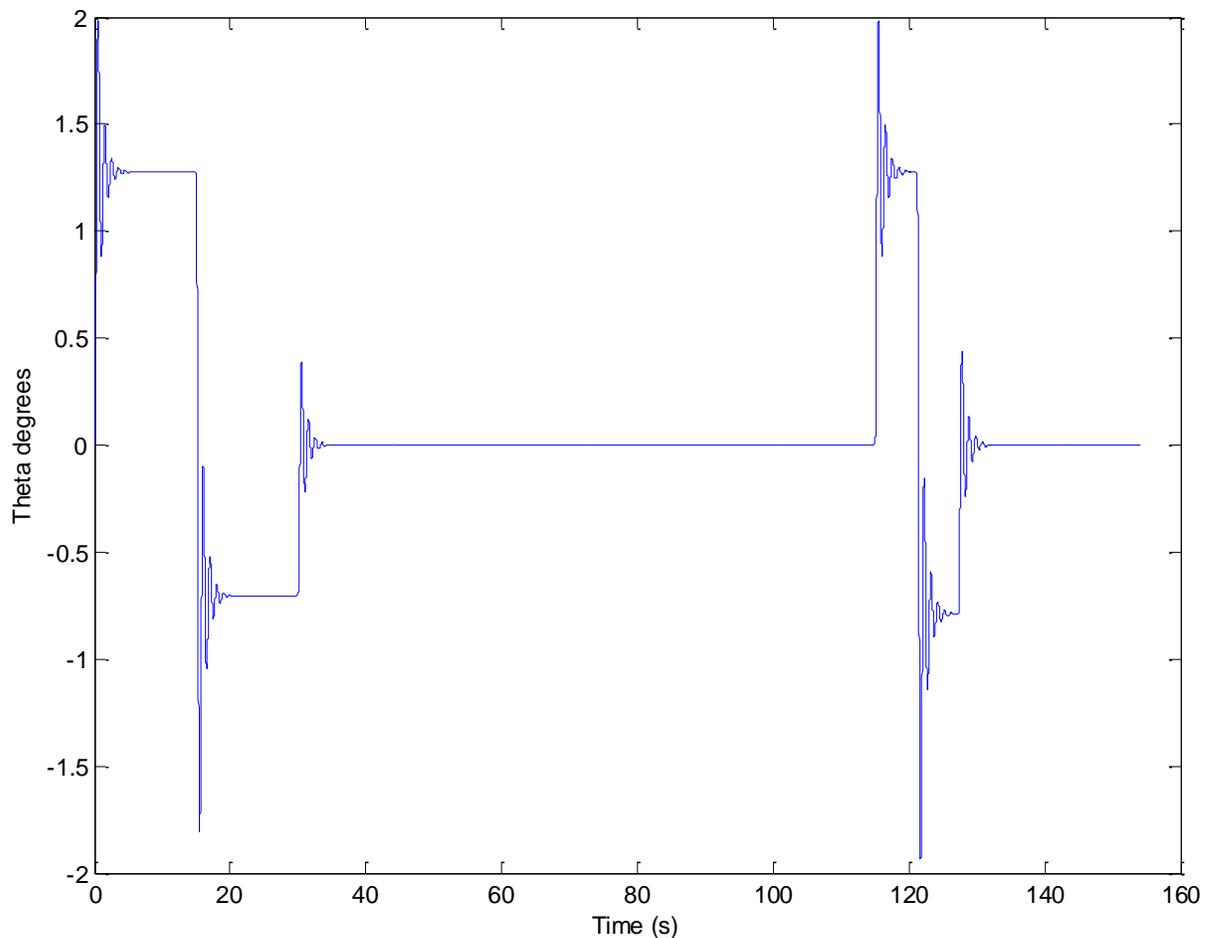


Figure 23: Pitch Angle vs Time

The vehicle begins to turn once a steering angle is introduced. Since the radius of the turns in the track does not vary drastically the delta angles are only changed slightly through the turns to keep the vehicle on the road. This results in a graph with drastic changes between steady state turn angles as can be seen in Figure 23. Also contributing to the drastic changes in the graph would be the lack of steering lag as seen in Figure 24, which would show the steer angle increasing at a slope, rather than instantaneously. Of course it does not take very long to turn the wheel to reach the required steady state steering angles, but taking the lag into account would replace the instantaneous steering angle change with large slopes.

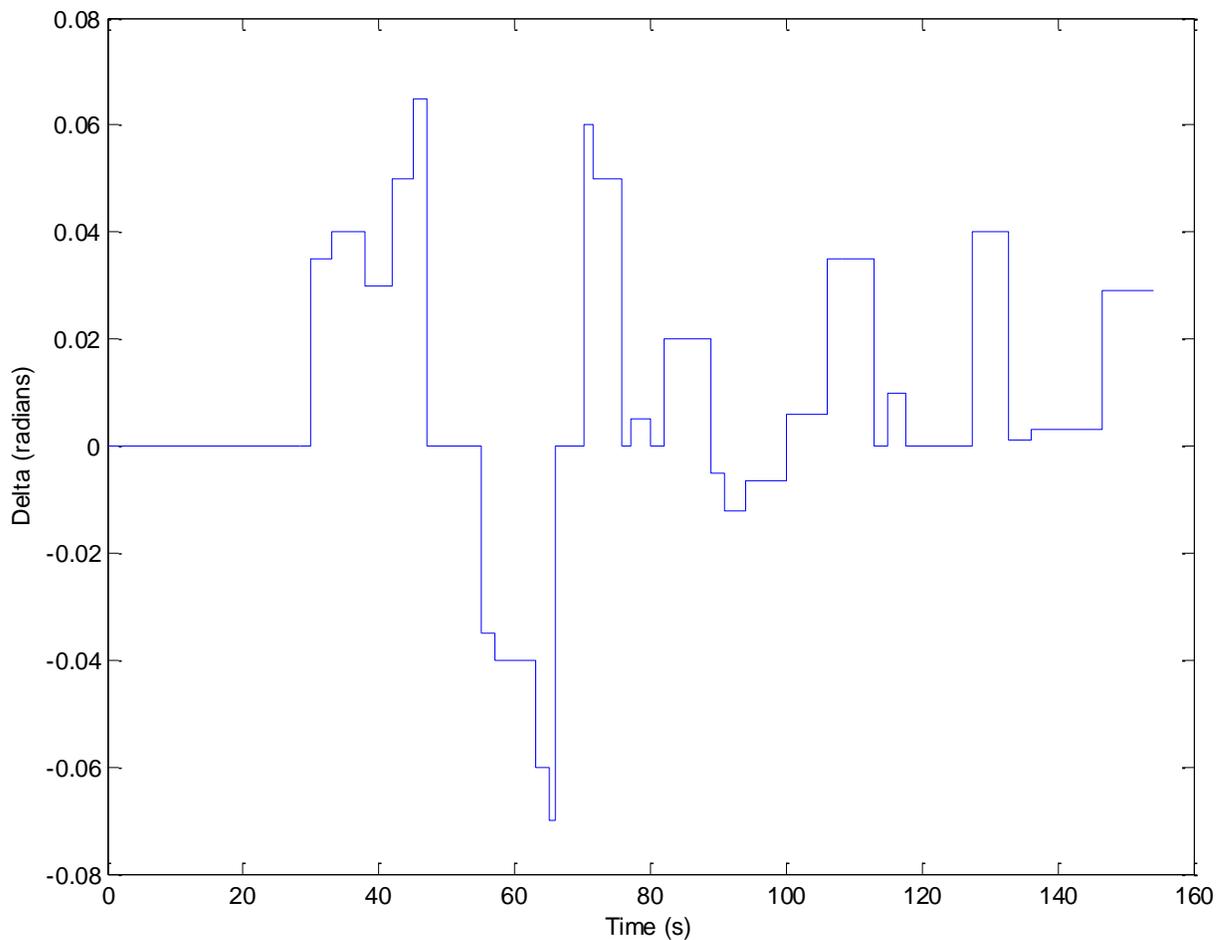


Figure 24: Steering Angle vs Time

The vehicle turning introduces acceleration into y-axis which is perpendicular to the vehicle path of travel and opposite of the centripetal force caused by the vehicle traveling about the radius as seen in Figure 26. This acceleration is developed by the tires laterally and can be seen in Figure as the green vector. This acceleration is caused by a lateral force developed by the tires is what keeps the vehicle from sliding off the road. The lateral force also causes load transfer from the right tires onto the left tires resulting in compression of the springs on the left of the car and stretching of the right springs. The lateral force caused by the tires can increase as the steering angle and speed of the vehicle increases. Increasing speed or steering angle of course can cause rollover, when the load on the vehicle is transferred from one side of the vehicle to the other. It can be clearly seen that lateral acceleration is only present while the vehicle is in the process of turning as seen in Figure 25 and 26.

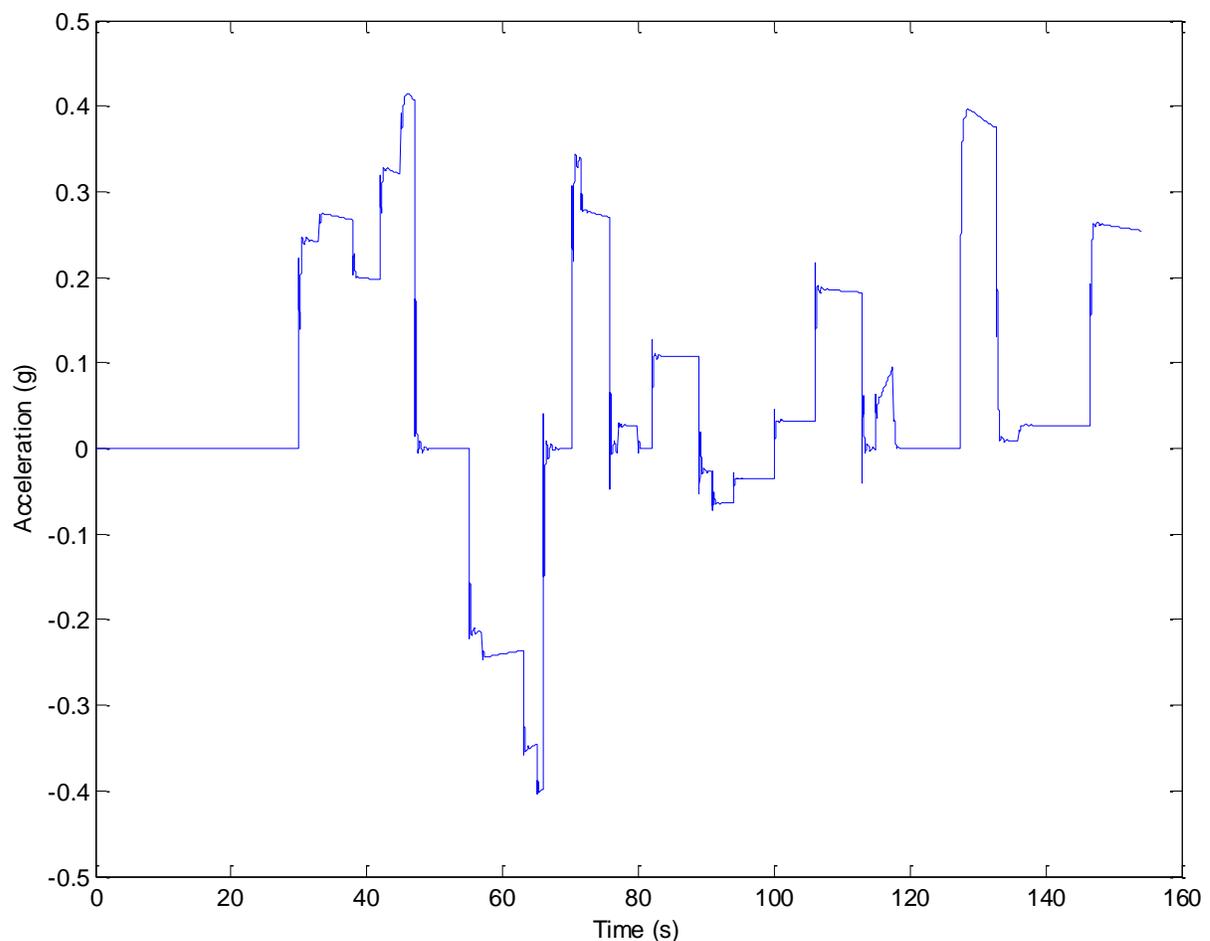


Figure 25: Lateral Acceleration vs Time

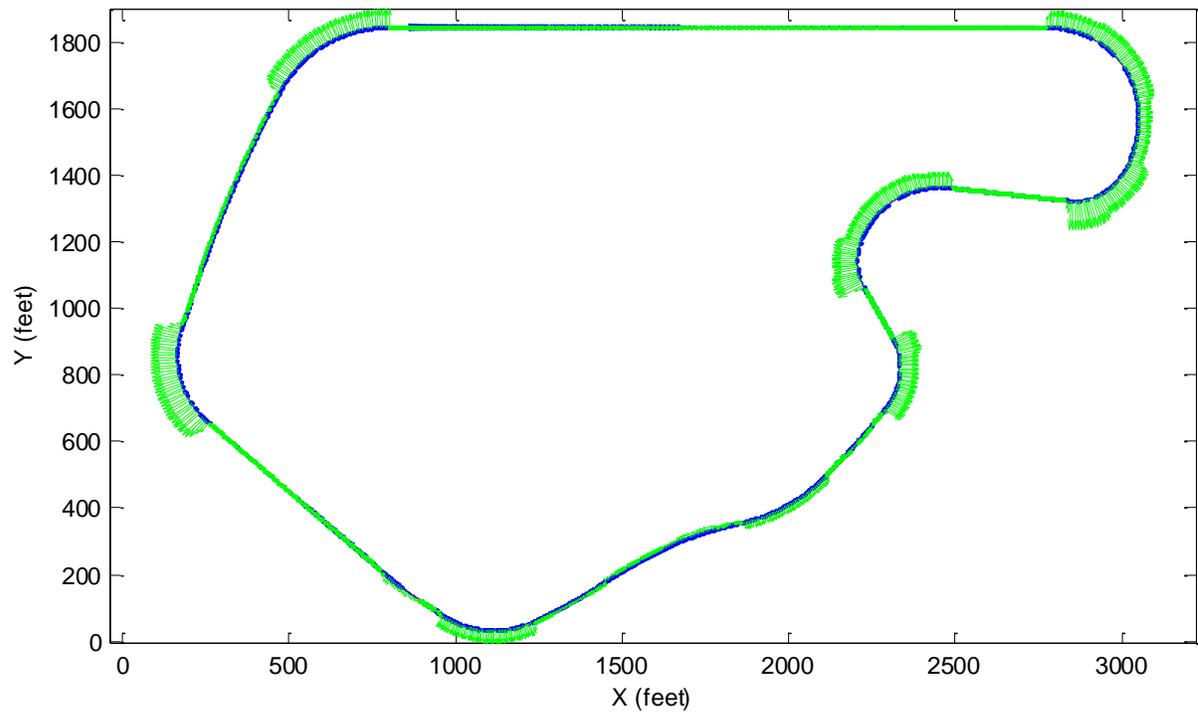


Figure 26: Plot of X and Y Accelerations on CG Path

Obviously if an acceleration or force is present laterally it will cause a velocity to develop in the y-axis as can be seen in Figure 27. These velocities will always be perpendicular to the longitudinal motion of the vehicle, since the vehicle is able to move in any direction on the x-y plane. The combination of rotation about the z-axis and the direction which the vehicle is turning directly effects whether the velocity is positive or negative why the velocity fluctuates so rapidly.

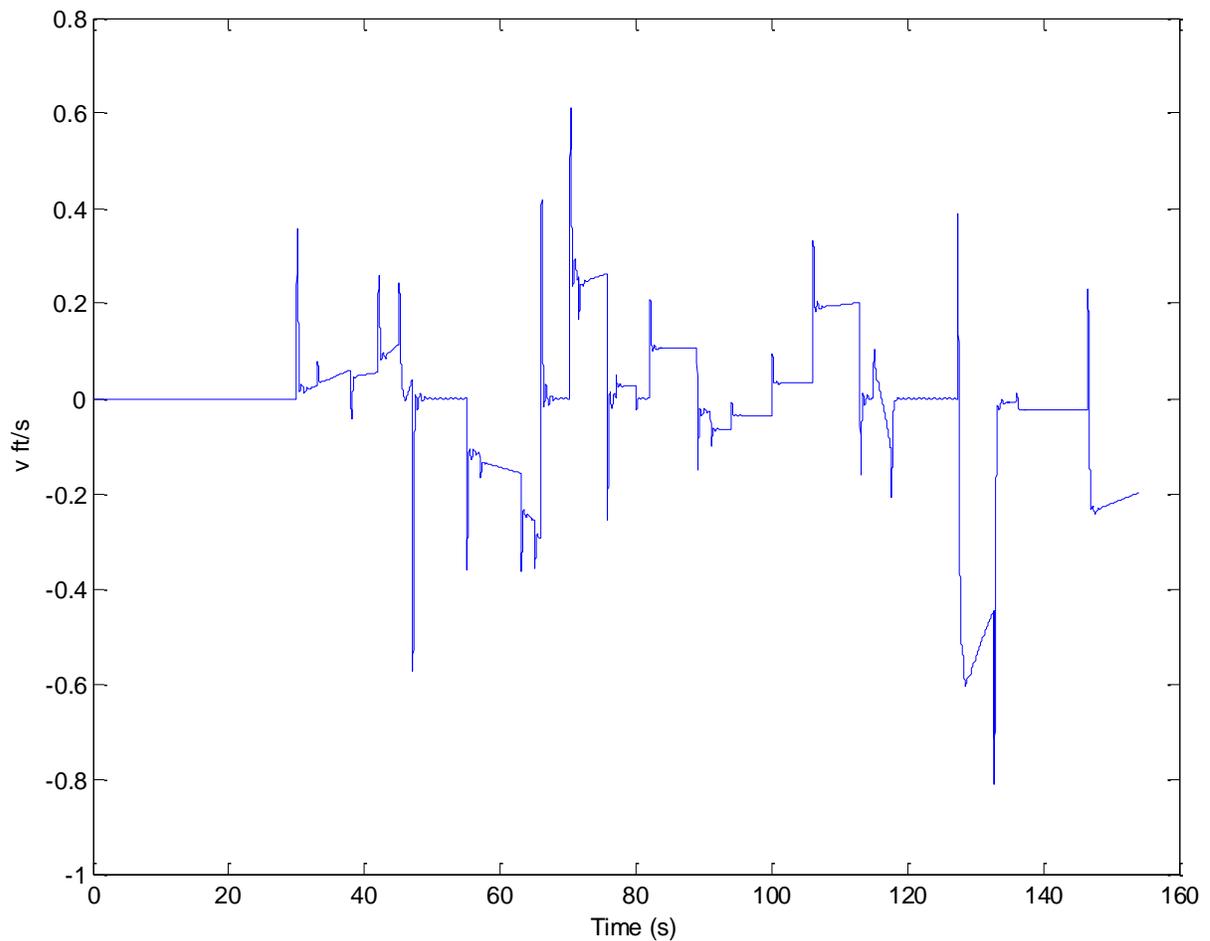


Figure 27: Lateral Velocity vs Time

However when the overall speed of the vehicle takes into account the velocity about both the x and y axes the rapid fluctuation has no noticeable effect on the plot of speed versus time as seen in Figure 28. It is easy to see that the magnitude of the lateral velocity is small compared to the longitudinal velocity. The only time that the lateral velocity would reach any significant value would be if the vehicle began to skid out of control and travel laterally.

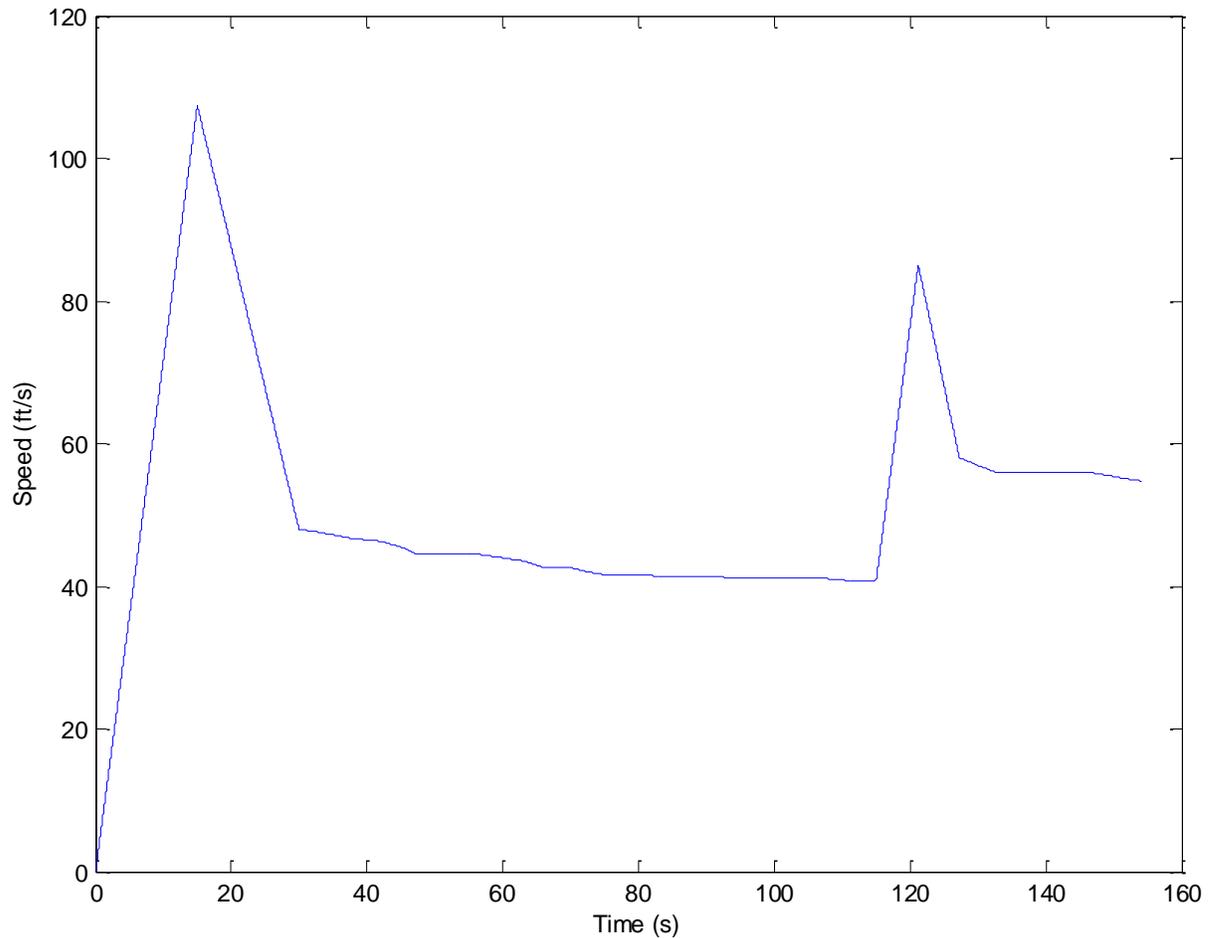


Figure 28: Speed vs Time

Referring back to the Figure 24 the plot of steering angle clearly has a direct effect on the yaw rate of the vehicle during the simulation. As the vehicle enters the first two turns the yaw rate increase drastically drops to zero and then increases drastically in the opposite direction as seen in Figure 29. This clearly represents the first two turns on the track. The yaw rate represents how fast the vehicle is rotating about the z-axis, which would be the highest during the tightest turns on the track. The heading angle can also be verified by examining the direction, which the car would have to travel. Clearly the vehicle starts by traveling east and begins to turn to the right. This causes the angle to remain at zero until it starts turning, which causes the angle to increase. The angle then decreases during the second turn. Of course the rest of the turns on the track are to the right, therefore the heading angle continues

to increase towards the 365 mark as seen in Figure 30, as the vehicle approaches the end of the track where the vehicle once again is facing east as seen in Figure 20.

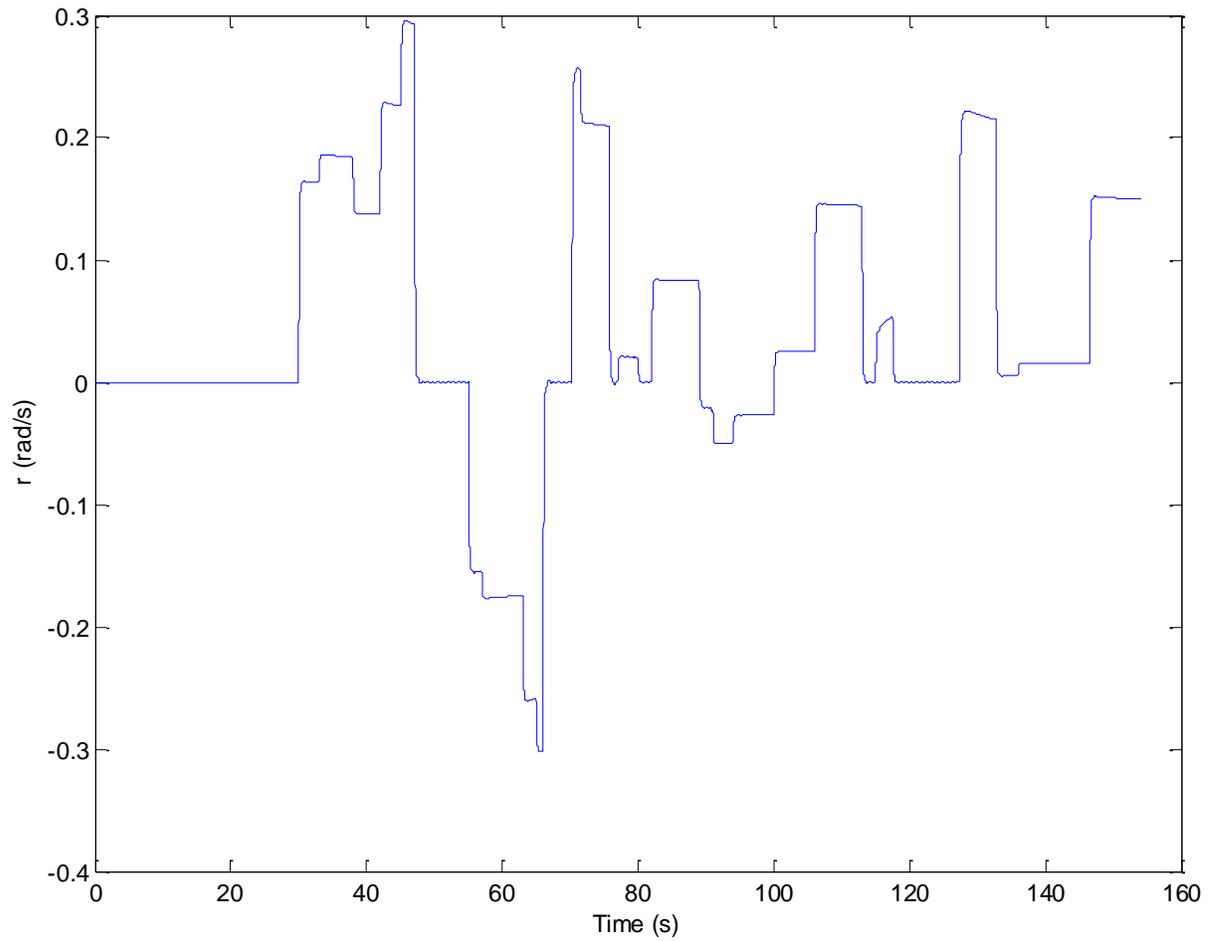


Figure 29: Yaw Rate vs Time

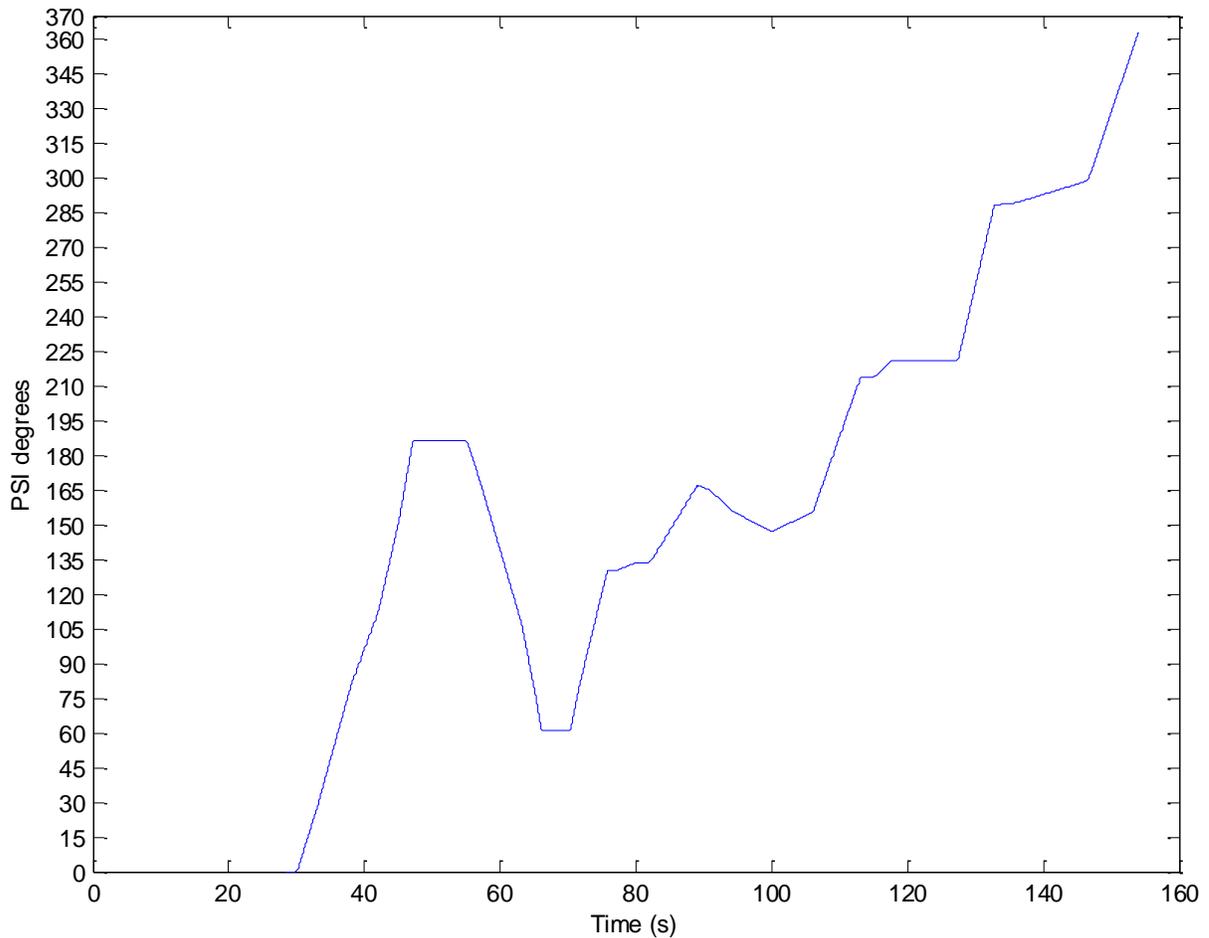


Figure 30: Heading Angle vs Time

The angle at which the vehicle rolls about the y-axis can be verified by referring to the lateral acceleration, lateral velocity, or simply by examining the racetrack. When a car is driven through a long turn such as a highway entrance or exit ramp the driver will have the feeling of being pushed away from the direction which the car is turning. This is caused by the lateral force pushing in the opposite direction of the turn. As mentioned before this causes the vehicle load to shift away from the direction which the car is turning as well. Since this causes the load to shift and the springs to either compress or stretch depending on which way the load is being transferred it will cause the vehicle to rotate about the y-axis at some angle, which is referred to as ϕ . It is easy to see that since the vehicle is traveling straight at first that there would be no angle since there is no load shift. Once the vehicle enters the first turn

load shift begins in the same direction as the lateral force, causing the vehicle to rotate in the direction of the load shift. For example since in the first turn the lateral force is to the left in a right turn the lateral force would be negative causing a negative roll angle as seen in Figure 31. This would mean that the load would transfer from the right tires on to the left tires, which would cause the left side of the vehicle to be lower and the right side of the vehicle to rise. When the vehicle turns to the left the roll angle becomes positive and the load shifts in the opposite direction as can be seen in the second turn of the track. Following the rest of the graph, it is clear that all of the changes in roll angle correspond with a turn in the track.

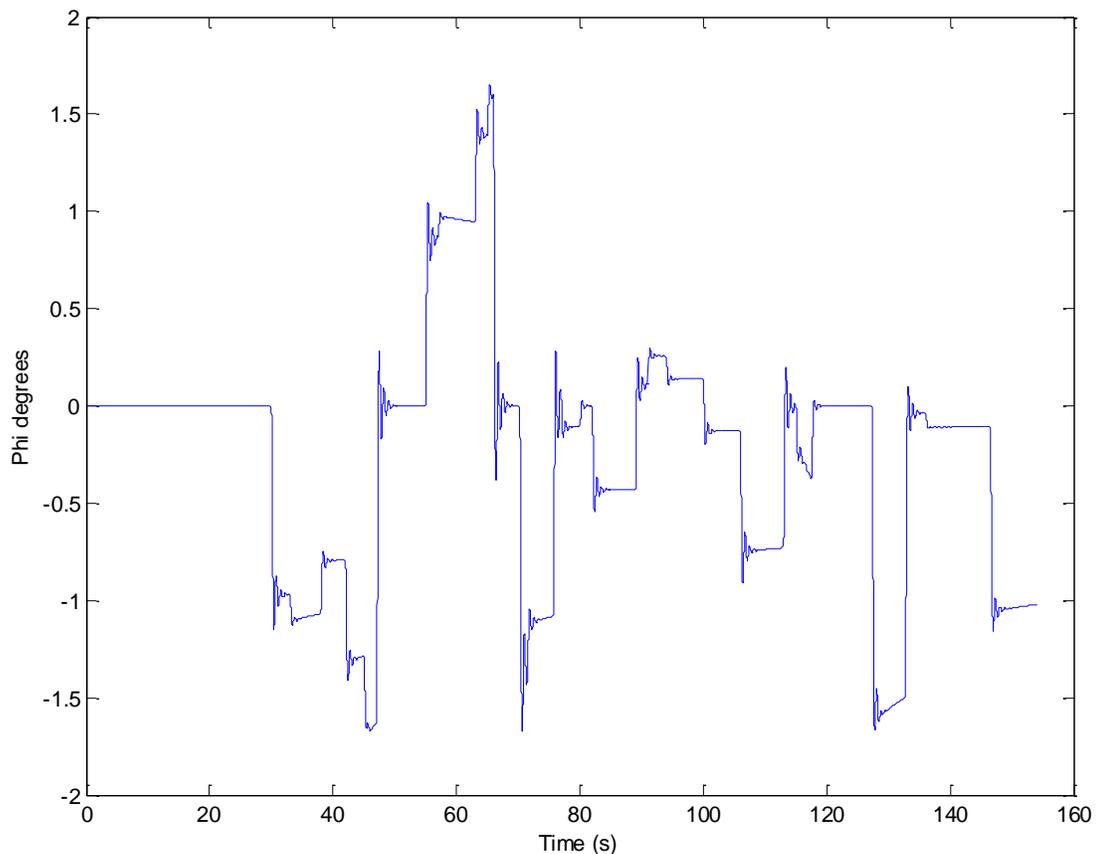


Figure 31: Roll Angle vs Time

Once the washout algorithms are properly tuned for the given simulation the three-degree of freedom and six-degree of freedom systems can be compared. This can be done by taking a

snapshot of a particular maneuver. Given that when the vehicle enters the first turn on the track the brakes were released the longitudinal force would be minimal only the lateral force is examined to compare the abilities of the two systems. First of all comparing the solid line to the graphs of lateral acceleration of the two motion base in Figure 23 shows that the two graphs are similar verifying the algorithms are working properly. The short dashed line represents the three-degree of freedom or “rotation only” motion base. The varied dashed line represents the six-degree of freedom motion base. It is difficult to see the lines in this view seen in Figure 32, but it is clear that both of the lines follow the acceleration profile correctly only straying from the path during changes in the acceleration, which obviously happen during changes in steering angles since there is no longitudinal acceleration applied through the turns.

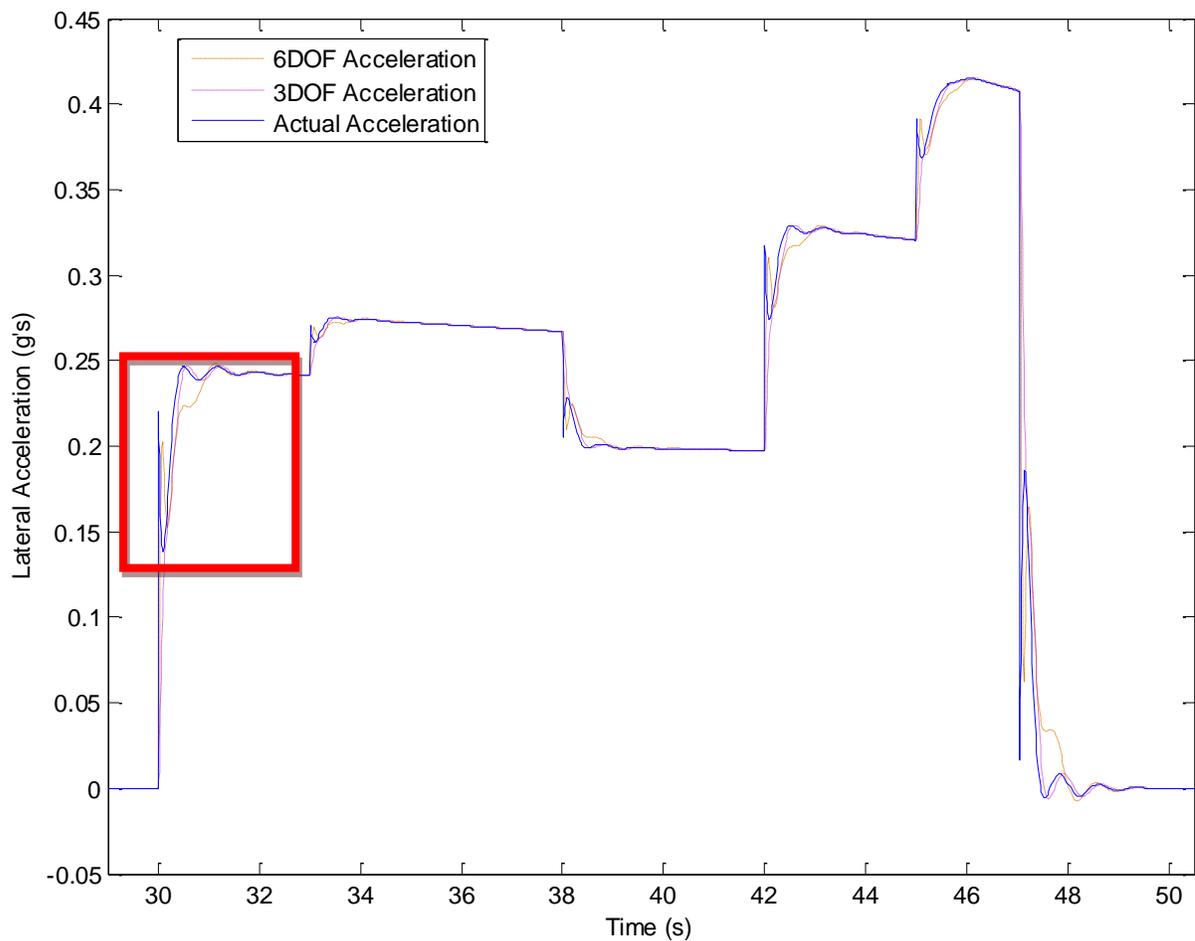


Figure 32: Lateral Accelerations of Simulation, 3DOF, and 6DOF Motion Base

To form a better opinion on which base appears to be more accurate closer snapshots will be necessary. The first part of the graph as seen in Figure 33 shows the instant the steer angle is input. At this point the lateral acceleration first appears and clearly both systems are not able to precisely represent the required acceleration. The first problem noticed is that at between 30-30.2 seconds the rotation only system is unable to replicate the first wave of acceleration. This system simply misses that wave and keeps increasing the amplitude so that it matches the actual acceleration at approximately 30.5 seconds. The six-degree of freedom system however is able to make an attempt at replicating the first wave, but with approximately 0.07s of lag and misses the amplitude of the crest by approximately 0.061g's. The situation is similar at the trough, the six-degree of freedom system is approximately 0.05g's above the actual acceleration of the car in the trough with a lag of 0.095s. Taking a look at the crest of the second wave the rotation only motion base is more accurate, having an acceleration difference of only 0.0036g's at the top of the crest, with only a lag of approximately 0.067s. The six-degree of freedom system is less accurate at this point with the crest of the wave 0.081g's and a lag of 0.11s. Looking at the trough of the second wave the rotation only is almost identical in amplitude with a difference of 0.0013g's but with a lag of approximately 0.0625s. The six-degree of freedom system is approximately off by a lag of 0.22s and 0.056g's below the actual acceleration. By the time both systems are to the crest of the third wave the acceleration differences are almost negligible. The rotation only system is only off by an acceleration of 0.0005g and a lag of 0.1s. The amplitude of the six-degree of freedom system is off by 0.0052g's and the lag is approximately 0.03s.

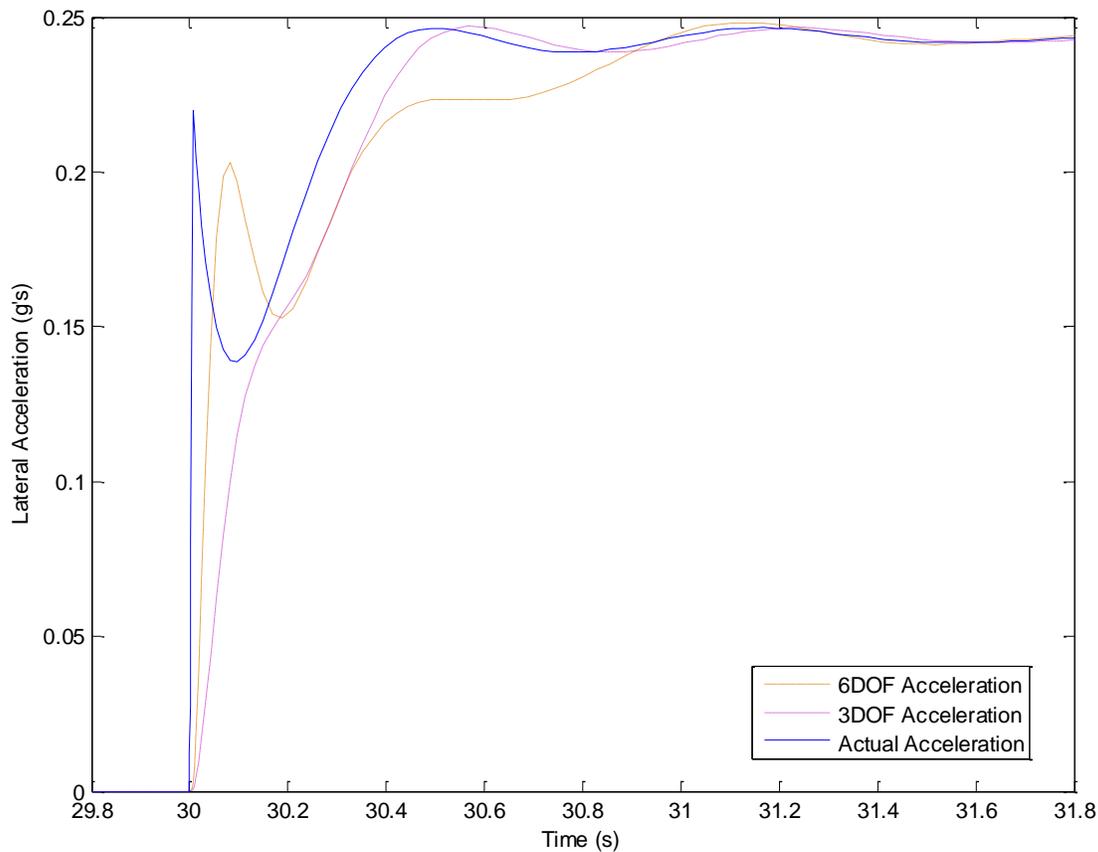


Figure 33: Lateral Accelerations of Turn 1

Using these values the peak or max error can be determined by simply dividing the difference in peak amplitude by the actual acceleration. This resulted in the three degree of freedom system having better performance after the start of the third peak. However the six degree of freedom system performed with a more consistent error throughout the first turn, and made an attempt to recreate the acceleration of the first wave. This attempt to recreate the first wave, appears to be the reason why the six degree of freedom system takes longer to match the wave amplitudes. The rotation only base is able to match amplitudes after the first wave, because the change in acceleration required is minimal. Another reason appears to be that the system did not attempt the decrease the acceleration in the manner that the six degree of freedom system did, therefore it took less time to reach the required magnitude. This allowed this motion base to make the minimal acceleration change required for the second wave, while the six degree of freedom system attempted to maintain a minimal lag and

decreased the magnitude to maintain proper wave structure. Since it maintained the wave structure, it required more time to reach the required magnitude.

	Peak 1 Error	Peak 2 Error	Peak 3 Error	Peak 4 Error	Peak 5 Error
Actual	0	0	0	0	0
3 DOF	Missing Cue	Missing Cue	0.39	0.147	0.055
6 DOF	7.54	9.84	8.93	6.32	0.57

This turn can be examined using another method to describe the error. Utilizing the root mean square deviation the deviation in acceleration can be determined at any time in the simulation. The following equation is used in order to calculate this deviation:

$$\sqrt{\frac{\sum_{i=1}^n (a_{base} - a_{sim})^2}{n}}$$

Using this equation yields the graph depicted in Figure 34 for the first turn. It is clear that the deviation from the actual acceleration is highest when the vehicle first enters the first turn. The varied dashed line which represents the six-degree of freedom system has less error for the first 0.1s, but the short dashed line, which represents the rotation only system has less error during the next 0.1s. The six-degree of freedom system is more precise initially, because of the system's ability to replicate the first wave on Figure as mentioned before. The rotation only system eventually builds up magnitude and reaches the actual acceleration at approximately 30.55 seconds, which explains why the error approaches zero at this point in Figure 34. It takes the six-degree of freedom system until approximately 30.9 seconds to reach the magnitude of the actual acceleration, which causes the rotation only system to perform more precise from 30.1-30.9 seconds. However after this point the six-degree of freedom system is more precise. Review of this graph shows the rotation only system appears to have less deviation from the actual acceleration overall. However review of Figure 33 shows that the rotation only system is unable to replicate the initial acceleration in the turn,

therefore provide missing cues in the simulation. The rotation only system is able to reach the maximum magnitude quicker, which causes the RMS error to be larger for the six-degree of freedom system, which takes longer to reach this magnitude. This is verified by calculating the area under the two curves to get a definitive RMS value. The six degree of freedom system has a value of 0.68, while the three degree of freedom system has a value of 0.36. The three degree of freedom system therefore appears more accurate, however the missed cues are what enables the system to have the better RMS values. These missed cues are undesirable and make this system the less accurate one, even though the RMS values show otherwise. This information will need to be used to compare the tradeoffs of each system in order to determine which system is more suitable for specific applications.

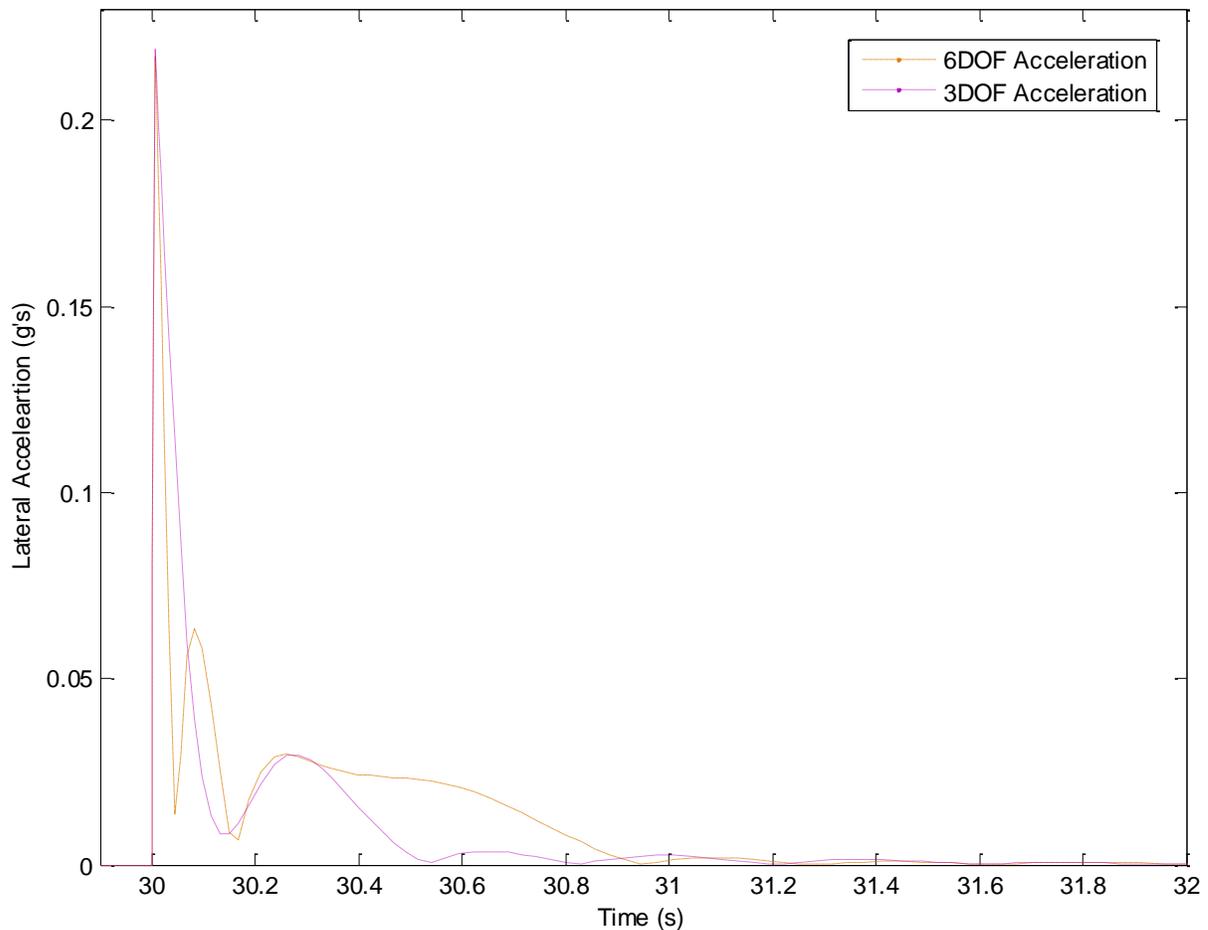


Figure 34: Root Mean Square Error Turn 1

Next the longitudinal acceleration is examined. Examination of the longitudinal acceleration through the whole track reveals that the six-degree of freedom system is more accurate. Most notable in the graph in Figure 35 is that the three-degree of freedom system is overshooting the steady state accelerations creating false cues.

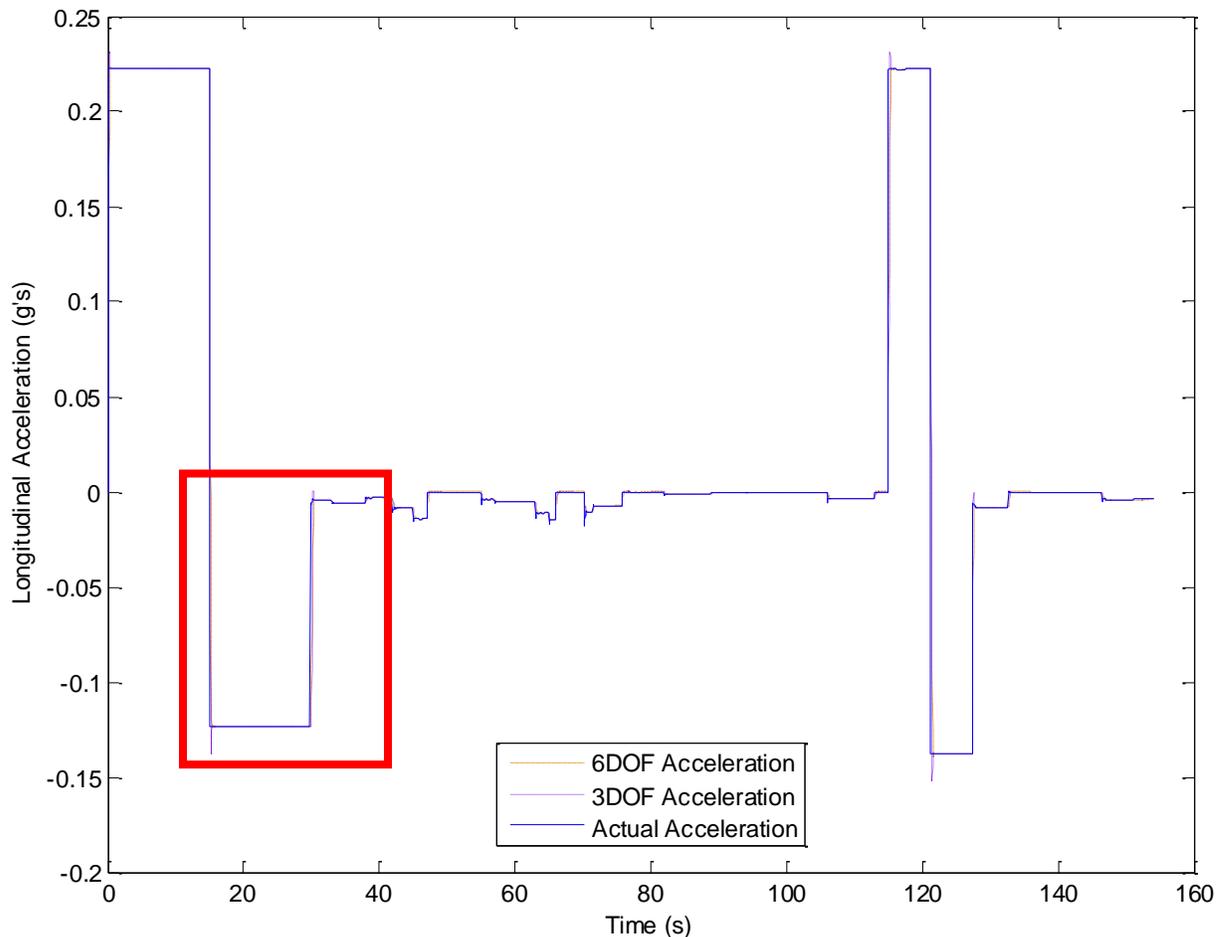


Figure 35: Longitudinal Accelerations of Simulation, 3DOF, and 6DOF Motion Base

The following maneuver is chosen for examination: the initial braking after accelerating through the first straight followed by braking, releasing the brake while entering the first turn. In this view very small false cues can be seen even for the six-degree of freedom

system, but they are very small in magnitude and should have no effect on the feel of the maneuver.

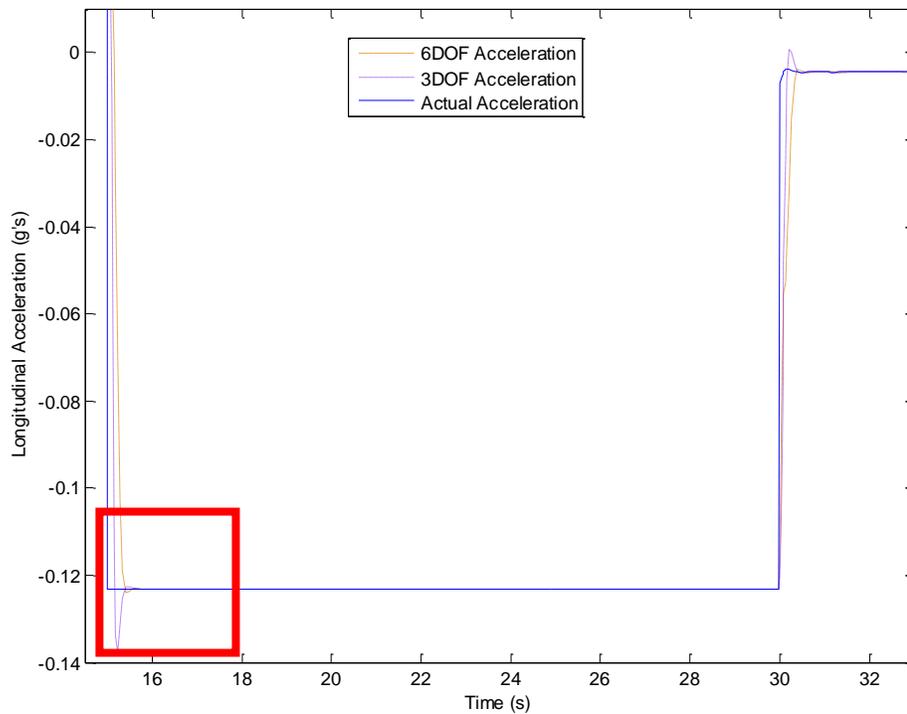


Figure 36: Longitudinal Acceleration Braking, Coast, and Entering Turn 1

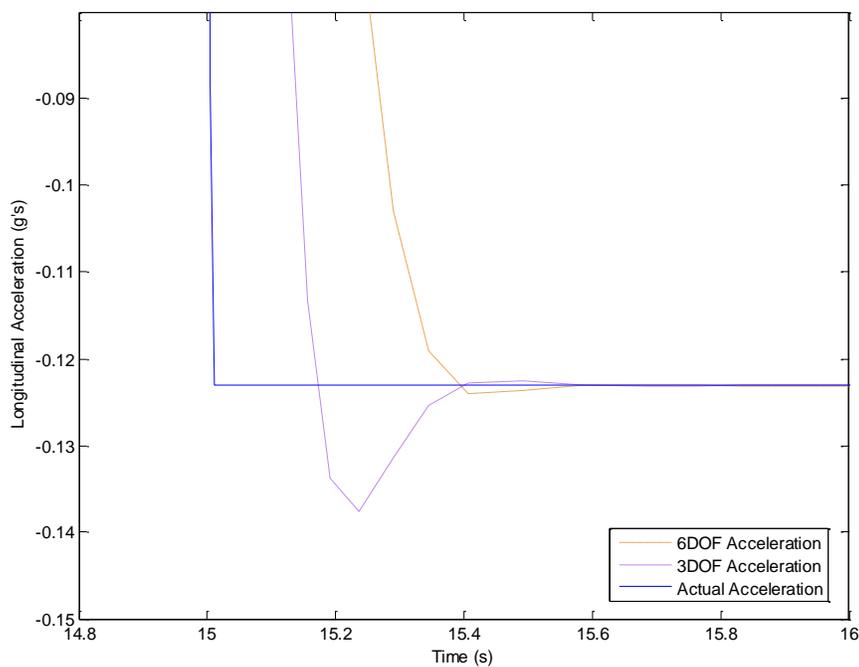


Figure 37: Longitudinal Acceleration Braking, Coast, and Entering Turn 1 Close Up

Examination of the root mean square error for this maneuver depicted in Figure 37, reveals that at first glance the three degree of freedom motion base appears to be more accurate. Solving for the area under each graph reveals that the six degree of freedom system has a RMS value of 0.175 versus a 0.126 for the three degree of freedom system. The three degree of freedom has less error because the system decelerates at a faster rate, which is what causes the larger false cue. Unfortunately the root mean square error disguises the false cues when the magnitude is squared, which changes the sign making the base appear more accurate. A quick glance at the Figure 36 reveals that in fact the three-degree of freedom system is more precise prior to the false cues, but because the acceleration magnitude changes so rapidly it overshoots the magnitude of the actual acceleration. This causes the three-degree of freedom motion base to correct quickly by changing the magnitude of acceleration quickly creating an undesirable false sensation or cue.

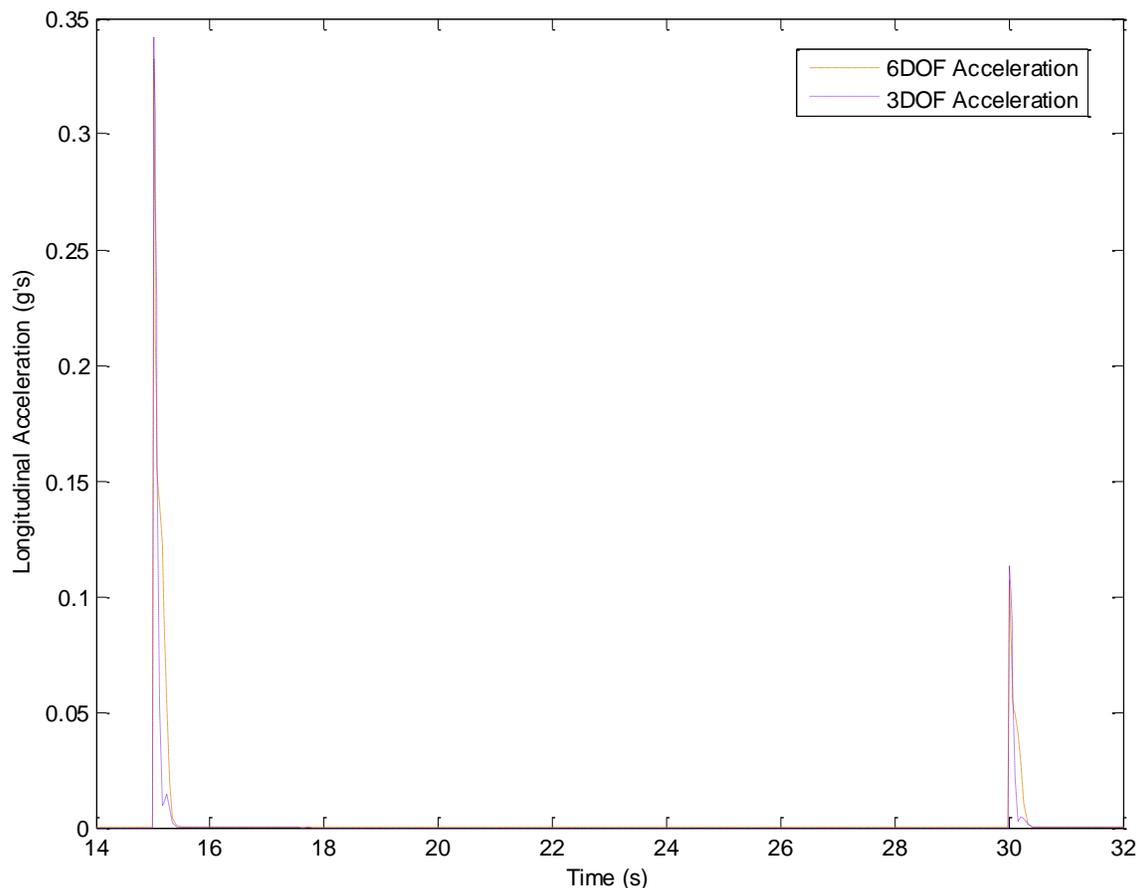


Figure 38: Root Mean Square Error Braking, Releasing Brake, and Entering Turn 1

Chapter 6: Future Work

Vehicle simulation and motion base have many practical applications. This vehicle simulation and its associated washout algorithm can be adapted for use in all of these applications. One of the obvious applications is for a real-time full scale vehicle simulation. In order to make the program more realistic several changes would need to be made to the vehicle simulation program. This program could be modified to use a full scale steering wheel to input the steer angle rather than inputting it manually. The program also could be adapted to use a brake, gas, and clutch pedals. The brake would measure what percentage of the brake is depressed in order to accurately calculate the brake force applied. Similarly the percentage of the gas pedal is depressed can be measured in order to calculate the engines throttle response. To do this research would need to be done in order to relate pedal depression, engine speed, torque, and horsepower in order to accurately model acceleration based on pedal depression. An example of a chassis dynamometer, which could be used in research, is shown in Figure 38.

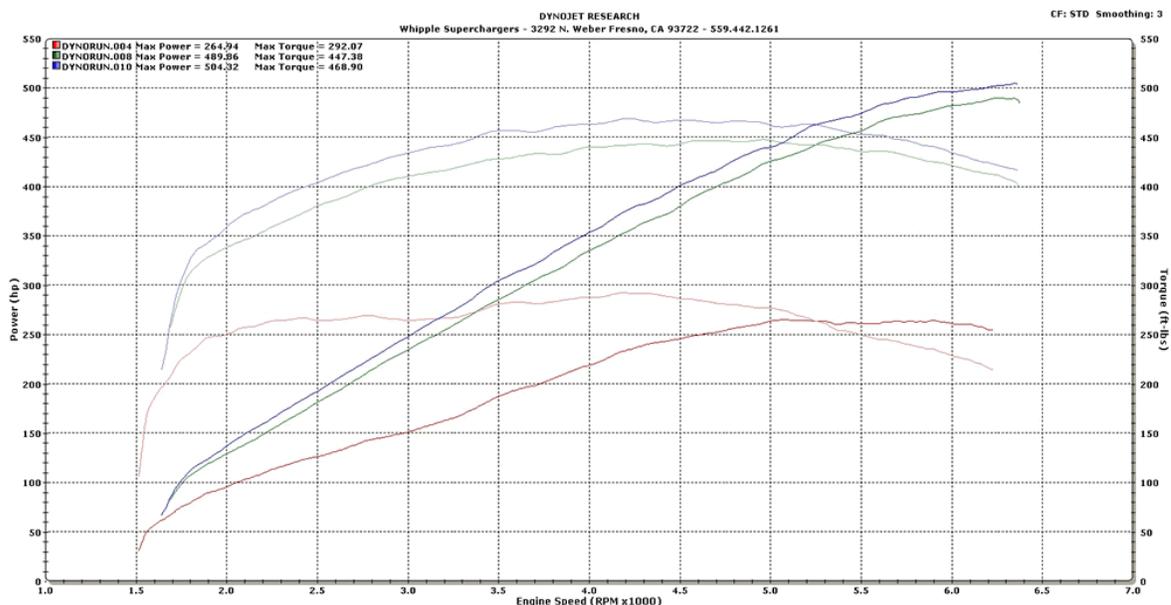


Figure 39: Dynamometer Printout

Figure reprinted from Whipple Superchargers. 15 April 2011 <
http://www.whipplesuperchargers.com/Images/productimages/05Mustang_Dyno_Large.gif>.

In addition to using torque and horsepower curves to calculate acceleration the number of gears and the ratio will need to be considered. Analysis of the gear ratios will more accurately represent the acceleration and velocity of vehicle in a real-time simulation. In order to properly model the clutch research will need to be done to understand the relationship between, clutch depression, engine revolutions, and gear sizes.

These simulations can be used in both commercial and personal applications. For instance this washout algorithm program and motion base could be adapted for use in current driving or racing video games. The vehicle simulation, motion base, and washout algorithms could be used specifically for driving studies for multiple industries. Automotive manufactures could use the program to compare vehicle performance by altering certain vehicle specifications. The simulator could be used for training new drivers before they actually drive on the road or for training commercial drivers to handle new vehicles. Insurance companies and vehicle accident investigators could use the software to conduct research into driver reaction times. The industry of accident reconstruction relies heavily on using reaction times in order to determine what happened in a vehicle accident.

Several reaction times could be studied using a real-time vehicle simulator. The simulator could be used to measure the reaction time to steer to avoid and how long it takes for a driver to turn the wheel to reach the desired steering angle. The following reaction times could also be measured:

- Remove foot from gas pedal to brake
- Remove foot from clutch
- Shift gears
- React to a hazard
- React to a hazard at night

In addition to these studies the program could potentially be used in a specific case. A program could be created to simulate the motion of all of the vehicles involved except for the vehicle being driven in real-time. This would enable a driver to be put in all of the proposed

accident situations by all of the expert witnesses. The driver could be tested in this situation to see if it is possible for the driver to react and prevent the collision.

Real-time simulations could also be used to calculate post collision dynamics in accident situations. There are currently vehicle accident simulations, but none of them have real-time pre-collision simulations. This could be done by using basic principles used in the field of accident reconstruction. The simulation program could be adapted to use both conservation of momentum and conservation of energy to calculate post collision velocities. Knowing the post collision velocities the change in velocity could easily be determined. Algorithms also would need to be developed in order to calculate the principle direction of force. These equations would be used to determine whether the collision would be centric or eccentric in nature. Based on this moments could be calculated and change in velocity can also be determined. Once the post collision velocity, principle degree of force, and moments are calculated these values could be input into the vehicle simulation allowing the vehicle to continue to final rest using the original dynamics program.

Another potential research area would be using the motion base for study occupant dynamics in vehicle crashes. This could be done by using a car seat of a similar configuration and attaching the chair to the motion base. A crash test dummy could then be placed in the seat and buckled in. The crash test dummy data could then be downloaded from accelerometers built into the test dummy in order to find what accelerations and forces were experienced during the crash.

Chapter 7: Conclusion

The vehicle simulation was developed, implemented, and tested. The vehicle simulation was ran and verified to be free of errors. The x position, y position, yaw, pitch, and roll and steer angle were all at the proper magnitude. The longitudinal velocity, lateral velocity, longitudinal acceleration, and lateral acceleration graphs all were determined to be of proper magnitude and direction.

The washout algorithms were developed, implemented and tested. The lateral and longitudinal accelerations were then imported into the Simulink program, which yielded the two motion bases ability to simulate acceleration. The accelerations were checked after being filtered through both the high pass and low pass filters and were determined to be working correctly.

Review of the peak error determined that the three-degree of freedom system tended to reach and match the magnitude of the actual acceleration and maintain it as long as there were not any high frequency accelerations. Review of the root mean square deviation show a spike in error during the transition between accelerations, which is a sign that a missing cue is present that was caused by changing steering inputs and applications of brakes or applications of the throttle. However since the steering inputs, brake, and throttle applications do not happen frequently the missing cues only happened for a short time, which allowed the motion base to match the magnitude of the actual acceleration. It was determined that since the three-degree of motion base can only use rotation that it is only suitable for low frequency maneuvers, and steady state accelerations. This motion base consistently has missing cues when high frequency accelerations are present, but tends to match the magnitude of the actual acceleration quicker than the six-degree of freedom motion base. The reason the magnitude is reached faster was that the magnitude of acceleration is constantly rising and not following the high frequency wave that is the missing cue. The actual acceleration during the time the missing cue consisted of a full wave, which caused a rapid drop in acceleration followed by a rapid ascent.

Review of the peak error of the six degree of freedom system determined that there were not any missing cues during any maneuvers. The peak error for this system is more consistent than in the three-degree of freedom system. It was determined it was more consistent because there was no missing cues, and that the motion base was able to move laterally to produce more acceleration at a faster rate. Looking at the root mean square error of the six-degree of freedom motion base determined that this motion base more accurately represented the overall acceleration profile. The main problem with this system was that there was lag in the response to changes in acceleration due to the fact that the system is unable to increase the magnitude of the acceleration fast enough. The lag however does decrease quickly as the system does not wait until it hits the magnitude of the actual acceleration. The lateral movement ability of this system is what allows this system to make the rapid changes in acceleration. The ability to make the rapid changes in acceleration required in a high frequency acceleration response was determined to be why there were no missing cues in this motion base.

Having reviewed the error of both systems it was determined that the overall errors of each system over a single maneuver are close. However the six-degree of freedom motion base has less lag and does not have any missing cues. This system more faithfully represents the acceleration profile, even though the magnitudes eventually are matched better by the three-degree of freedom system.

Considering the previous findings applications can be determined for each of the motion bases. The three-degree of freedom system would be more suitable for applications, which do not require precise representation of the actual accelerations. Applications such as personal entertainment and new driver training may not need systems that simulate all motion cues. However training for professional drivers such as commercial vehicle operators, racing drivers, police, and other emergency personnel may require the six-degree of freedom systems. Operation of these vehicles may require the driver to notice certain motion cues, which can be the beginning of the vehicle entering the brink of losing control. Applications,

which absolutely would require the more accurate motion base would be for automotive design, court case simulations, and virtual vehicle crash tests. If a vehicle was designed using the three degree of freedom system missing cues could cause some undesirable ride characteristics, which would be high frequency and simply would not be duplicated by this system. Lawyers in a court case would quickly see the overlooked missing cues, which would make the evidence inadmissible. Virtual crash tests may involve finding the accelerations, which a passenger in the vehicle would feel. If a three-degree of freedom system was used once again the high frequency accelerations would not be produced. This would prevent the high frequency from being recorded, which could prevent the magnitude acceleration from reaching a injury threshold that a six-degree of freedom system could have made.

In conclusion both motion base systems have the ability to represent the acceleration of a vehicle. If cost is an issue and the motion base is not being used for research or in a profession requiring precision the three-degree of freedom system is the best choice. However if precision is required than the six-degree of freedom motion base is the best choice

Bibliography

Bernard, James E. Class Lectures. Vehicle Dynamics. Iowa State University, Ames, IA. Aug 2009-Dec 2009.

Covert, Adrian. "Inside the World's Most Advanced Driving Simulator." Popsci. 29 July 2009. 2 March 2011 <<http://www.popsci.com/cars/article/2009-07/inside-worlds-most-advanced-driving-simulator>>.

Gillespie, Thomas D. Fundamentals of Vehicle Dynamics. Warrendale: Society of Automotive Engineers, 1992.

Grant, Peter R., and Lloyd D. Reid. "Motion Washout Filter Tuning: Rules and Requirements." Journal of Aircraft 34.2 (1997): 144-151.

Grant, Peter R., and Lloyd D. Reid. "Protest: An Expert System for Tuning Simulator Washout Filters." Journal of Aircraft 34.2 (1997): 152-159.

Grant, Peter R., and Lloyd D. Reid. "Motion Algorithm for Large-Displacement Driving Simulator." Transportation Research Record 1403 : 98-106. (1993)

Idan, Moshe, and Meyer A. Nahon. "Off-Line Comparison of Classical and Robust Flight Simulator Motion Control." Journal of Guidance, Control, and Dynamics. 22.5 (1999): 702-709.

Nahon, Meyer A., Lloyd D. Reid. "Simulator Motion-Drive Algorithms: A Designer's Perspective." J. Guidance 13.2 (1990): 356-362.2

National Advanced Driving Simulator Overview. Iowa City: National Advanced Driving Simulator, The University of Iowa, 2008-2010.

Parrish, Russell V. , James E. Dieudonne, and Roland L. Bowles. "Coordinated Adaptive Washout for Motion Simulators." J. Aircraft 12.1 (1975): 44-50.

Appendix A: Vehicle Dynamics Matlab Code

```

clear all;
clc;
l = 8.925;
tf = 5.192;
tr = 5.208;
b = 4.730;
a = 4.195;
cg = 1.5;
c=3274.84;
C=3429.15;
wf = 1903.23;
wr = 1687.77;
m = 111.61;
Izz = 2222.62;
Ixx=868.68;
calphar = 11460;
calphaf = 11460;
mew=0.8;

g=32.2;
w=2.5;
A=3*.0135;

v(1)=0;
r(1)=0;
X(1)=800;
Y(1)=-1845;
s(1)=0;
u(1)=0.000000000001;
phi(1)=0;
phidot(1)=0;
theta(1)=0;
thetadot(1)=0;
K=82542.429;
pbar=0.5;
phibar= -4;
str=0.5; %understeer is positive
Q=str/phibar;
tspan = [0:.005:154];
k = (wf/(calphaf))- (wr/(calphar));
r_turn = (u^2)/(.2*g);
kbar=53865;
kbbbar=6035.29;
Iyy=1364.42;

fx(1)=0;
q=1.3;
dt=0.005;

```

```

for t=1:length(tspan)-1

    delta(t)=0;
    z(t)=0*delta(t);
    vz(t)=z(t);
    az(t)=z(t);
    BFf(t)=0;
    BFr(t)=0;

    if tspan(t)>0.01
    delta(t) = 0;
    BFr(t)=-800;

    end

    % If BF is negative vehicle should accelerate

if tspan(t)>15
    BFf(t)=250;
    BFr(t)=BFf(t)/1.3;
    end
if tspan(t)>30
% %
    BFf(t)=0;
    BFr(t)=BFf(t)/1.3;
    delta(t)=0.035;
end

if tspan(t)>33
    delta(t)=0.04;
end

if tspan(t)>38
    delta(t)=0.03;
end
%
if tspan(t) >42
    delta(t)=0.05;
end
if tspan(t) >45
    delta(t)=0.065;
end
if tspan(t) >47.05
    delta(t)=0;
end
if tspan(t) >55
    delta(t)=-0.035;
end
if tspan(t) >57
    delta(t)=-0.04;
end

```

```

if tspan(t) >63
    delta(t)=-0.06;
end
if tspan(t) >65
    delta(t)=-0.07;
end
if tspan(t) >66
    delta(t)=0;
end
if tspan(t) >70.25
    delta(t)=0.06;
end
if tspan(t) >71.5
    delta(t)=0.05;
end
if tspan(t) >75.7
    delta(t)=0;
end
if tspan(t) >77
    delta(t)=0.005;
end
if tspan(t) >80
    delta(t)=0;
end

if tspan(t) >82

    delta(t)=0.02;
end
if tspan(t) >89
    delta(t)=-0.005;
end
if tspan(t) >91
    delta(t)=-0.012;
end
if tspan(t) > 94
    delta(t)=-0.0065;
end
if tspan(t) > 100
    delta(t)=0.006;
end
if tspan(t) > 106
    delta(t)=0.035;
end
if tspan(t) > 113
    delta(t)=0;
end

if tspan(t) > 115
    delta(t)=0.01;
    BFr(t)=-800;
end

if tspan(t)>117.5

```

```

    delta(t)=0;
    BFr(t)=-800;
end

if tspan(t)>121.2
    BFf(t)=280;
    BFr(t)=BFf(t)/1.3;
end

if tspan(t) >127.3
    BFf(t)=0;
    BFr(t)=0;
    delta(t)=0.04;
end
if tspan(t) >132.7
    delta(t)=0.001;
end

if tspan(t) >136
    delta(t)=0.003;
end
if tspan(t) > 146.5
    delta(t)=0.029;
end

deltar(t)=Q*phi(t);

thetafl(t)=atan((v(t)+a*r(t))/(u(t)+(5.2/2)*r(t)));
alphafl(t)=thetafl(t)-delta(t);

thetafr(t)=atan((v(t)+a*r(t))/(u(t)-(5.2/2)*r(t)));
alphafr(t)=thetafr(t)-delta(t);

thetarl(t)=atan((v(t)-b*r(t))/(u(t)+(5.2/2)*r(t)));
alpharl(t)=thetarl(t)-deltar(t);

thetarr(t)=atan((v(t)-b*r(t))/(u(t)-(5.2/2)*r(t)));
alpharr(t)=thetarr(t)-deltar(t);

fyrl(t) = -calphar*alpharl(t);
fyrr(t) = -calphar*alpharr(t);
fyfl(t) = -calphaf*alphafl(t);
fyfr(t) = -calphaf*alphafr(t);

DeltaNrl(t)=-kbbar*thetarl(t);
DeltaNrr(t)=-kbbar*thetarr(t);
DeltaNfl(t)=-kbbar*thetafl(t);
DeltaNfr(t)=-kbbar*thetafr(t);
DeltaNr(t) = (-K*phi(t)-c*phidot(t))/(tr/2);
DeltaNf(t) = (-K*phi(t)-c*phidot(t))/(tf/2);
wfl(t) = 0.5*wf+DeltaNf(t)*(pbar/2)+0.5*DeltaNfl(t);
wfr(t) = 0.5*wf- DeltaNf(t)*(pbar/2)+0.5*DeltaNfr(t);
wrl(t) = 0.5*wr + DeltaNr(t)*((1-pbar)/2)-0.5*DeltaNrl(t);

```

```

wrr(t) =0.5*wr - DeltaNr(t)*((1-pbar)/2)-0.5*DeltaNrr(t);

if (X(t)>-164.04)
    mew(t)=0.8;
end
if (X(t)<-164.04)
    mew(t)=0.8;
end
%Tire Tracks
Xfl(t)=a+X(t);
Xfr(t)=a+X(t);
Xrl(t)=X(t)-b;
Xrr(t)=X(t)-b;

Yfl(t)=Y(t)-0.5*tf;
Yfr(t)=Y(t)+0.5*tf;
Yrl(t)=Y(t)-0.5*tr;
Yrr(t)=Y(t)+0.5*tr;

if ((mew(t)*wfl(t))/2)< abs(fyfl(t))
    fyfl(t)=-mew(t)*wfl(t)*sign(alphafl(t))*(1-
(mew(t)*wfl(t))/(4*calphaf*abs(tan(alphafl(t)))));

else
    fyfl(t) = -calphaf*alphafl(t);

end
if ((mew(t)*wfr(t))/2)< abs(fyfr(t))
    fyfr(t)=-mew(t)*wfr(t)*sign(alphafr(t))*(1-
(mew(t)*wfr(t))/(4*calphaf*abs(tan(alphafr(t)))));

else
    fyfr(t) = -calphaf*alphafr(t);
end
if ((mew(t)*wrl(t))/2)< abs(fyrl(t))
    fyrl(t)=-mew(t)*wrl(t)*sign(alpharl(t))*(1-
(mew(t)*wrl(t))/(4*calphar*abs(tan(alpharl(t)))));

else
    fyrl(t) = -calphar*alpharl(t);
end
if ((mew(t)*wrr(t))/2)< abs(fyrr(t))
    fyrr(t)=-mew(t)*wrr(t)*sign(alpharr(t))*(1-
(mew(t)*wrr(t))/(4*calphar*abs(tan(alpharr(t)))));

else
    fyrr(t) = -calphar*alpharr(t);

end

```

end

```

BFfl(t)=0.5*BFf(t);
BFfr(t)=0.5*BFf(t);
BFrl(t)=0.5*BFR(t);
BFrr(t)=0.5*BFR(t);

BFfl_N(t)=BFfl(t)/(wfl(t));
BFfr_N(t)=BFfr(t)/(wfr(t));

BFrl_N(t)=BFrl(t)/(wrl(t));
BFrr_N(t)=BFrr(t)/(wrr(t));

if(BFfl_N(t)>mew(t))
    BFfl(t)=mew(t)*(wfl(t));
    BFfl_N(t)=BFfl(t)/(wfl(t));
end

if(BFfr_N(t)>mew(t))
    BFfr(t)=mew(t)*(wfr(t));
    BFfr_N(t)=BFfr(t)/(wfr(t));
end

if(BFrl_N(t)>mew(t))
    BFrl(t)=mew(t)*(wrl(t));
    BFrl_N(t)=BFrl(t)/(wrl(t));
end

if(BFrr_N(t)>mew(t))
    BFrr(t)=mew(t)*(wrr(t));
    BFrr_N(t)=BFrr(t)/(wrr(t));
end

if(BFfl(t)>mew(t)*(wfl(t))*cos(alphafl(t)))
    BFfl(t)=mew(t)*(wfl(t))*cos(alphafl(t));
    fyfl(t)=-mew(t)*(wfl(t))*sin(alphafl(t));
end
if(BFrl(t)>mew(t)*(wfr(t))*cos(alphafr(t)))
    BFfr(t)=mew(t)*(wfr(t))*cos(alphafr(t));
    fyfr(t)=-mew(t)*(wfr(t))*sin(alphafr(t));
end
if(BFrl(t)>mew(t)*(wrl(t))*cos(alpharl(t)))
    BFrl(t)=mew(t)*(wrl(t))*cos(alpharl(t));
    fyfl(t)=-mew(t)*(wrl(t))*sin(alpharl(t));
end
if(BFrr(t)>mew(t)*(wrr(t))*cos(alpharr(t)))
    BFrr(t)=mew(t)*(wrr(t))*cos(alpharr(t));
    fyrr(t)=-mew(t)*(wrr(t))*sin(alpharr(t));
end
if((BFfl(t))^2+(fyfl(t))^2>(mew(t)*wfl(t))^2)
    fyfl(t)=-sqrt((mew(t)*wfl(t))^2-(BFfl(t))^2)*sign(alphafl(t));
end
if((BFfr(t))^2+(fyfr(t))^2>(mew(t)*wfr(t))^2)

```

```

    fyfr(t)=-sqrt((mew(t)*wfr(t))^2-(BFfr(t))^2)*sign(alphafr(t));
end
if((BFrl(t))^2+(fyrl(t))^2>(mew(t)*wrl(t))^2)
    fyrl(t)=-sqrt((mew(t)*wrl(t))^2-(BFrl(t))^2)*sign(alpharl(t));
end
if((BFrr(t))^2+(fyrr(t))^2>(mew(t)*wrr(t))^2)
    fyrr(t)=-sqrt((mew(t)*wrr(t))^2-(BFrr(t))^2)*sign(alpharr(t));
end

fyr(t)=fyrr(t)+fyrl(t);
Fxlr(t)=-BFrl(t);
Fxrr(t)=-BFrr(t);
Fxlfl(t)=-BFfl(t);
Fxrf(t)=-BFfr(t);

Fwx=-0.5*(BFrl(t)+BFrr(t)+BFfr(t)+BFfl(t));

Fylf(t)=fyfl(t)*cos(delta(t))+Fxlfl(t)*sin(delta(t));
Fxlfl(t)=Fxlfl(t)*cos(delta(t))-fyfl(t)*sin(delta(t));

Fyrf(t)=fyfr(t)*cos(delta(t))+Fxrf(t)*sin(delta(t));
Fxrf(t)=Fxrf(t)*cos(delta(t))-fyfr(t)*sin(delta(t));
fx(t)=Fxlfl(t)+Fxrf(t)+Fxlr(t)+Fxrr(t);

ax(t)=fx(t)/(m*g);
ay(t)=(Fyrf(t)+Fylf(t)+fyr(t))/(m*g);

beta(t)=atan((v(t)/u(t)));

speed(t)=sqrt(u(t)^2+v(t)^2);

z(t)=0;
options=[];

r(t+1)=r(t)+(((Fylf(t)+Fyrf(t))*a-(fyrr(t)+fyrl(t))*b+(Fxlfl(t)-
Fxrf(t)+Fxlr(t)-Fxrr(t))*(5.2/2))/Izz)*dt;
v(t+1)=v(t)+((1/m)*(Fylf(t)+Fyrf(t)+fyrl(t)+fyrr(t))-u(t)*r(t)-cg*(-((K-
(wf+wr)*(cg))*phi(t)+c*phidot(t)+(Fylf(t)+Fyrf(t)+fyrl(t)+fyrr(t))*cg)/Ixx
))*dt;
s(t+1)=s(t)+r(t)*dt;
X(t+1)=X(t)+(u(t)*cos(s(t))-(v(t))*sin(s(t)))*dt;
Y(t+1)=Y(t)+((v(t))*cos(v(t))+u(t))*sin(s(t))*dt;
phi(t+1)=phi(t)+phidot(t)*dt;
phidot(t+1)=phidot(t)+(-((K-
(wf+wr)*(cg))*phi(t)+c*phidot(t)+(Fylf(t)+Fyrf(t)+fyrr(t)+fyrl(t))*cg)/Ixx
))*dt;
u(t+1)=u(t)+((1/m)*fx(t)+(v(t)*r(t)))*dt;
theta(t+1)=theta(t)+thetadot(t)*dt;

```

```
thetadot(t+1)=thetadot(t)+(((-(BFf(t)+BFr(t))*cg)-kbar*theta(t)-
C*thetadot(t))/Iyy)*dt;
```

```
end
```

```
phiD=phi*57.3;
thetaD=theta*57.3;
yawD=yaw*57.3;
sD=s*57.3;
rD=r*57.3;
Ay=ay*32.2;
Ax=ax*32.2;
```

```
figure (1);
subplot(2,2,1)
plot(tspan(1:length(delta)),delta,'red');
xlabel('Time (s)')
ylabel('Delta (radians)')
hold on;
subplot(2,2,2)
plot(tspan, r,'red');
xlabel('Time (s)')
ylabel('r (rad/s)')
hold on;
subplot(2,2,3)
plot(tspan, phi, 'red');
xlabel('Time (s)')
ylabel('Roll Angle (degrees)')
hold on;
subplot(2,2,4)
plot(tspan(1:length(beta)), beta, 'red');
xlabel('Time (s)')
ylabel('Beta (Degrees)')
hold on;
```

```
figure (2);
subplot(2,2,1)
plot(tspan(1:length(ax)), ax,tspan(1:length(ay)), ay);
xlabel('Time (s)')
ylabel('Acceleration (gs)')
subplot(2,2,2)
plot(tspan(1:length(speed)), speed)
xlabel('Time (s)')
ylabel('Speed (ft/s)')
subplot(2,2,3)
plot(tspan(1:length(Fxlf)), Fxlf,tspan(1:length(Fxrf)),
Fxrf,tspan(1:length(Fxlr)), Fxlr,tspan(1:length(Fxrr)), Fxrr)
xlabel('Time (s)')
ylabel('Fx')
subplot(2,2,4)
plot(tspan(1:length(Fylyf)), Fylyf,tspan(1:length(Fyryf)),
Fyryf,tspan(1:length(fyrl)), fyrl,tspan(1:length(fyrr)), fyrr)
xlabel('Time (s)')
```

```
ylabel('Fy')
```

```
figure (3);
subplot(3,1,1)
plot(tspan, u)
xlabel('Time (s)')
ylabel('u ft/s')
subplot(3,1,2)
plot(tspan,v)
xlabel('Time (s)')
ylabel('v ft/s')
subplot(3,1,3)
plot(tspan(1:length(speed)), speed)
xlabel('Time (s)')
ylabel('Speed (ft/s)')
```

```
figure (4)
subplot(3,1,1)
plot(tspan(1:length(wfl)), wfl,tspan(1:length(wfr)), wfr,
tspan(1:length(wrl)), wrl, tspan(1:length(wrr)), wrr);
xlabel('Time (s)')
ylabel('Normal Loads')
hold on;
subplot(3,1,2)
plot(tspan(1:length(BFfl_N)), BFfl_N,tspan(1:length(BFfr_N)),
BFfr_N,tspan(1:length(BFrl_N)), BFrl_N,tspan(1:length(BFrr_N)), BFrr_N );
xlabel('Time(s)')
ylabel('BF/N')
subplot(3,1,3)
plot(tspan(1:length(BFfl)), BFfl,tspan(1:length(BFfr)),
BFfr,tspan(1:length(BFrl)), BFrl,tspan(1:length(BFrr)), BFrr );
xlabel('Time(s)')
ylabel('Brake Force')
```

```
figure (5)
```

```
I=imread('limerock.jpg');
rgb=imresize(I,1.7679);
```

```
image(rgb);
set(gca, 'YDir', 'norm')
hold on;
plot(X,-Y, 'red')
```

```
xlabel('X (feet)')
ylabel('Y (feet)')
axis equal
```

```
% hold on;
```

```
fid = fopen('vehiclerun_limerock.txt', 'wt'); % Open for writing
```


Appendix B: Vehicle Vector Matlab Code

```

clear all
clc
%Goes vehiclerun.txt
%SimTime    VehicleX    VehicleY    VehicleZ    VelocityX    VelocityY
VelocityZ    AccelX    AccelY    AccelZ    Roll    Pitch    Yaw    RollD    PitchD
YawD
load 'vehiclerun_limerock.txt'

SimTime      =vehiclerun_limerock(:,1);
VehicleX     =vehiclerun_limerock(:,2);
VehicleY     =vehiclerun_limerock(:,3);
VehicleZ     =vehiclerun_limerock(:,4);
VelocityX    =vehiclerun_limerock(:,5);
VelocityY    =vehiclerun_limerock(:,6);
VelocityZ    =vehiclerun_limerock(:,7);
AccelX       =vehiclerun_limerock(:,8);
AccelY       =vehiclerun_limerock(:,9);
AccelZ       =vehiclerun_limerock(:,10);
Roll         =vehiclerun_limerock(:,11);
Pitch        =vehiclerun_limerock(:,12);
Yaw          =vehiclerun_limerock(:,13);
RollD        =vehiclerun_limerock(:,14);
PitchD       =vehiclerun_limerock(:,15);
YawD         =vehiclerun_limerock(:,16);

figure(4);
plot(VehicleX,-VehicleY,'r');hold on;
n=25;
temp=VehicleX(1:n:length(SimTime));
quiver(VehicleX(1:n:length(SimTime)),-
VehicleY(1:n:length(SimTime)),AccelX(1:n:length(SimTime)).*cos(-
Yaw(1:n:length(SimTime))),AccelX(1:n:length(SimTime)).*sin(-
Yaw(1:n:length(SimTime))));
quiver(VehicleX(1:n:length(SimTime)),-
VehicleY(1:n:length(SimTime)),AccelY(1:n:length(SimTime)).*cos(-
Yaw(1:n:length(SimTime))+pi/2),AccelY(1:n:length(SimTime)).*sin(-
Yaw(1:n:length(SimTime))+pi/2),'g');
xlabel('X (feet)')
ylabel('Y (feet)')

axis('equal');
hold off;

figure(5)
plot(VehicleX,-VehicleY,'r');hold on;
n=50;
temp=VehicleX(1:n:length(SimTime));

```

```
quiver(VehicleX(1:n:length(SimTime)),-VehicleY(1:n:length(SimTime)),-  
AccelX(1:n:length(SimTime)),AccelY(1:n:length(SimTime)));
```

```
axis('equal');  
hold off;
```

Appendix C: Motion Base Parameter Matlab Code

```
whp=8;% natural frequency high pass filter rad/s
zhp=0.707% high pass damping
tau=0.01;% high pass time constant

% the Parker setup is over damped.
wllc=40% natural frequency lower level controller rad/sec
zllc=1% damping lower level controller

% Rotation Only Controller
wllc2=20 ;% natural frequency rotation only controller rad/s
zllc2=0.707 damping rotation only controller
```

Appendix D: Washout Algorithm Simulink Code

