

2017

# Vehicle Stabilization during Critical Cornering Scenarios Using Sliding Surface Control

Christina Elyse Boyd

*Bucknell University*, [ceb034@bucknell.edu](mailto:ceb034@bucknell.edu)

Follow this and additional works at: [https://digitalcommons.bucknell.edu/masters\\_theses](https://digitalcommons.bucknell.edu/masters_theses)

---

## Recommended Citation

Boyd, Christina Elyse, "Vehicle Stabilization during Critical Cornering Scenarios Using Sliding Surface Control" (2017). *Master's Theses*. 184.

[https://digitalcommons.bucknell.edu/masters\\_theses/184](https://digitalcommons.bucknell.edu/masters_theses/184)

This Masters Thesis is brought to you for free and open access by the Student Theses at Bucknell Digital Commons. It has been accepted for inclusion in Master's Theses by an authorized administrator of Bucknell Digital Commons. For more information, please contact [dcadmin@bucknell.edu](mailto:dcadmin@bucknell.edu).

VEHICLE STABILIZATION DURING CRITICAL CORNERING SCENARIOS  
USING SLIDING SURFACE CONTROL

by

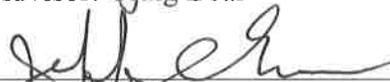
Christina Boyd

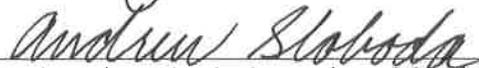
A Thesis

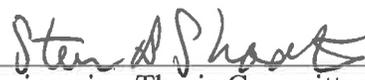
Presented to the Faculty of  
Bucknell University  
In Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Mechanical Engineering

Approved:

  
\_\_\_\_\_  
Advisor: Craig Beal

  
\_\_\_\_\_  
Department Chairperson: Jeffrey Evans

  
\_\_\_\_\_  
Engineering Thesis Committee Member: Andrew Sloboda

  
\_\_\_\_\_  
Engineering Thesis Committee Member: Steven Shooter

April 20<sup>th</sup>, 2017  
April 2017

## Acknowledgements

I would like to express my sincere gratitude to my thesis and academic advisor, Craig Beal. Craig Beal offered me an early research opportunity and got me started on this journey as well as provided patient and focused mentoring throughout the journey. His insightful guidance and encouragement at checkpoints along the way were instrumental in my progress. His commitment, attention to detail, and affable demeanor are greatly appreciated. I would also like to express my appreciation to my thesis committee members, Andrew Sloboda and Steven Shooter, for their encouragement and interest in my work. My thanks to Bucknell University and the Mechanical Engineering Department for providing funding and resources to make this project possible and to Sean Brennan for allowing use of the Penn State Test Track for vehicle testing. Finally, I would like to thank my parents for their support throughout my graduate studies.

## TABLE OF CONTENTS

List of Tables .....	v
List of Figures .....	vi
List of Parameters .....	viii
Abstract .....	xi
1 Introduction.....	1
1.1 Motivation.....	1
1.1.1 Trail Braking.....	2
1.2 Prior Work .....	6
2 Thesis Statement .....	12
3 Contributions.....	13
3.1 Development of a Tool that Enables 3D Visualization of the Vehicle Dynamics .....	13
3.2 Design of a Robust and Effective Controller for Critical Cornering Scenarios.....	14
3.3 Determination of Last Possible Safe Braking Point.....	14
4 Organization.....	15
5 Vehicle Models .....	16
5.1 Tire Models.....	18
5.2 3D Phase Portrait .....	22
5.2.1 Phase Portrait Validation .....	27
5.2.2 Stable Operating Bounds .....	31
6 Controller.....	34
6.1 Control Approach Selection.....	34
6.1.1 MPC Control.....	34
6.1.2 PID Control.....	36
6.1.3 H-Infinity Control .....	37
6.1.4 Sliding Surface Control.....	38
6.1.5 Selection Matrix (Methodology).....	41
6.2 Controller Design.....	46
6.2.1 Path-Following Function.....	47
6.2.2 Sliding Surface Control.....	50

7	Simulated Results.....	61
7.1	Baseline Controller Results.....	61
7.2	Sliding Surface Controller Results.....	64
7.3	Minimum Brake Distance .....	73
8	Conclusion .....	76
9	Bibliography .....	78
Appendices.....		xiii
Appendix A.....		xiii
Appendix B .....		xiv
Sliding Surface Controller Code .....		xiv
Four Wheel Vehicle Dynamics - Equations of Motion.....		xxii
Baseline Controller Code.....		xxiv

## List of Tables

Table 1. P1 Test Vehicle Parameters. ....	28
Table 2. Selection matrix compares potential control approaches. Control approaches are ranked for each of the weighted evaluation criteria. ....	43
Table 3. Selection matrix compares potential control approaches. Control approaches are scored for each criteria. The highest score represents the best suited control approach - highlighted in dark grey. ....	45
Table 4. Baseline Controller Simulation Inputs. ....	63
Table 5. Effect of each controller gain on tracking and overshoot. ....	66
Table 6. Sliding Surface Controller Simulation Inputs. ....	66
Table 7. Hand-tuned controller gains. ....	66
Table 8. Comparison to road radius design guidelines for highways ["Horizontal..."]. ....	67
Table 9. Sliding Surface Robustness Simulation Inputs. ....	69
Table 10. Minimum braking distance for various $\mu$ and R. ....	74

## List of Figures

Figure 1. g-g Diagram displaying the acceleration limits of a vehicle. ....	3
Figure 2. g-g Diagram illustrating the racing line of an expert racecar driver employing the trail braking technique.....	4
Figure 3. g-g Diagram illustrating an average driver braking and steering through a turn.....	5
Figure 4. $\beta - \gamma$ state-plane trajectory at $U_x = 150$ km/h, $\delta = 0$ rad [Inagaki 1994].....	7
Figure 5. $\beta - \dot{\beta}$ phase-plane trajectory at $U_x = 150$ km/h, $\delta = 0$ rad [Inagaki 1994]. ....	7
Figure 6. Vehicle trajectory for minimum time cornering maneuver [Velenis 2008]. ....	8
Figure 7. Four wheel planar vehicle model.....	16
Figure 8. Lateral brush tire model.....	19
Figure 9. Combined tire force limit. ....	20
Figure 10. Vehicle stability in the 2D phase portrait for $3^\circ$ steering, $U_{x,initial} = 15$ m/s, and $\mu = 0.75$ with a single stable equilibrium.....	23
Figure 11. Vehicle stability in the 3D phase portrait for $0^\circ$ of steering and no braking. $\kappa = 0$ and $\mu = 0.7$ . ....	25
Figure 12. Vehicle stability in the 3D phase portrait for $3^\circ$ of steering with braking. $\kappa = -0.05$ and $\mu = 0.7$ .....	25
Figure 13. Streamline spiraling effect seen with large amounts of braking.....	26
Figure 14. P1 electric vehicle used during experimental testing. ....	27
Figure 15. Comparison of experimental and theoretical data for $8^\circ$ steering without braking ( $\kappa = 0$ ) for a stable case. ....	30
Figure 16. Comparison of experimental and theoretical data for $8^\circ$ steering without braking ( $\kappa = 0$ ) for an unstable case. ....	30
Figure 17. Phase portrait for $2.5^\circ$ steering, no braking, and $\mu = 0.75$ . Roadway constraint is for a 110 m radius.....	33
Figure 18. Convex objective function with one minimum versus a nonconvex objective function with multiple minima. ....	36
Figure 19. Ideal sliding phase versus a practical sliding phase with chattering [Janardhanan 2016]. ....	40
Figure 20. Hierarchical controller structure. ....	46
Figure 21. Speed dependent look-ahead. ....	48
Figure 22. Behavior of curvature error and radius error. ....	49
Figure 23. Sliding surface for a cornering radius of 110 m. ....	51
Figure 24. Vehicle trajectories converge to a safe sliding surface.....	52

Figure 25. Surface plots of $S$ at several $U_y$ , $r$ , $U_x$ vehicle states along with the calculated control inputs indicated in red.....	56
Figure 26. Feasible regions of control in the 3D phase portrait for a 110 m cornering radius, $\mu = 0.4$ , $K_r = 28$ , $K_\beta = 50$ , and $K_{U_x} = 0.02$ .....	57
Figure 27. Slice of 3D phase portrait, showing feasible regions of control at $U_x = 10$ m/s, for $R = 110$ m, $\mu = 0.4$ , $K_r = 28$ , $K_\beta = 50$ , and $K_{U_x} = 0.02$ .....	58
Figure 28. Slice of 3D phase portrait, showing feasible regions of control at $U_x = 20$ m/s, for $R = 110$ m, $\mu = 0.4$ , $K_r = 28$ , $K_\beta = 50$ , and $K_{U_x} = 0.02$ .....	58
Figure 29. Position of the vehicle relative to the roadway path.....	62
Figure 30. Position of the vehicle relative to the roadway path. $d_{LA}$ represents the look-ahead distance, 10 m. ....	63
Figure 31. Vehicle response with baseline controller for $R = 110$ m, $U_{x,initial} = 28$ m/s, $U_{y,initial} = 0$ m/s, $r_{initial} = 0$ rad/s, $K_e = 0.2$ , $d_{LA} = 10$ . ....	64
Figure 32. Vehicle response for $\mu = 0.4$ and $\mu = 0.9$ with sliding surface controller in place. $U_{x,initial} = 28$ m/s, $U_{y,initial} = 0$ m/s, $r_{initial} = 0$ rad/s, and $R = 110$ m. ....	68
Figure 33. Vehicle lateral error from roadway centerline for various friction coefficients. $U_{x,initial} = 28$ m/s, $U_{y,initial} = 0$ m/s, $r_{initial} = 0$ rad/s, and $R = 110$ m. ....	70
Figure 34. Vehicle lateral error from roadway centerline for various initial longitudinal velocities. $\mu = 0.4$ , $R = 110$ m, $U_{y,initial} = 0$ m/s, and $r_{initial} = 0$ rad/s.....	70
Figure 35. Vehicle lateral error from roadway centerline for various cornering radii. $\mu = 0.4$ , $U_{x,initial} = 28$ m/s, $U_{y,initial} = 0$ m/s, and $r_{initial} = 0$ rad/s.....	71
Figure 36. Sequence of steering and braking inputs for a vehicle navigating a 110 m corner with hierarchical sliding surface control.....	72

## List of Parameters

$U_y$  – Lateral velocity

$U_x$  – Longitudinal velocity

$r$  – Yaw rate

$m$  – Vehicle mass

$I_{zz}$  – Yaw moment of inertia

$a$  – Distance from the vehicle CG to the front axle

$b$  – Distance from the vehicle CG to the rear axle

$d$  – Vehicle track width

$L$  – Vehicle wheelbase length

$g$  – Gravity

$\mu$  – Tire friction coefficient

$\mu_s$  – Tire sliding friction coefficient

$\delta_l$  – Left front steering angle

$\delta_r$  – Right front steering angle

$\alpha_{lf}$  – Slip angle

$\alpha_{slide}$  – Slip angle at which entire contact patch slides on the road

$\alpha_{peak}$  – Slip angle associated with peak friction force

$F_x$  – Longitudinal tire force

$F_y$  – Lateral tire force

$F_z$  – Normal force on tire

$C_\alpha$  – Linear tire cornering stiffness (lateral direction)

$C_x$  – Linear tire cornering stiffness (longitudinal direction)

$\kappa$  – Longitudinal slip

$\omega$  – Wheel rotational speed

$R_e$  – Effective rolling radius of the wheel

$U_{x,wheel}$  – Longitudinal velocity of the wheel center

$\gamma$  – Maximum stable yaw safety factor

$S$  – Sliding surface function

$N$  – North position of the vehicle in the global frame

$E$  – East position of the vehicle in the global frame

$N_{d_{LA}}$  – Predicted vehicle position from look-ahead (North)

$E_{d_{LA}}$  – Predicted vehicle position from look-ahead (East)

$e_{LA}$  – Lateral error from roadway centerline

$\rho_{Error}$  – Curvature error

$N_{IC}$  – Center of the actual cornering radius (North)

$E_{IC}$  – Center of the actual cornering radius (East)

$K_e$  – Total lateral error gain (Baseline controller)

$d_{LA}$  – Look-ahead distance

$\kappa_{d_{LA}}$  – Look-ahead distance gain

$K_\tau$  – Curvature gain

$K_r$  – Yaw rate gain

$K_\beta$  – Sideslip gain

$K_{U_x}$  – Longitudinal velocity gain

$R_{Road}$  – Radius of the road

$R_{Control}$  – Target cornering radius used in sliding surface control

$s_B$  – Distance along a path

$\Delta\psi$  – Heading difference between the vehicle and roadway

$\psi$  – Vehicle heading

$\psi_r$  – Road heading

## Abstract

While effective in improving handling and passenger safety, current vehicle control systems are generally limited to braking or steering control. This project presents an approach which integrates steering and braking actuation to further improve vehicle stability in critical cornering scenarios. A 3D phase portrait visualization tool enables examination of lateral velocity, longitudinal velocity, and yaw rate. This tool is used to determine vehicle stability under different operating conditions to inform the design of a controller. The proposed hierarchical controller defines a path-following function for the desired cornering radius and determines appropriate braking and steering inputs, using sliding surface control, to drive the vehicle to the desired path. A low-complexity vehicle model is used to formulate the sliding surface, while a high-fidelity model is used to determine optimal inputs. Simulations show that the sliding surface controller design is more effective than a baseline steering controller in keeping the vehicle on the roadway. Examination reveals that the complex sequence of braking and steering inputs is only feasible with the addition of a modern vehicle control system. While average drivers lack the ability to effectively employ such complex sequencing, modern control systems are capable of this coordination. When entering corners at speeds within the capability of the vehicle, but beyond the ability of the driver, these control sequences can help maintain stability to avoid an accident.

# 1 Introduction

## 1.1 Motivation

Road traffic injuries are the ninth leading cause of death globally, with an estimated 1.2 million people killed each year [Toroyan 2015]. More specifically run-off road crashes account for 60% of single vehicle crashes [Liu 2011]. 95% of the reasons for run-off road crashes are attributed to drivers, including driver performance errors, such as overcompensation, and driver decision errors, such as entering a corner too fast for its geometry or conditions [Liu 2011]. As a result, vehicle control systems are being developed to improve vehicle handling and passenger safety. These include systems already widely accepted, such as anti-lock brake systems, which prevent wheel lock-up, and brake-based electronic stability control systems, which apply the brakes to rotate the vehicle in the intended direction. In addition to braking and traction systems, active front steering systems further improve yaw and lateral vehicle stability [Adireddy 2014]. However, these vehicle control systems employ either braking or steering control. Current research now strives to integrate both braking and steering control to further improve vehicle stability. This paper focuses on leveraging the advantages of integrated braking and steering control, as seen in trail braking technique used by racecar drivers, for average drivers to further improve vehicle stability in critical cornering scenarios.

### 1.1.1 Trail Braking

Driver assistance in critical cornering scenarios, using coordination of braking and steering, is of specific interest. With that in mind, a racing technique used to navigate corners at high speeds is considered as a starting point for this research. Highly effective in racing, the trail braking technique is specifically intended to minimize time by achieving maximum speed through a turn. If the focus is shifted from maximizing speed through a turn, to moderating speed through a turn to allow the vehicle to stay within stable operating bounds while cornering, the technique can be adapted to assist the driver of a passenger vehicle to remain in control when driving at the limits of their capability.

The trail braking technique suggests that a controller based on coordinated braking and steering control is more effective than individual application of either braking or steering in critical cornering scenarios. This technique entails braking past the entrance of a corner and allows expert racing drivers to travel faster through the corner while maintaining vehicle stability. The concept behind this technique might be applied to passenger vehicles to avoid an accident when cornering at speeds that are within the capability of the vehicle but beyond the ability of the driver.

The basis of trail braking originates from the traction friction circle or g-g diagram, as shown in Figure 1, which describes a vehicle's longitudinal acceleration and lateral acceleration limits based on the force-producing limits of the tires [Mitchell 2004]. The diagram shows that there is always a tradeoff between longitudinal acceleration

(braking) and lateral acceleration (steering) when driving at the handling limits. For example, if the driver brakes at the limits, the longitudinal deceleration of the vehicle increases, and its peak lateral acceleration capability, beneficial for navigating corners, decreases. Therefore, the peak capacity to brake or turn is decreased by use of the other actuator.

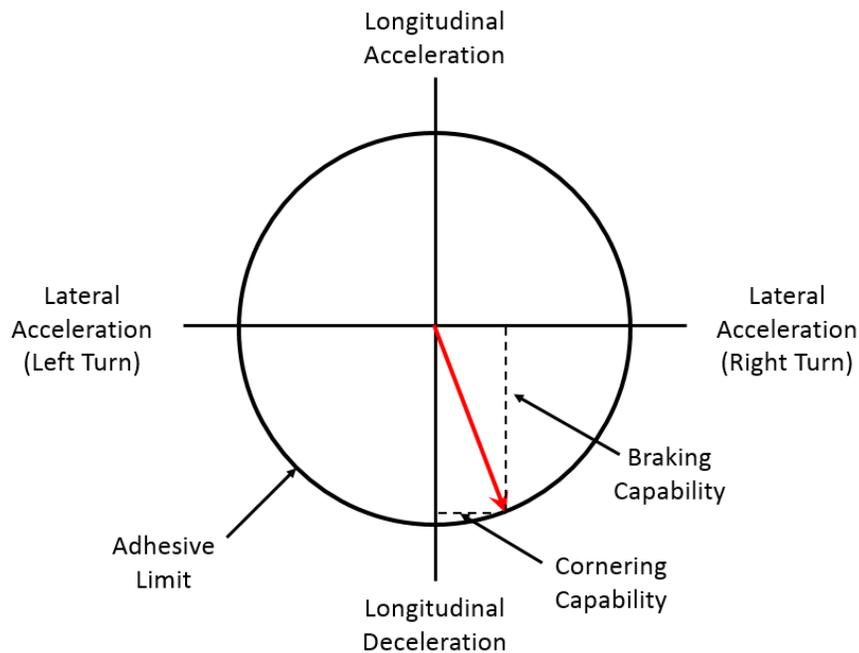


Figure 1. g-g Diagram displaying the acceleration limits of a vehicle.

Expert racing drivers follow a cornering sequence to utilize a vehicle's maximum acceleration limits by traversing the perimeter of the traction circle, allowing them to travel faster through a corner, as seen in Figure 2. That is, an expert racing driver navigates a corner by suddenly braking prior to the entrance to a turn (Points 1 to 2); slowly releasing the brakes from initiation of and throughout the turn (Points 2 to 3); and then completely releasing the brakes at the apex of the turn (Point 4) [Mitchell 2004].

This coordination of inputs allows the driver to make full use of the tire force capabilities and travel faster through a corner.

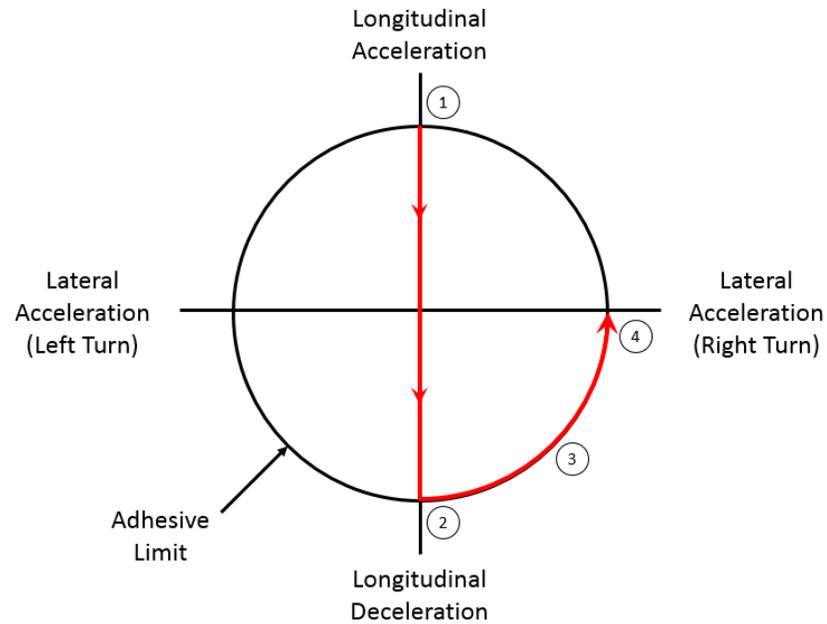


Figure 2. g-g Diagram illustrating the racing line of an expert racecar driver employing the trail braking technique.

Racing drivers are capable of performing this task; however, average drivers lack this experience. When entering a corner, average drivers are accustomed to braking (Point 1 to Point 2), and then releasing the brakes to steer through the corner (Point 3 to Point 4), which does not utilize the vehicle's capability to combine deceleration with cornering throughout the turn, as seen in Figure 3.

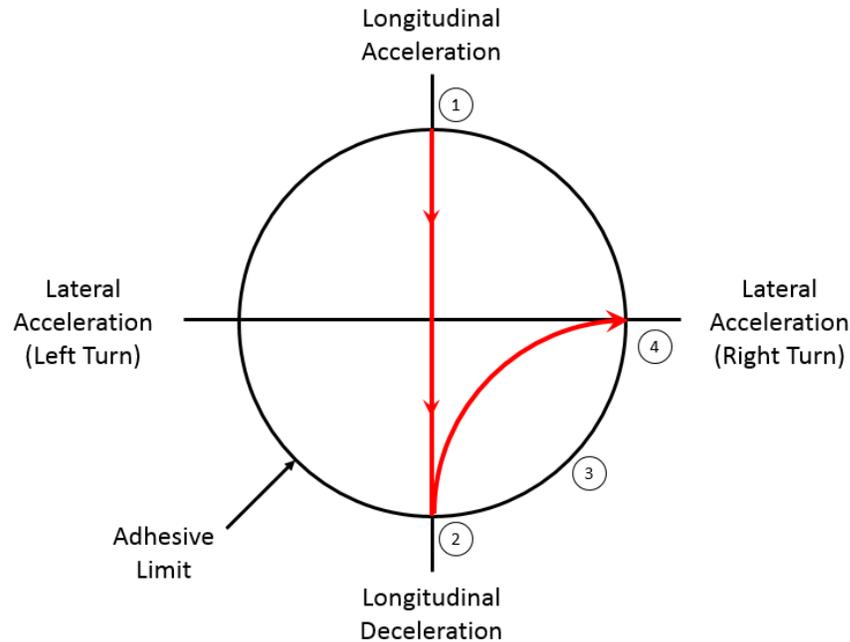


Figure 3. g-g Diagram illustrating an average driver braking and steering through a turn.

While average drivers lack the experience to employ trail braking or a similar cornering sequence, a control system capable of replicating these dynamics can help a driver safely navigate a corner in an emergency. The focus of the trail braking technique is to navigate a corner in the minimum time, by maximizing the speed at which the corner is traversed. However, the trail braking technique can be tailored, with the help of a control system, to adjust speed through a corner and maintain vehicle control within lane bounds to prevent a corner overrun. This tailored approach shifts the focus from racing to safe cornering in situations where an average driver enters a corner too fast. The focus of this paper is on leveraging the advantages of integrated braking and steering control for average drivers to further improve vehicle stability and safety in critical cornering scenarios.

## 1.2 Prior Work

Prior work in vehicle dynamics touched on ways of visualizing vehicle stability as well as methods of stabilizing and controlling vehicles in various accident avoidance scenarios. Past research in vehicle stability, by Inagaki et. al., investigated the effect of steering on vehicle stability using a 2D phase portrait method to assess the vehicle sideslip angle and yaw rate dynamics, as seen in Figure 4, as well as the vehicle sideslip angle and angular velocity phase-plane method, as seen in Figure 5 [Inagaki 1994]. The study determined that the side slip angle and angular velocity phase-plane method is more effective in representing nonlinear vehicle stability than the conventional state-plane model with side slip angle and yaw velocity. Analysis of the 2D phase portrait was helpful in identifying the boundaries beyond which stable limits are exceeded and in visualizing the effect of steering on the lateral and rotational motions of the vehicle. The 2D phase portraits, however, did not enable the longitudinal effects present with braking to be displayed.

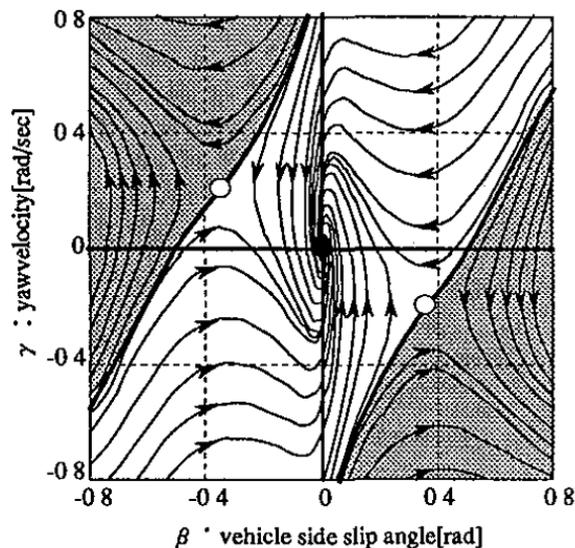


Figure 4.  $\beta - \gamma$  state-plane trajectory at  $U_x = 150$  km/h,  $\delta = 0$  rad [Inagaki 1994].

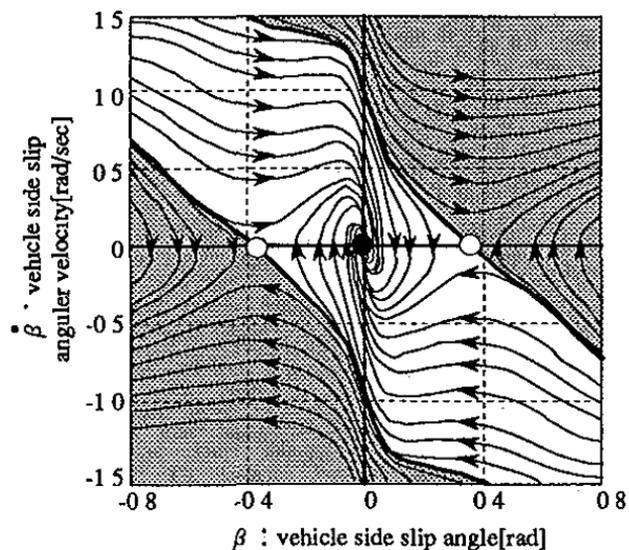


Figure 5.  $\beta - \dot{\beta}$  phase-plane trajectory at  $U_x = 150$  km/h,  $\delta = 0$  rad [Inagaki 1994].

While Inagaki's et. al. study defined the effect of steering, another study by Velenis et. al. defined the optimal turn geometry to minimize time around a given corner, replicating trail braking maneuvers [Velenis 2008]. Velenis et. al. laid out the steering,

braking, throttle commands as described by the g-g diagram and showed that the optimal path for minimum time cornering is to enter the corner on the outside of the lane, move to the inside of the lane at the apex, and exit the corner by returning to the outside of the lane, as seen in Figure 6. The optimal vehicle states for these aggressive cornering maneuvers utilized high sideslip angles and yaw rates.

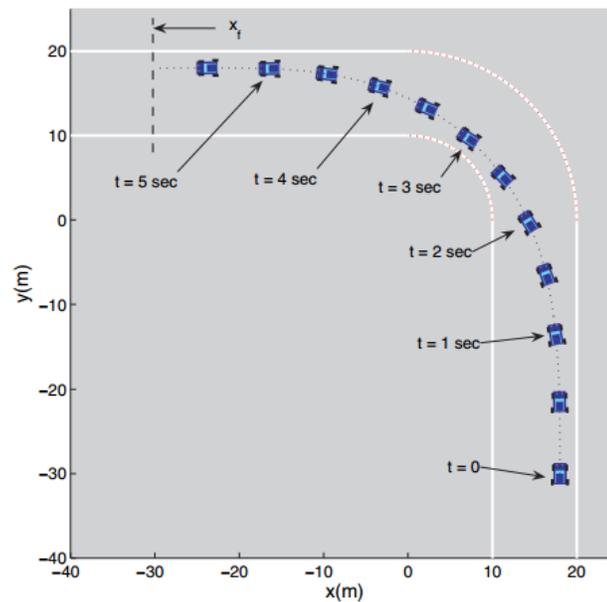


Figure 6. Vehicle trajectory for minimum time cornering maneuver [Velenis 2008].

Research by Adireddy et. al. focused on the development of a controller which integrated wheel torque and steering control inputs. For example, Adireddy et. al. presented a vehicle chassis control system, based on Model Predictive Control and a simplified tire model, to enhance vehicle yaw motion and to track a desired trajectory by controlling front steer and wheel torque inputs [Adireddy 2014]. Collision avoidance or double lane change maneuvers were carried out to determine the effectiveness of the controller. Results indicated that the performance of the controller, based on the

simplified tire model, is similar to that of a controller with the more complex Magic Formula tire model. While this controller was not specifically used for cornering scenarios, it used steering and braking inputs for safety maneuvers, and presents a framework which may be expanded upon for the development of a controller for cornering scenarios.

In addition to the work in controlling the dynamics, focus also has been given to sensing the point of critical cornering. Nozaki found that the point of critical cornering can easily be sensed based on a change in body slip angle and a low roll rigidity. Large body slip and low roll rigidity lead to improved sensing of the critical cornering point. Determination of the last possible safe control point could be helpful in the design of a controller for critical cornering scenarios [Nozaki 2005].

Effort by Kwak and Park was geared towards using multiple sliding mode control to enhance stability and handling in emergency situations through individual wheel braking. Individual wheel braking improved longitudinal as well as yaw motions of the vehicle. For example, braking was used to slow the vehicle down in the longitudinal direction and brake-torque control was used to apply a yaw moment to the vehicle, rotating it in the intended direction. The need for a robust control approach was also stressed due to the complex, nonlinear nature of the vehicle dynamics. The paper alluded to multiple sliding mode control as a viable and simple approach for achieving this requirement in cornering applications. However, the developed controller involved only braking control [Kwak 2001].

Additional research focused on stabilizing vehicles in the drift region. Nozaki and Ueda stated that while it was dangerous to rely solely on differentiation steering during cornering maneuvers, leveraging differentiation active steering assistance in the drift region improves stability. Drivers struggle to rely solely on differentiation steering because the steering wheel must be continuously turned and mapped to a turn while cornering. However, when differentiation steer assistance, which adjusts the steering wheel angle based on feedback about vehicle slip angle, was combined with differentiation steering, the vehicle was able to remain in the drift area and subsequently regain stability. The intent of differentiation steer assistance was to implement timely countersteer and avoid spin out in conditions where the wheels were slipping [Nozaki 2007].

While Nozaki and Ueda's study investigated drift control, another study by Yoshino and Nozaki investigated traction control and camber angle control in cornering margins. In situations where lateral acceleration is high, the cornering force of the front wheels decreases and becomes less effective. Therefore traction control was applied to inner and outer wheels to improve steering effectiveness at high lateral accelerations. Vehicle response improved for double lane change maneuvers; however, spinout was still unavoidable in the total skid margin where there is little room in the forward and reverse directions of the tires. Tire camber angle control was instead applied in the total skid margin to reduce the vehicle's lateral acceleration and prevent a skid out. Camber angle control was shown to be more effective than traction control for increasing maneuverability and stability in the total skid margin. The study investigated the

independent effects of braking and a type of steering control and found each to be useful in different regions of a corner; however, the study did not investigate the combined effect of braking and steering control [Yoshino 2014].

Attention has also been given to utilizing the trail braking technique for racing. Kritayakirana and Gerdes developed an autonomous racing controller with the goal of tracking the racing line while utilizing maximum tire limits to finish the race in the fastest time [Kritayakirana 2012]. Kritayakirana and Gerdes broke the problem down into two phases, path generation and vehicle control. First, path generation defined the optimal path, or racing line. The optimal path was based on clothoid curves which offer a number of advantages, such as linearly changing curvature. Then, vehicle control determined the braking, throttle, and steering inputs necessary to track the racing line at the limits of handling. The controller was specific to autonomously controlled racing vehicles, but shows promise for development of a non-intrusive driver assistance system that works in conjunction with the driver, to provide assistance during emergency maneuvers.

Each of these studies has contributed to a better understanding of vehicle dynamics by addressing different aspects of vehicle stability and different control approaches to maintaining stability. However, none of the research noted addresses integrated braking and steering to aid in emergency cornering for passenger vehicles. The following research specifically focuses on coordinated braking and steering to maintain vehicle stability in cornering maneuvers for improved passenger vehicle safety.

## 2 Thesis Statement

The proposed work is to develop a controller based on combined steering and braking to prevent run-off-road crashes when cornering. The outcomes of this work are (1) a tool that enables 3D visualization of vehicle dynamics, (2) a robust and effective controller for critical cornering scenarios, and (3) determination of the last possible braking point based the controller presented in this paper. Ultimately, the goal is to derive a controller that safely and stably maneuvers the vehicle to remain on the road during critical cornering scenarios.

### 3 Contributions

This research advances the understanding of safe vehicle cornering maneuvers for passenger vehicles by deriving a controller to safely navigate corners in over speed conditions. Major elements of the research and their respective contributions are highlighted below.

#### 3.1 Development of a Tool that Enables 3D Visualization of the Vehicle Dynamics

A tool that enables 3D visualization serves as the starting point for the proposed research. It is very common in vehicle control to work with a model that assumes constant vehicle speed. While this approach works well for lateral or rotational control of the vehicle at highway speeds, it fails to properly represent the coupling between lateral, longitudinal, and rotational motions. Expansion of a commonly employed 2D tool to a 3D tool enables greater insight into the relationship between the variables that impact safe and stable corner maneuvering.

Data obtained during experimental cornering scenarios validates this tool. The experimental testing builds confidence in the tool, enabling reliable prediction of vehicle performance under a variety of operating conditions.

### 3.2 Design of a Robust and Effective Controller for Critical Cornering Scenarios

The design of a control approach for cornering focuses on replicating the skills of expert drivers and leveraging the advantages of the trail braking racing technique for average drivers of passenger vehicles. The presented controller enables examination of various inputs and cornering conditions to determine the order in which and the extent to which steering and braking may best be applied to maintain control during cornering. The goal is a non-invasive controller design that takes into account road conditions and vehicle parameters and utilizes the vehicle's maximum handling capabilities to keep the car safely on the road. Simulations provide feedback on the robustness and effectiveness of the controller under varying conditions.

### 3.3 Determination of Last Possible Safe Braking Point

Given the selected controller design, the last possible braking point where the controller can safely negotiate the corner is estimated. This information could be used in conjunction with vehicle-to-infrastructure communication systems capable of identifying upcoming cornering conditions. For example, an on-board controller could predict the time at which the controller would need to assist the driver to maintain safe lane tracking. Additionally, this information could enable safe higher speed autonomous vehicle operation in the future.

## 4 Organization

The remainder of this thesis is organized as follows. Chapter 5 presents the vehicle models used to describe the vehicle dynamics, the 3D phase portrait tool utilized in this work, and the unfavorable operating bounds. Also discussed is the experimental test data used to validate the developed 3D model. Chapter 6 addresses the various control approaches examined and identifies the approach best suited for this cornering application. The hierarchical structure of the path-following and sliding surface controller is also presented. Chapter 7 examines simulations comparing the sliding surface controller to a baseline steering controller to illustrate the effectiveness and robustness of the designed controller. Finally, the sequence of steering and braking inputs needed throughout the corner, and the last possible safe braking point are reviewed.

## 5 Vehicle Models

Many different vehicle models can be used in the analysis of vehicle dynamics. One common model is the bicycle model. This model assumes a constant longitudinal velocity,  $U_x$ , to allow for linearity in the equations of motion. It is based on the small angle approximation and is a reasonable approximation of the slow changing vehicle speed due to the significant mass of the vehicle. However, this model however does not allow the effects of braking to be analyzed and breaks down when large changes longitudinal vehicle speed are present.

Therefore, a four wheel vehicle model with variable  $U_x$  is used as the basis for this research. This complexity allows a detailed representation of the dynamics, without overcomplicating calculations or significantly increasing computation time. The four wheel vehicle model has three states and results in a set of nonlinear equations which describe the rotational, lateral, and longitudinal velocities of the vehicle.

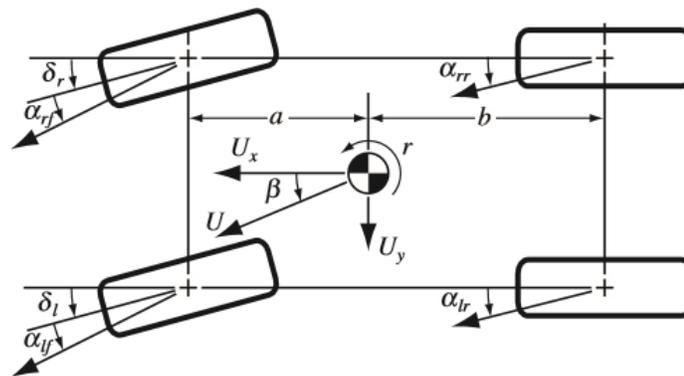


Figure 7. Four wheel planar vehicle model.

The model, illustrated in Figure 7, yields the following equations of motion:

$$\dot{U}_y = \frac{1}{m} [(F_{xlf} \sin \delta_l + F_{yly} \cos \delta_l) + (F_{xrf} \sin \delta_r + F_{yry} \cos \delta_r) + F_{yly} + F_{yry}] - rU_x \quad (\text{Eq. 1})$$

$$\dot{r} = \frac{1}{I_{zz}} \left[ a \left( (F_{xlf} \sin \delta_l + F_{yly} \cos \delta_l) + (F_{xrf} \sin \delta_r + F_{yry} \cos \delta_r) \right) - b(F_{yly} + F_{yry}) \right. \\ \left. + d \left( (F_{xrf} \cos \delta_r - F_{yry} \sin \delta_r) + F_{xrr} - (F_{xlf} \cos \delta_l - F_{yly} \sin \delta_l) - F_{xlr} \right) \right] \quad (\text{Eq. 2})$$

$$\dot{U}_x = \frac{1}{m} [(F_{xlf} \cos \delta_l - F_{yly} \sin \delta_l) + (F_{xrf} \cos \delta_r - F_{yry} \sin \delta_r) + F_{xly} + F_{xry}] + rU_y \quad (\text{Eq. 3})$$

where  $U_y$  is the lateral velocity,  $r$  is the yaw rate,  $U_x$  is the longitudinal velocity, and the parameters  $m$  and  $I_{zz}$  are the vehicle mass and yaw moment of inertia, respectively.

Parameters  $a$  and  $b$  represent the distance from the vehicle CG to the front axle and the rear axle, respectively, and parameter  $d$  is the track width. The vehicle slip angles, which are physically the difference between the true heading and true travel direction, are described in terms of vehicle states and the steering angle input:

$$\alpha_{lf} = \tan^{-1} \frac{(U_y + ar)}{(U_x - dr)} - \delta_l \quad (\text{Eq. 4})$$

$$\alpha_{rf} = \tan^{-1} \frac{(U_y + ar)}{(U_x + dr)} - \delta_r \quad (\text{Eq. 5})$$

$$\alpha_{lr} = \tan^{-1} \frac{(U_y - br)}{(U_x - dr)} \quad (\text{Eq. 6})$$

$$\alpha_{rr} = \tan^{-1} \frac{(U_y - br)}{(U_x + dr)} \quad (\text{Eq. 7})$$

Unlike the bicycle model, the vehicle slip angle equations, are not dependent on the small angles approximation. The forces in equations 1-3 are determined using these vehicle slip angle equations in conjunction with a tire model.

## 5.1 Tire Models

The tire forces in the equations of motion are found using either a lateral brush tire model or coupled lateral-longitudinal brush tire model. The lateral brush tire model, illustrated in Figure 8, is based on a physical model of a tire consisting of a rigid ring surrounded by brushes or threaded elements that interact with the road over a rectangular contact patch. The contact patch is known as the portion of the tire that is in contact with the road. When cornering, the brushes at the front of the contact patch are unstressed because this rubber just begins to come in contact with the road. However, the brushes at the back of the contact patch are characterized by high stress and deflect as the wheel moves laterally throughout the contact patch. This stress in the brushes is proportional to the displacement of the wheel.

The lateral brush model uses the concepts of force demand and force availability to determine the total tire force. Using solid mechanics principles, the tire force demand can be determined based on the stiffness,  $C$ , and deflection of the brushes. The tire force demand increases linearly with tire slip angle for small slip angles, as seen in Figure 8, where the slope is characterized by a linear tire cornering stiffness,  $C_\alpha$ . However, the amount of force available to keep the brushes in contact with the road is limited by the normal load and surface friction coefficients. A constant friction coefficient is assumed throughout the contact, but the normal load is assumed to be parabolic, resulting in a parabolic pressure distribution. This leads to a parabolic distribution of available adhesion friction force and parabolic distribution of available sliding friction force.

When the force demand exceeds the force available, the tire brushes begin sliding. As the slip angle continues to increase, the contact patch continues to slide until the tire is fully sliding and produces a force equivalent to the maximum sliding friction.

Notice in Figure 8 that the tire initially behaves linearly at low slip angles as the tire force is dependent only on the brush deflections, not the friction coefficient. As the force demand exceeds the available friction force, the curves move away from the linear slope and the friction force available dominates [Beal 2011].

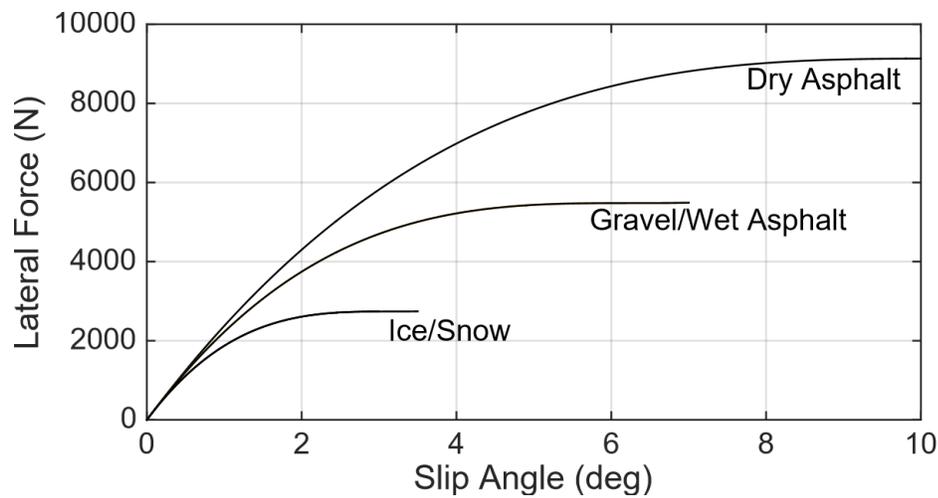


Figure 8. Lateral brush tire model.

The relationship between the slip angle and the tire force is given by the following piecewise expression:

$$\alpha_{slide} = \tan^{-1} \frac{q\mu F_z}{C_\alpha} \quad (\text{Eq. 8})$$

$$q = \frac{C_\alpha \tan \alpha}{\mu F_z} \quad (\text{Eq. 9})$$

$$F_y = \left\{ \begin{array}{ll} \left( -q + \frac{2 - R_e}{3} q|q| - \frac{1 - \frac{2}{3}R_e}{9} q^3 \right) \mu F_z, & \alpha \leq \alpha_{slide} \\ -sgn(\alpha) \mu F_z, & otherwise \end{array} \right\} \quad (\text{Eq. 10})$$

where  $\alpha_{slide}$  is the angle at which the entire contact patch slides on the road. The smaller angle associated with peak friction force can be found by differentiation of the model and is given by:

$$\alpha_{peak} = \tan^{-1} \frac{3\mu F_z}{C_\alpha} \quad (\text{Eq. 11})$$

The brush tire model can also be formulated to couple the lateral and longitudinal tire forces. The total force on a tire is the vector sum of the lateral and longitudinal components and is limited by the total available force. That is,

$$F_{tot} = \sqrt{F_x^2 + F_y^2} \leq \mu F_z, \quad (\text{Eq. 12})$$

as illustrated in Figure 9.

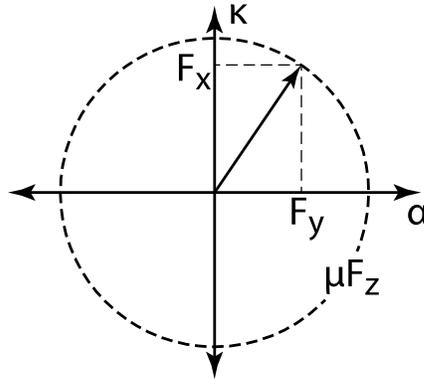


Figure 9. Combined tire force limit.

The coupled tire model relies on the calculation of longitudinal tire slip, the difference between the free rolling velocity and actual rotational speed of each wheel, written as:

$$\kappa = \frac{\omega R_e - U_{x,wheel}}{U_{x,wheel}} \quad (\text{Eq. 13})$$

where  $\omega$  is the wheel rotational speed,  $R_e$  is the effective rolling radius of the wheel, and  $U_{x,wheel}$  is the longitudinal velocity of the wheel center. Note that the magnitude of the longitudinal tire slip indicates the severity of braking. Furthermore, the longitudinal tire slip is conceptually a measure of the brush stresses in the longitudinal direction.

Similarly, the lateral tire slips, given in equations 4-7, are a measure of the brush stresses in the lateral direction. The longitudinal tire slip, in conjunction with the lateral tire slips from equations 4-7, can be used to model the coupled forces,  $F_x$  and  $F_y$  on each axle, which are given here in equations 14-17.

$$\xi = \sqrt{C_x^2 \left( \frac{\kappa}{1 + \kappa} \right)^2 + C_\alpha^2 \left( \frac{\tan \alpha}{1 + \kappa} \right)^2} \quad (\text{Eq. 14})$$

$$F = \begin{cases} \xi - \frac{1}{3\mu F_z} \left( 2 - \frac{\mu_s}{\mu} \right) \xi^2 + \frac{1}{9\mu^2 F_z^2} \left( 1 - \frac{2\mu_s}{3\mu} \right) \xi^3, & \xi \leq 3\mu F_z \\ \mu_s F_z, & \xi > 3\mu F_z \end{cases} \quad (\text{Eq. 15})$$

$$F_x = \frac{C_x}{\xi} \left( \frac{\kappa}{1 + \kappa} \right) F \quad (\text{Eq. 16})$$

$$F_y = \frac{-C_\alpha}{\xi} \left( \frac{\tan \alpha}{1 + \kappa} \right) F \quad (\text{Eq. 17})$$

These tire forces can be used in conjunction with the equation of motion shown in equations 1-3, to model the dynamics of the vehicle.

## 5.2 3D Phase Portrait

The vehicle behavior must be well understood before designing a controller. Phase portraits are a common tool used to understand how dynamic systems respond under various inputs and parameters. They are created by plotting the state derivatives of a system over time and visually show the stability of a system. Initial conditions with trajectories which converge to a node are considered stable operating points while those with trajectories that diverge are considered unstable.

Past research in this area utilized a 2D phase portrait method, investigating yaw rate and vehicle sideslip, to visualize the vehicle dynamics and determine vehicle stability, similar to the illustration shown in Figure 10 [Inagaki 1994]. However, this well-known 2D phase portrait assumes a constant longitudinal velocity and does not allow the effects of braking to be analyzed. This 2D tool serves as a starting point, but is expanded upon to enable examination of lateral velocity, yaw rate, and longitudinal velocity as well as determination of vehicle stability under different operating conditions. The 3D phase portrait supports investigation of braking while turning, which is the basis of the proposed controller.

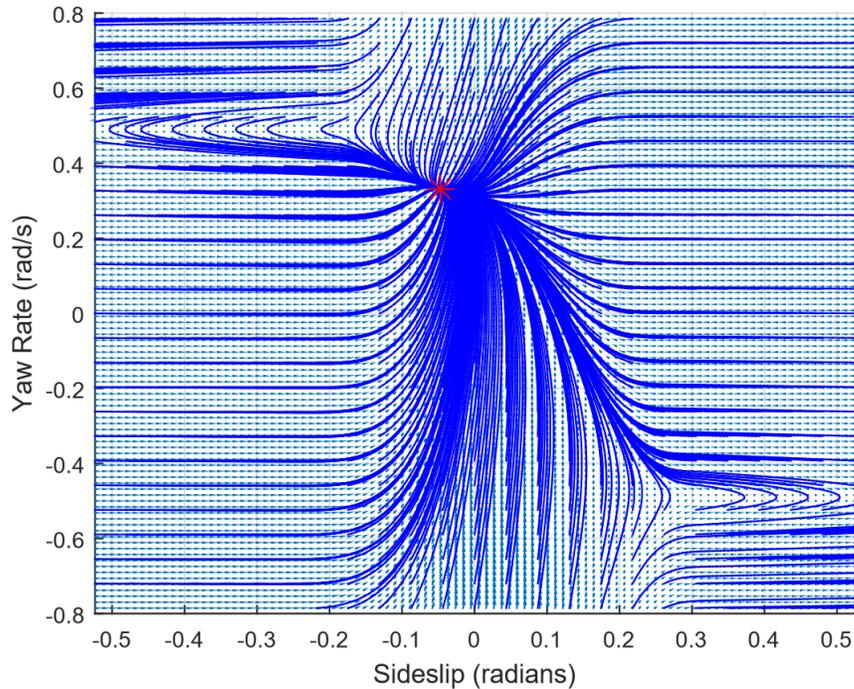


Figure 10. Vehicle stability in the 2D phase portrait for  $3^\circ$  steering,  $U_{x,initial} = 15$  m/s, and  $\mu = 0.75$  with a single stable equilibrium.

The nonlinear dynamic equations describing  $U_y$ ,  $r$ , and  $U_x$ , shown in equations 1-3, were coded in MATLAB to create the 3D phase portrait model. Streamlines of vehicle trajectories are included, given an estimated coefficient of friction, cornering stiffness, steering angle, longitudinal tire slip, and vehicle parameters for various initial conditions. In addition, the stable equilibria at every longitudinal velocity (if existing) are computed and plotted in red stars. The stable equilibria are located by computing the lateral velocity and yaw rate that correspond to the intersection of the  $U_y$  and  $r$  nullclines.

Previous research found that if a vehicle had a stable operating region in the 2D plane, trajectories within the boundary converged to a single equilibrium and stopped, as seen in Figure 10. However, Figures 11 and 12 illustrate that with the 3D approach,

which accounts for varying velocity (i.e., braking), stable streamlines converge toward a stable operating curve and then travel along this stable operating curve. The concept of a single equilibrium is no longer useful. Therefore, the concept of a single equilibrium is replaced by the concept of a stable operating curve, shown by red stars.

Initial conditions with trajectories that reach a stable operating curve are considered stable operating points. Initial conditions with trajectories that did not reach a stable operating curve are considered unstable operating points. Vehicle stability for the most basic case, with 0 degrees steering and no braking, is shown in Figure 11, along with the stable equilibria at each longitudinal velocity. Note that the stable equilibria are based on constant steering, constant braking, and a constant friction coefficient.

Investigation focused on how factors such as the coefficient of friction, braking (longitudinal tire slip), and steering angle affected the stable operating curve and vehicle trajectories. The coefficient of friction tightens the stability boundary while longitudinal tire slip affects the severity of the drop in the streamline's longitudinal velocity before reaching the stable operating curve. Figure 11 illustrates that with no longitudinal tire slip, the streamlines only drop slightly in longitudinal velocity before converging with the stable operating curve, due to natural losses such as friction. However, with a front and rear longitudinal tire slip of 5% (braking), the streamlines experience a greater drop in longitudinal velocity,  $U_x$ , before reaching the stable operating curve, as seen in Figure 12. In addition, steering angle primarily affects the orientation of the stable operating curve. For 0 degrees of steering with no longitudinal tire slip, as seen in Figure 11, the stable operating curve remains vertical, but for increased steering angles such as 3 degrees of

steering, as seen in Figure 12, the curve shifts towards positive yaw rates and negative lateral velocities for large longitudinal velocities. Note that the stable operating curve shown in Figure 12 is based on constant steering, constant braking, and a constant friction coefficient.

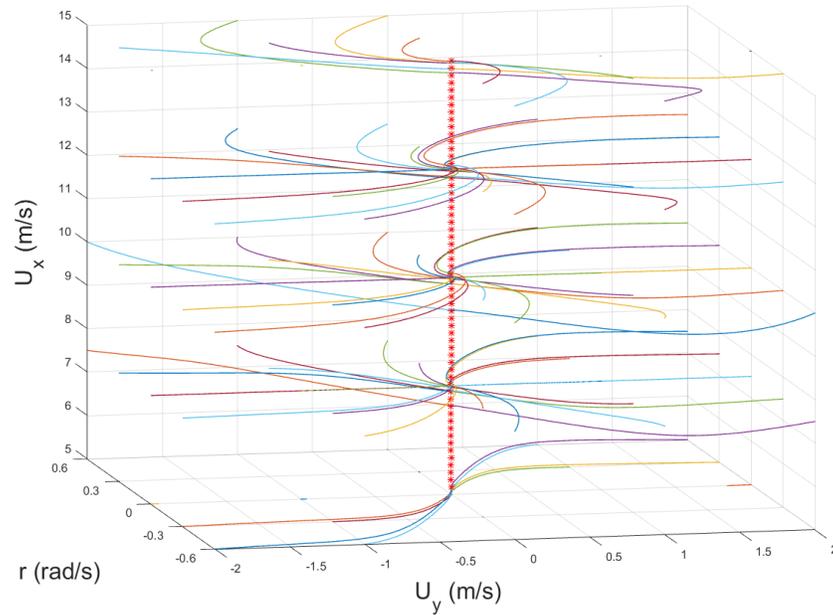


Figure 11. Vehicle stability in the 3D phase portrait for  $0^\circ$  of steering and no braking.  $\kappa = 0$  and  $\mu = 0.7$ .

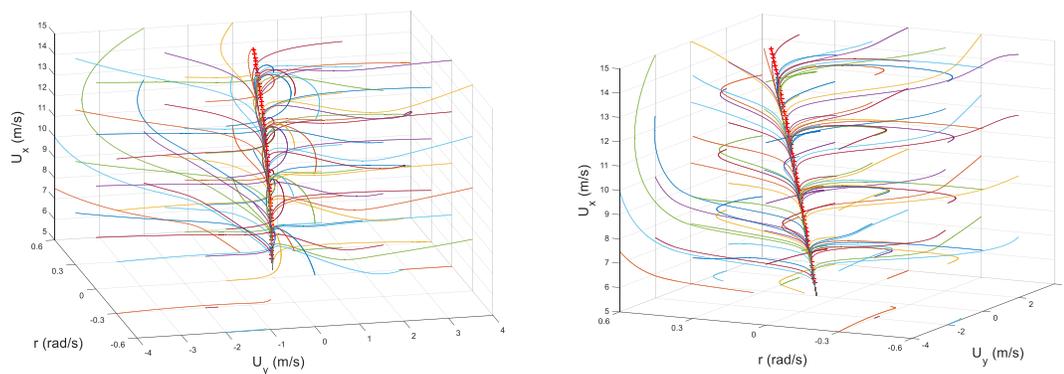


Figure 12. Vehicle stability in the 3D phase portrait for  $3^\circ$  of steering with braking.  $\kappa = -0.05$  and  $\mu = 0.7$ .

Another aspect that becomes apparent in 3D is that streamlines spiral around the stable operating curve as they converge toward it. Figure 13 illustrates that this spiraling effect becomes more pronounced with a larger longitudinal tire slip and thus more braking. The spiraling effect suggests that a heavily braked vehicle, recovering from a large yaw rate/sideslip, experiences some amount of oscillation in the yaw rate/sideslip before reaching steady state.



Figure 13. Streamline spiraling effect seen with large amounts of braking.

The 3D phase portrait tool facilitates examination of lateral velocity, yaw rate, and longitudinal velocity as well as determination of vehicle stability under different operating conditions. Evaluating results of various operating scenarios (i.e., braking and steering inputs) led to a better understanding of the dynamic behavior of the vehicle

cornering. The ability to visualize these dynamics serves to inform the design of a controller focused on both braking and steering inputs during turning.

### 5.2.1 Phase Portrait Validation

Upon completing the 3D phase portrait visualization tool, experimental testing was conducted at a Pennsylvania State University test track, using a Bucknell-owned steer-by-wire electric vehicle to validate the model. The P1 test vehicle, shown in Figure 14, is instrumented to provide measurable steering and braking inputs and is outfitted with a GPS system to capture vehicle velocity and vehicle attitude data. This test vehicle is rear-wheel drive and the steering system offers independently actuated left and right wheel steering. Vehicle mass and handling characteristics, presented in Table 1, are similar to those of a sports sedan, with the exception of an unusually small moment of inertia. The low yaw moment of inertia is due to the central location of the batteries. The parameters associated with this test vehicle match those used in the simulations above, experimental tests, and simulations presented throughout this paper.



Figure 14. P1 electric vehicle used during experimental testing.

Table 1. P1 Test Vehicle Parameters.

<b>Parameter</b>	<b>Symbol</b>	<b>Value</b>	<b>Units</b>
Vehicle Mass	$m$	1724	$kg$
Moment of Inertia, Yaw	$I_{zz}$	1100	$\frac{kg}{m^2}$
Wheelbase Length	$L$	2.5	$m$
Wheel Effective Rolling Radius	$R_e$	0.3085	$m$
Distance (Front Axle – CG)	$a$	1.35	$m$
Distance (Rear Axle – CG)	$b$	1.15	$m$
Track Width	$d$	0.81	$m$
Cornering Stiffness, Front	$C_{af}$	90000	$\frac{N}{rad}$
Cornering Stiffness, Rear	$C_{ar}$	138000	$\frac{N}{rad}$

Testing was conducted on a clean asphalt surface with an estimated friction coefficient of 0.75. Tests such as straight steering and cornering, with and without braking, were performed from left turns and the experimental data is compared to the theoretical model. Both stable and unstable initial conditions are considered.

For example, experimental data of a test at 8 degrees of steering without braking for a stable initial condition, as seen in Figure 15, shows trends very close to the theoretical streamline in MATLAB under matching conditions. In addition, experimental data of a test at 8 degrees of steering without braking for an unstable initial condition, as seen in Figure 16, is compared to the theoretical streamline in MATLAB under matching conditions. While the experimental vehicle trajectory diverges from the theoretical

streamline at high sideslip angles, the 3D phase portrait model suggests the same trends as those observed during experimental testing.

Evaluation of the results includes consideration of sources of error. A known source of error is un-modeled vehicle dynamics. For instance, the vehicle model assumes constant longitudinal tire slip, cornering stiffness, and tire friction. However, the experimental data reveals that longitudinal tire slip along with cornering stiffness (resulting from a sloped portion of the test track) varied throughout the test.

Additionally, the exact coefficient of friction between the tire and road is uncertain and was estimated. Instrumentation accuracy is another known source of error; in particular, the lateral velocity sensing from GPS is believed to exhibit delays and offsets. Despite unmodeled vehicle dynamics and instrumentation error, the 3D phase portrait model suggests the same trends as those observed during experimental testing and serves as a working tool for visualizing stable and unstable operating regions.

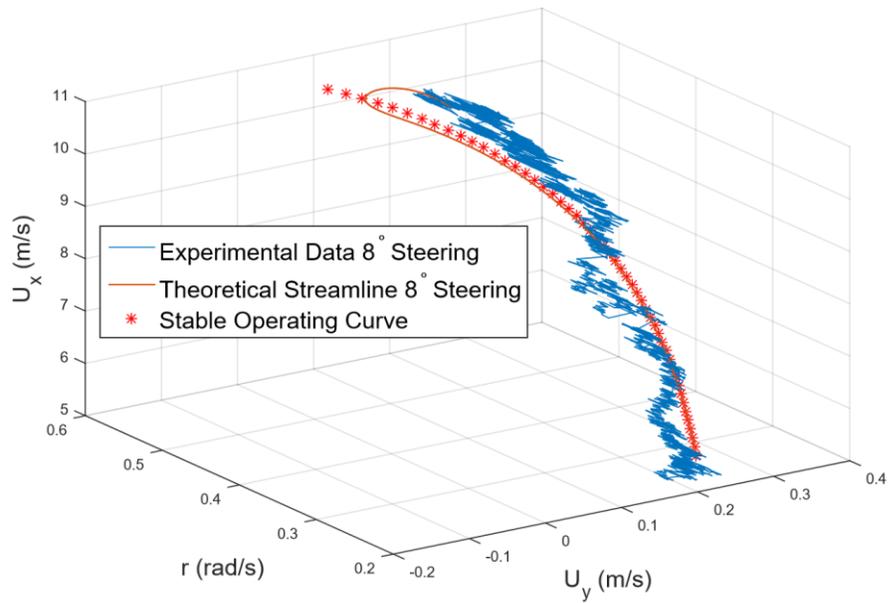


Figure 15. Comparison of experimental and theoretical data for  $8^\circ$  steering without braking ( $\kappa = 0$ ) for a stable case.

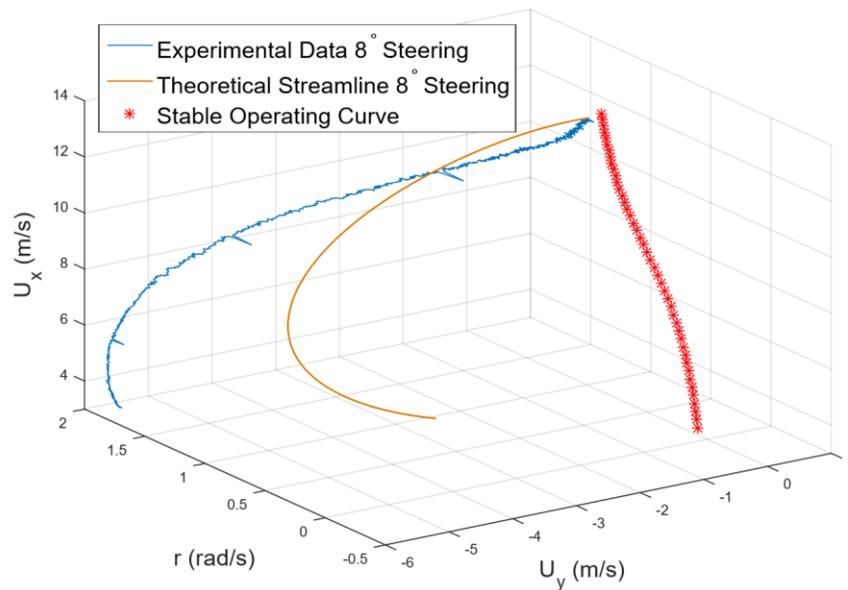


Figure 16. Comparison of experimental and theoretical data for  $8^\circ$  steering without braking ( $\kappa = 0$ ) for an unstable case.

### 5.2.2 Stable Operating Bounds

For a control system to be effective, the boundaries which keep the vehicle stable and on the roadway must be identified. The 3D phase portrait representation is used to visualize the desirable and undesirable operating regions. For a vehicle to remain stable, the steady-state acceleration of the vehicle,  $rU_x$ , must be lower than the maximum achievable tire acceleration,  $\mu g$ . Therefore, the yaw rate must satisfy equation 19,

$$|r|U_x \leq \mu g \gamma \quad (\text{Eq. 18})$$

$$|r| \leq \frac{\mu g \gamma}{U_x} \quad (\text{Eq. 19})$$

where  $\mu$  is the tire friction coefficient,  $g$  is gravity, and  $\gamma$  is a safety factor that provides conservatism when set to values less than 1.0. Additionally, the vehicle must remain on the road. Distance along a path is described by equation 20. Taking the time derivative of equation 20 results in equation 21 and produces a roadway constraint described by equation 22,

$$s_{Arc Length} = R_{Desired} \theta \quad (\text{Eq. 20})$$

$$U_x = R_{Desired} \dot{\theta}, \text{ where } \dot{\theta} = r \quad (\text{Eq. 21})$$

$$r_{Desired} = \frac{U_x}{R_{Desired}} \quad (\text{Eq. 22})$$

where  $R_{Desired}$  is the radius of the corner and  $r_{Desired}$  is for a steady-state curve.

The phase portrait for an uncontrolled vehicle at 2.5° steering angle in Figure 17 shows the stability and road geometry constraints for a 110 m cornering radius. The

vehicle states must remain below the yaw line to remain stable and on the roadway constraint line to remain on the road. In addition, the intersection of the yaw and roadway constraint lines limits the maximum desirable longitudinal cornering speed.

Setting equations 19 and 22 equal results in the following limitation on  $U_{x,max}$ :

$$U_{x,max} = \sqrt{\mu g R_{Road}} \quad (\text{Eq. 23})$$

Constant speed equilibria with constant 2.5° steering are also shown in Figure 17 for reference. Should a 110 m cornering radius be desired, a controller is needed to select the sequence of varying braking and steering inputs which drive the vehicle states from the constant speed equilibria curve to the roadway constraint curve, placing the vehicle on the intended cornering radius.

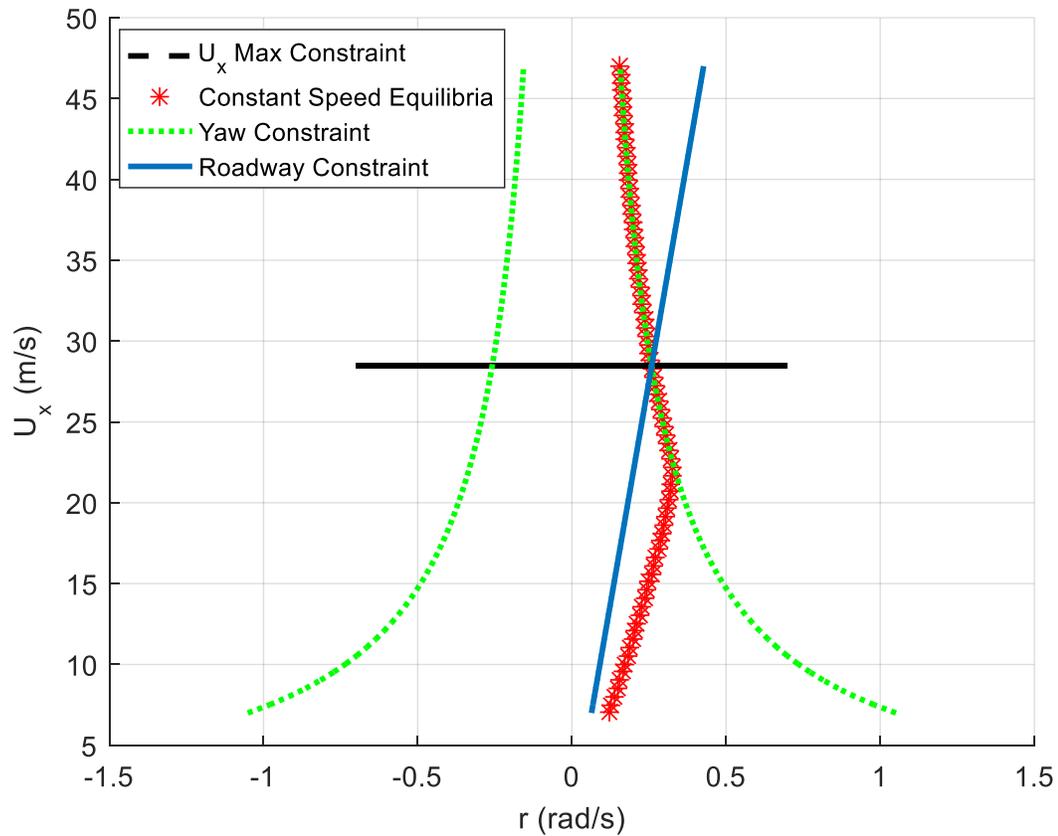


Figure 17. Phase portrait for  $2.5^\circ$  steering, no braking, and  $\mu = 0.75$ . Roadway constraint is for a 110 m radius.

It is unlikely the average driver will properly respond in the regions outside of and near the stability boundaries. A control system can complement driver actions in these regions. If the control system senses a vehicle near the stable boundaries or not on the sliding surface, corrective action can be taken to assist the driver. The insight gained from this research offers a better understanding of vehicle dynamics and stable operating boundaries, and serves as the foundation to now select a control approach appropriate for critical cornering scenarios.

## 6 Controller

### 6.1 Control Approach Selection

An effective controller for critical cornering scenarios must perform two main functions, (1) stabilize the vehicle and (2) ensure the vehicle tracks the corner geometry and remains on the road. Four different common control approaches, Model Predictive Control (MPC), Proportional–Integral–Derivative (PID) control, H-Infinity control, and sliding surface control, were considered for this purpose. The advantages and disadvantages of each control approach were examined to select most appropriate approach for cornering maneuvers.

#### 6.1.1 MPC Control

MPC is a straightforward approach, based on well-understood concepts [Gorinevsky 2016]. It performs an optimization to find the best sequence of inputs to drive the vehicle states to the desired operating point and is advantageous because of its ability to consider future time steps. Further, the dynamics of the vehicle are built into the optimization problem. Therefore, depending on the cost function, the control effort may be more likely to follow the natural dynamics of the vehicle. In contrast, a control approach which does not anticipate future events and take appropriate control action could result in the control effort opposing the natural dynamics of the vehicle. A

controller which is less likely to fight the natural dynamics also means there is a smaller chance that the actuator will saturate.

The MPC approach is ideal when there are multiple inputs and outputs, as is the case for the controller being designed [Gorinevsky 2016]. In addition, it explicitly handles constraints on the system and is easy to maintain as a change in the model does not necessitate complete redesign [Gorinevsky 2016]. While MPC is not necessarily more or less robust than classical feedback, it can be more easily adjusted for robustness than classical feedback [Arkun 1988].

MPC is more reliable with convex optimization problems. When certain nonlinearities are introduced, issues can arise because the problem may no longer be a convex optimization problem [Jaganmohan 2014]. Convex objective functions include only one minimum, as shown in Figure 18a, making it easy for the optimization algorithm to identify the optimal point. Nonconvex objective functions on the other hand include more than one local minima, as shown in Figure 18b, creating potential for the optimizer to stop at a local minima instead of the global minima or true optimal point.

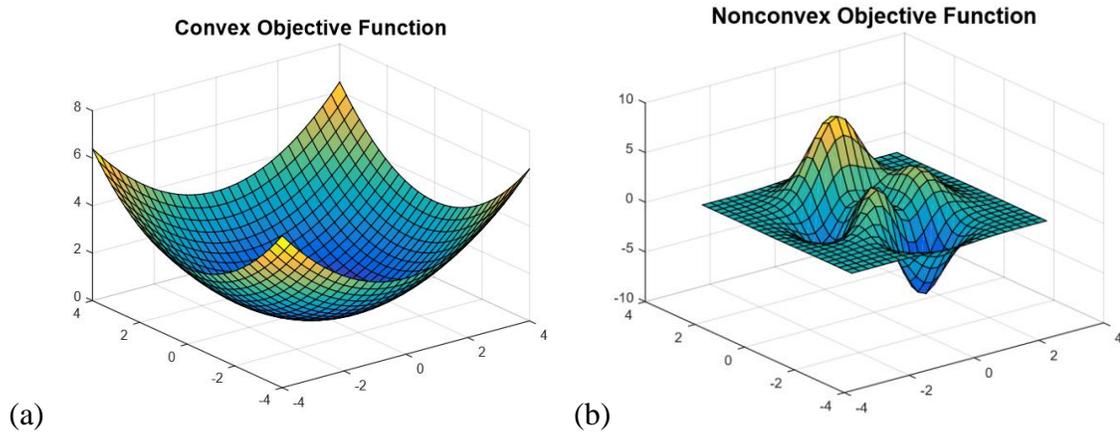


Figure 18. Convex objective function with one minimum versus a nonconvex objective function with multiple minima.

### 6.1.2 PID Control

PID control is one of the best understood control approaches as it has a long history and is commonly used in industrial applications. The proportional, derivative, and integral gains must be tuned for a controller to be robust, as each gain has its advantages and disadvantages. Proportional control reduces problems caused by disturbances, but increases response time and may result in a large offset in the tracking performance which may not bring the system to the desired point. Integral control is useful in eliminating steady state error, but may produce significantly slower response times [“An Introduction...” 2016]. Derivative gain reduces oscillation and overshoot, but amplifies any high frequency noise present in a system. Therefore all three gains must be balanced for robust performance.

A disadvantage of PID controllers is that they do not have the ability to consider future time steps such as that of Model Predictive Control. However, feedback loops take

the system output into consideration at each time step and enable the system to adjust to meet the desired output response. The major drawback to the use of PID control for the cornering application is that the controller would require multiple PID controllers working together as there are multiple inputs/outputs. Difficulties can arise with the feedback loop when multiple inputs/outputs are introduced because it is more challenging to reduce multiple errors at the same time.

There is also potential, in cornering, for the actuator, or more likely the tire forces, to become saturated with PID control. If the actuator saturates, the feedback loop is broken and the system runs in open loop as long as the actuator remains saturated [Honeywell 2016]. If the tire forces reach their limit the vehicle may lose traction and spin out. It is also important to note that PID control was designed for linear systems while the algorithms used in this vehicle dynamics model are nonlinear. PID control can be used in the presence of nonlinearities, but is not as easy in this case, as limitations apply. While, at first glance, PID is an easy controller to implement, another control approach may be better suited for cornering applications.

### 6.1.3 H-Infinity Control

In H-Infinity control, a problem is set up as a mathematical optimization problem and a controller which subsequently solves the optimization problem is found. One advantage of H-Infinity control is that it is readily applicable to problems involving multivariate systems [Skogestad 2007]. However, a disadvantage of this approach is the

experience and background needed to successfully design this type of controller, as the mathematical functions and theory are highly complex. The model of the system must be fairly accurate. Additionally, nonlinear constraints, such as saturation, are not usually handled very well [Helton 1999]. Simultaneously optimizing robust performance and robust stabilization, both of which are needed in this application, is also challenging. H-Infinity loop-shaping is one method that can be used to overcome this, but it adds further complexity [Glover 1992]. While the H-Infinity control approach could be effective in this application, it may require more time to learn the background theory than is available.

#### 6.1.4 Sliding Surface Control

Sliding surface is a control approach which defines a stable sliding surface and drives all states to this desired operating surface. This control approach works well with nonlinear systems, with multiple inputs/outputs, and has low complexity. The governing equations for this problem are defined by simple equations. Sliding surface control entails two phases: (1) the reaching phase where the vehicle trajectory is driven to the sliding surface and (2) the sliding phase where the vehicle trajectory is controlled to slide along the sliding surface [Barzamini 2011]. An advantage of sliding surface control is its robustness. For example, once the sliding surface is reached, the system dynamics are governed entirely by the sliding surface parameters and not the original system parameters. In addition, sliding surface dynamics are independent of disturbances and

model uncertainties, contributing to the robustness of a sliding surface controller [Barzamini 2011].

One undesirable effect sometimes seen in sliding surface control is a phenomenon known as chattering, as shown in Figure 19. Ideally, once the sliding surface is reached, the vehicle will smoothly slide along this surface. In reality though, the control switch exhibits delay and continually switches the control signal about the sliding surface, leading to unwanted high frequency oscillations [Janardhanan 2016]. Chattering can deteriorate the controller performance and have a harmful effect on actuators, particularly if high frequency oscillations cause mechanical wear on the system. Although chattering is a problem, various methods can be used mitigate or eliminate it. One such method is to replace the discontinuous “sign” term in the sliding phase control law with a continuous smooth approximation; however, this approach can result in loss of robustness and is only effective in the specific case when there are no uncertainties in the system [“A Quick...” 2016]. An alternative method to using the smooth approximations is to use higher order sliding control algorithms. These algorithms reduce the chattering issue without compromising the robustness, but are more complex mathematically.

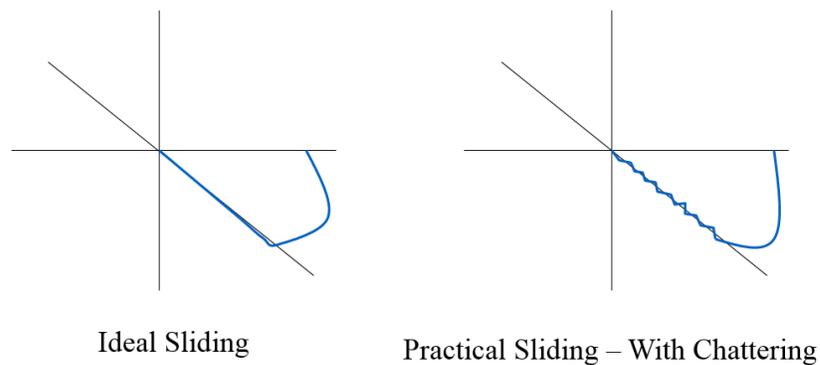


Figure 19. Ideal sliding phase versus a practical sliding phase with chattering [Janardhanan 2016].

While necessary to consider, chattering may not be an issue with this control system as the focus is on a non-intrusive driver assistance system. The goal of this controller is to get the vehicle onto the sliding surface at which point it is assumed the driver will be able to keep the vehicle within safe operating bounds. Increased issues with chattering could arise if further advancements of this proposed control system, such as an extension to an autonomous vehicles are pursued.

Another disadvantage of sliding surface control is that it does not have predictive ability such as that of MPC. Instead, it pushes the states toward the desired operating point at all instances which may result in the controller overriding the natural dynamics of the vehicle. Overriding the natural dynamics could lead to saturation of the actuator as larger inputs are needed.

Selection of a sliding surface for the trail braking application is relatively straightforward. A natural selection for the surface is the plane that keeps the vehicle on

the desired roadway curvature. This plane is defined by the yaw rate and longitudinal velocity, as described in Equation 24.

$$r_{Desired} = \frac{U_x}{R_{Desired}} \quad (\text{Eq. 24})$$

### 6.1.5 Selection Matrix (Methodology)

A selection matrix is used to arrive at the best control approach for critical cornering scenarios.

#### 6.1.5.1 Identify Evaluation Criteria & Weighting Evaluation Criteria

The set of evaluation criteria, identified in Table 2, are ordered and assigned a weighting factor, based on their relative importance to controller performance. The criteria are prioritized using a weighting ranging from 1 to 3 (3 – Most important; 2 – Important; 1 – Least Important). The three criteria weighted “3 - Most Important” for a controller approach are its ability to handle nonlinearities, its tracking performance, and its robustness against parameter uncertainties and disturbances. The 3D phase portrait model utilizes nonlinear dynamic algorithms based on the four wheel vehicle model and combined tire model so it is essential that the selected control approach be compatible with nonlinearities. Additionally, the controller must have high performance, ensuring the system tracks the desired cornering radius and does not run off the road. The controller must also remain robust against any parameter uncertainties and disturbances

encountered, ensuring the vehicle remains stable. These criteria are most important because the intended controller is for a nonlinear, cornering controller. In addition to these criteria, other criteria are necessary for a well performing controller. Five additional criteria are weighted “2 - Important”. They are reduction of chattering or oscillations, compatibility with multivariate systems, ability to handle constraints (such as actuator or tire limits), reduction of noise, and ability to predict future positions. The mathematical simplicity of the control approach, while weighted “1 - Least Important”, is included because selection of a highly complex control approach would require added time to learn the background control knowledge. Each control approach is then ranked for each of the weighted criteria, using a rank ranging from 1 to 3 (3 – Best; 2 – Neutral; 1 – Worst) as shown in Table 2.

Table 2. Selection matrix compares potential control approaches. Control approaches are ranked for each of the weighted evaluation criteria.

	<b>CRITERIA</b>	<b>Weight</b>	<b>H Infinity Control</b>	<b>Sliding Surface Control</b>	<b>Model Predictive Control</b>	<b>PID Control</b>
			Rank	Rank	Rank	Rank
	<b>Most Important</b>					
1	Ability to handle nonlinearities	3	2	3	1	1
2	Tracking performance	3	2	3	2	2
3	Robustness against parameter uncertainties and disturbances	3	2	3	2	2
	<b>Important</b>					
4	Reduction of chattering/oscillations	2	2	1	2	2
5	Compatibility with multivariate systems	2	3	3	3	2
6	Ability to handle constraints (ex. saturation of actuator or tire force)	2	3	2	2	2
7	Reduction of noise	2	2	2	2	2
8	Ability to predict future positions	2	1	1	3	1
	<b>Least Important</b>					
9	Mathematical simplicity of approach	1	1	2	2	3

#### 6.1.5.2 Score Control Approach against Evaluation Criteria

A score for each control approach, against each criterion, is found by multiplying its ranking by the respective weighting for the given criteria. The scores, for all criteria are totaled, for each concept design approach as shown in Table 3, and the design approach with the highest score is identified.

Table 3. Selection matrix compares potential control approaches. Control approaches are scored for each criteria. The highest score represents the best suited control approach - highlighted in dark grey.

	<b>CRITERIA</b>	<b>H Infinity Control</b>	<b>Sliding Surface Control</b>	<b>Model Predictive Control</b>	<b>PID Control</b>
		Score	Score	Score	Score
	<b>Most Important</b>				
1	Ability to handle nonlinearities	6	9	3	3
2	Tracking performance	6	9	6	6
3	Robustness against parameter uncertainties and disturbances	6	9	6	6
	<b>Important</b>				
4	Reduction of chattering/oscillations	4	2	4	4
5	Compatibility with multivariate systems	6	6	6	4
6	Ability to handle constraints (ex. saturation of actuator or tire force)	2	4	4	4
7	Reduction of noise	4	4	4	4
8	Ability to predict future positions	2	2	6	2
	<b>Least Important</b>				
9	Mathematical simplicity of approach	1	2	2	3
	<b>TOTAL SCORE</b>	37	<b>47</b>	41	36

The Sliding Surface Control approach received a score of 47, whereas the other three control approaches received scores from 36-41. Therefore, the Sliding Surface

Control approach is selected as the best suited control approach for the cornering application.

## 6.2 Controller Design

The selected controller design has a hierarchical structure; a path following function determines the target cornering radius and sliding surface control is used to define a stable sliding surface and drive the vehicle to this surface. The target cornering radius is identified based on the position and orientation of the vehicle relative to the desired path. Sliding surface control is then used to determine the braking and steering inputs which drive the vehicle states toward the selected cornering radius. The global states of the vehicle at the next instant in time are then measured and used in the feedback loop. A flow chart, outlining the control structure, is illustrated in Figure 20.

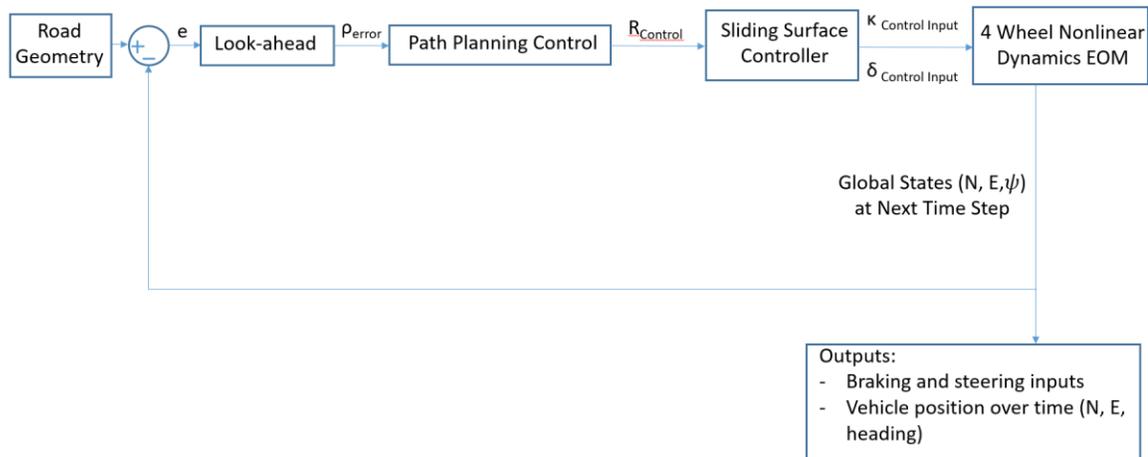


Figure 20. Hierarchical controller structure.

### 6.2.1 Path-Following Function

The path-following function identifies when a vehicle deviates from the roadway, and tightens or expands the target cornering radius to bring the vehicle back on path. The vehicle position at a future time step,  $N_{d_{LA}}$  and  $E_{d_{LA}}$ , can be predicted using speed dependent look-ahead:

$$d_{LA} = \kappa_{d_{LA}} U_x * dt \quad (\text{Eq. 25})$$

$$\begin{bmatrix} N_{d_{LA}} \\ E_{d_{LA}} \end{bmatrix} = g(d_{LA}(U_x), N, E) \quad (\text{Eq. 26})$$

where  $dt$  is the time step,  $\kappa_{d_{LA}}$  applies a gain to a constant time look-ahead, and  $N$  and  $E$  are the north and east global positions of the vehicle, respectively. In other terms,  $d_{LA}$  corresponds to the distance traveled in a fixed amount of time, as shown in Figure 21.

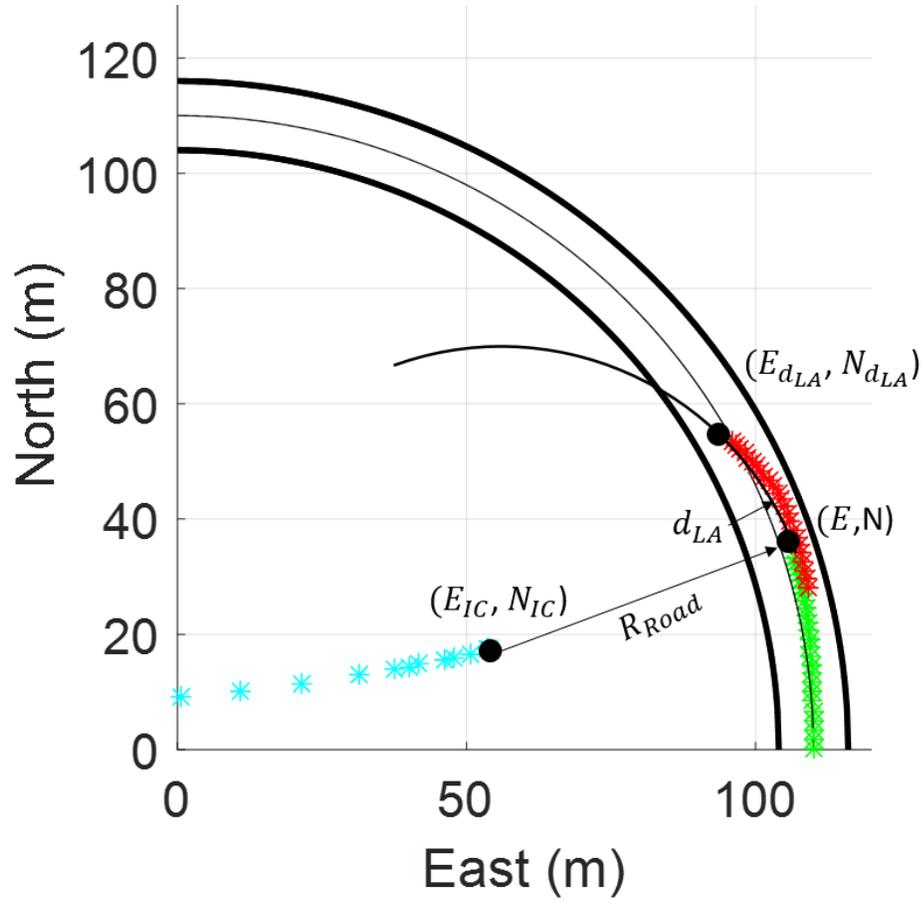


Figure 21. Speed dependent look-ahead.

There are many ways of determining  $N_{dLA}$  and  $E_{dLA}$  including GPS maps and camera-based look-ahead. Regardless of the approach utilized to determine the projected vehicle position, the lateral error associated with the predicted position can be expressed as follows:

$$e_{LA} = \sqrt{(E_{dLA} - E_{IC})^2 + (N_{dLA} - N_{IC})^2} - R_{Road} \quad (\text{Eq. 27})$$

where  $R_{Road}$  is the instantaneous radius of the road and  $N_{IC}$  and  $E_{IC}$  are the coordinates of the instantaneous center of curvature, as shown in Figure 21. The current curvature

error can then be determined based on the lateral error associated with the vehicle's projected position.

$$\rho_{Error} = \frac{1}{R_{Road}} - \frac{1}{(R_{Road} + (e_{LA}))} \quad (\text{Eq. 28})$$

Curvature error, as opposed to radius error, is used because curvature can be negative or positive to express the concavity of the desired bend and passes smoothly through zero as the concavity changes, as seen in Figure 22.

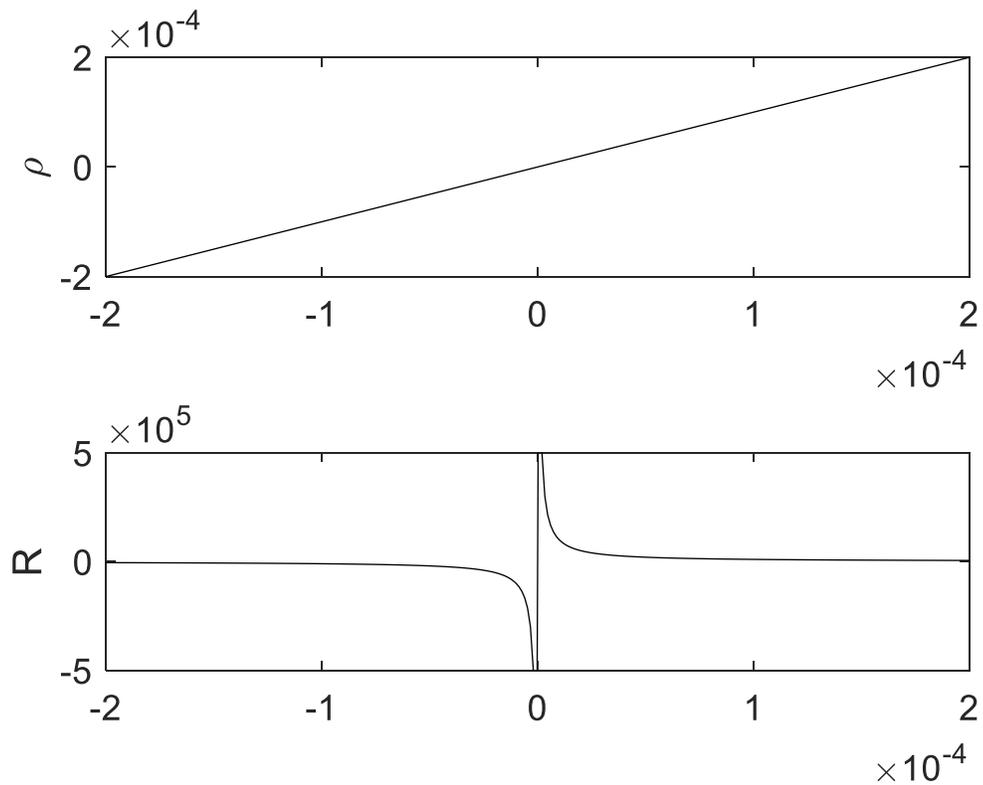


Figure 22. Behavior of curvature error and radius error.

The target cornering radius, based on the road curvature and curvature error, is given by:

$$R_{Control} = \frac{1}{\left(\frac{1}{R_{Road}} + K_{\tau}\rho_{Error}\right)} \quad (\text{Eq. 29})$$

where  $K_{\tau}$  weights the curvature error. Lower level control can then be used to drive the vehicle to this target cornering radius.

### 6.2.2 Sliding Surface Control

Sliding surface control is used as the lower-level control. The goal is to drive the vehicle dynamics to the sliding surface at which point it is assumed the driver will be able to keep the vehicle within safe operating bounds. Selection of a sliding surface for cornering applications is straightforward, following the development in section 6.1.4. The vehicle must remain stable as well as track the road curvature while cornering. A natural selection for the sliding surface is the plane which keeps the vehicle on the target cornering radius determined by the path-following function. This plane is defined by the yaw rate and longitudinal velocity as described in equation 30.

$$r_{Desired} = \frac{U_x}{R_{Control}} \quad (\text{Eq. 30})$$

This condition is independent of vehicle sideslip and can be extended to a planar surface, forming the stable sliding surface. For example, given a target cornering radius of 110 m, the sliding surface appears as shown in Figure 23.

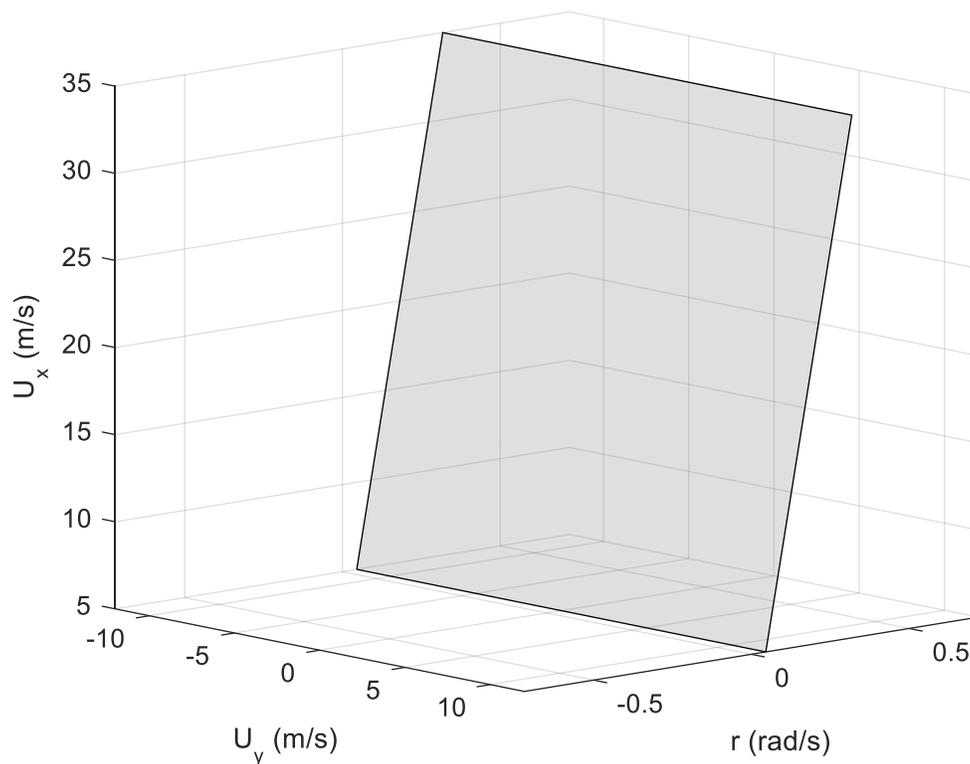


Figure 23. Sliding surface for a cornering radius of 110 m.

The dynamics of the sliding surface are defined by:

$$S = K_r(r - r_{Desired})^2 + K_\beta \left(\frac{U_y}{U_x}\right)^2 + K_{U_x}(U_x - U_{x,max})^2 \quad (\text{Eq. 31})$$

$$\begin{aligned} \dot{S} = & 2K_r \left(r - \frac{U_x}{R_{Control}}\right) \left(\dot{r} - \frac{\dot{U}_x}{R_{Control}}\right) + 2K_\beta \left(\frac{U_y}{U_x}\right) \left(\frac{\dot{U}_y}{U_x} - \frac{\dot{U}_x U_y}{U_x^2}\right) \\ & + 2K_{U_x}(U_x - U_{x,max})\dot{U}_x \end{aligned} \quad (\text{Eq. 32})$$

where  $S = 0$  corresponds to vehicle states which lie on the safe sliding surface. The surface value,  $S$ , can be thought of as the distance or error between the actual vehicle

state and the safe sliding surface. Figure 24 shows vehicle states being driven to the safe sliding surface for many different initial conditions. Notice that the trajectories eventually converge to a stem on the sliding surface. This stem corresponds to the series of  $U_y$  and  $r$  equilibrium points that lie on the plane at each longitudinal velocity. A sequence of varying braking and steering inputs, having  $U_y$  and  $r$  equilibrium points that match up with the stem, is selected to drive the vehicle to this sliding surface.

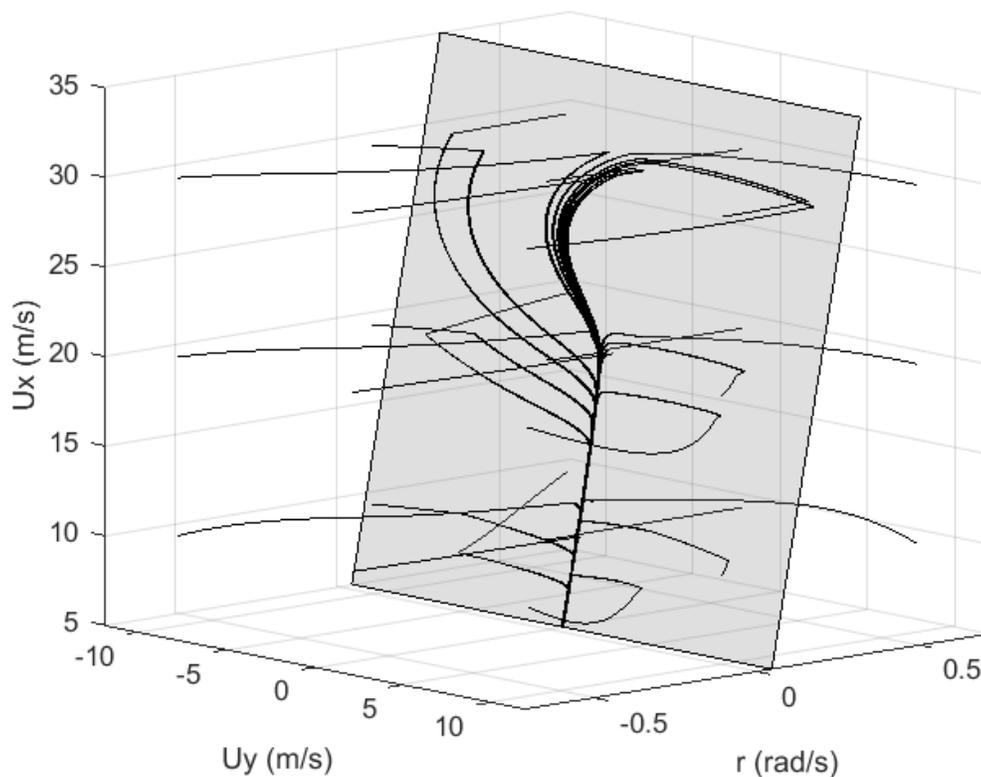


Figure 24. Vehicle trajectories converge to a safe sliding surface.

Yaw, sideslip  $\left(\frac{U_y}{U_x}\right)$ , and longitudinal velocity error terms are all included in the surface value function,  $S$ . The yaw cost term drives the vehicle to the target cornering

radius  $R_{Control}$  while the sideslip cost term prevents the vehicle from sliding laterally due to large  $U_y$ . The longitudinal velocity cost term causes the controller to reduce unnecessary braking by keeping  $U_x$  near the  $U_{x,max}$  boundary. Each error term is weighted with a gain as needed. Both the yaw and sideslip error terms are necessary to ensure the vehicle stays on the road. As a result, the gains on these error terms are much larger than the gain on the longitudinal velocity error term, which merely tries to keep the vehicle near to the driver's intended speed.

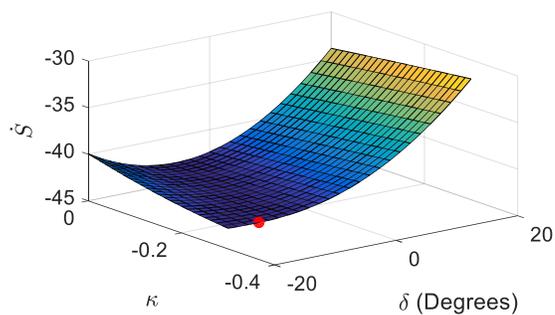
The feasible regions of control are those where  $S$  and  $\dot{S}$  are oppositely signed. For example when  $S$  is positive,  $\dot{S}$  must be negative to drive  $S$  to 0. Conversely, when  $S$  is negative,  $\dot{S}$  must be positive to drive  $S$  to 0. Because each state term in  $S$  is squared,  $S$  is always positive so the minimum (most negative) value of  $\dot{S}$  is always desired. Based on this approach, an optimization routine is developed to determine the braking and steering inputs that result in a minimum  $\dot{S}$ . The range of the braking tire slip input,  $\kappa$ , is constrained between -0.3 and 0 and the range of the steering input,  $\delta$ , is constrained between -20 degrees and 20 degrees. Most passenger vehicles are capable of achieving both of these input ranges.

Production stability control systems stabilize the vehicle by independently braking individual wheels. This creates a yaw moment that controls the orientation of the vehicle. The sliding surface controller, however, is designed for equal braking on all four wheels, as is the case with trail braking in racing. This has the primary effect of slowing the vehicle down. Alternative formulations, in which the controller has the authority to brake all four wheels independently, are also possible.

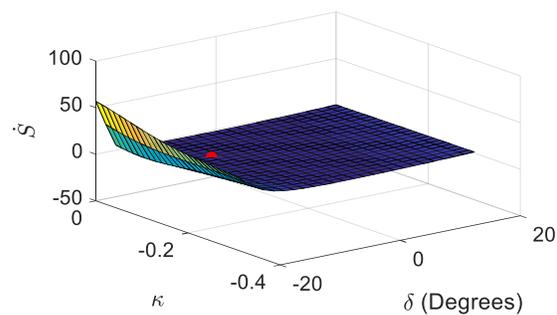
When developing the control structure, consideration is also given to accurate performance and feedback time. Surface plots of  $\dot{S}$  are created by calculating  $\dot{S}$  at a grid of steering and braking inputs; this enables evaluation of surface convexity. By inspection, it can be seen that some  $\dot{S}$  surfaces are nonconvex. Figure 20 shows various surfaces encountered across several  $U_y, r, U_x$  initial conditions, many of which are nonconvex. Nonconvex surfaces may include more than one local minima creating potential for an optimizer to stop at a local minima rather than a true optimal point; therefore an efficient, yet accurate, optimization technique is desired. The controller is designed to evaluate  $\dot{S}$  across a grid of steering and braking inputs at each time step and to select the control inputs corresponding to the minimum  $\dot{S}$ . This approach results in the most accurate computation. Initial results on a desktop computer indicate feasibility for real-time implementation of this optimization.

The accuracy of the computation method for obtaining controller inputs is confirmed by plotting the calculated controller inputs on the surface plots of  $\dot{S}$  for several vehicle states. Figure 25 illustrates that the calculated control inputs correspond to the respective minimum on each plot. This confirms that the controller accurately calculates the control inputs which correspond to a minimum  $\dot{S}$  at each respective vehicle state.

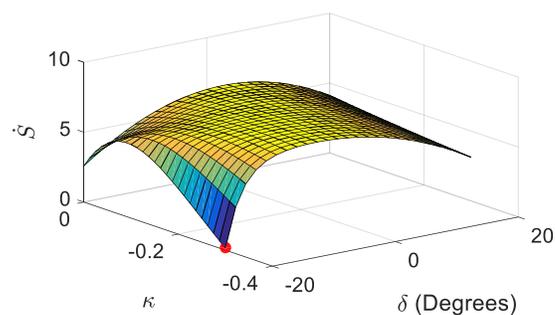
$U_y = -10$  m/s,  $r = -0.6$  rad/s,  $U_x = 10$  m/s  
 ● Minimum  $\dot{S}$



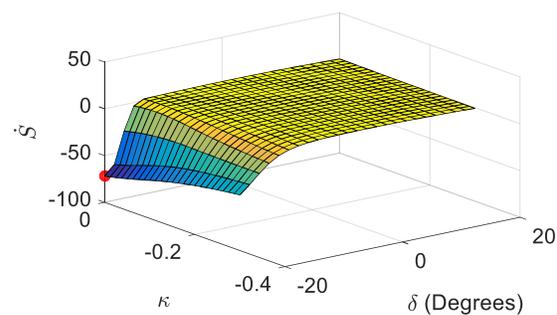
$U_y = -10$  m/s,  $r = 0$  rad/s,  $U_x = 30$  m/s  
 ● Minimum  $\dot{S}$



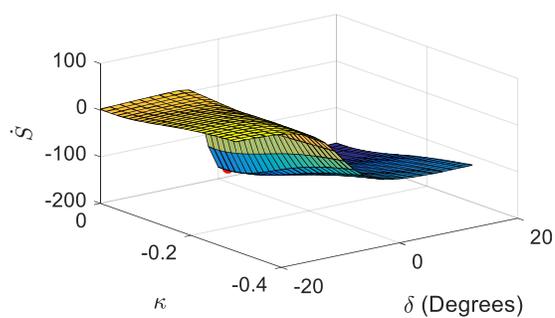
$U_y = -10$  m/s,  $r = 0.6$  rad/s,  $U_x = 20$  m/s  
 ● Minimum  $\dot{S}$



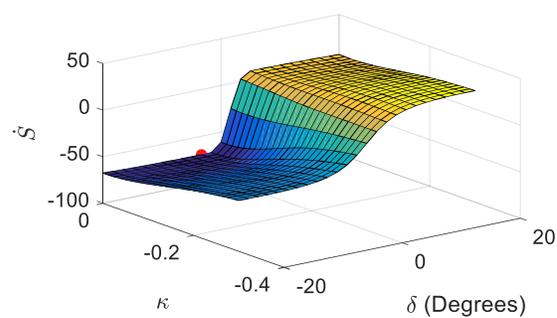
$U_y = -10$  m/s,  $r = 0.6$  rad/s,  $U_x = 30$  m/s  
 ● Minimum  $\dot{S}$



$U_y = 0$  m/s,  $r = -0.6$  rad/s,  $U_x = 20$  m/s  
 ● Minimum  $\dot{S}$



$U_y = 0$  m/s,  $r = 0.6$  rad/s,  $U_x = 30$  m/s  
 ● Minimum  $\dot{S}$



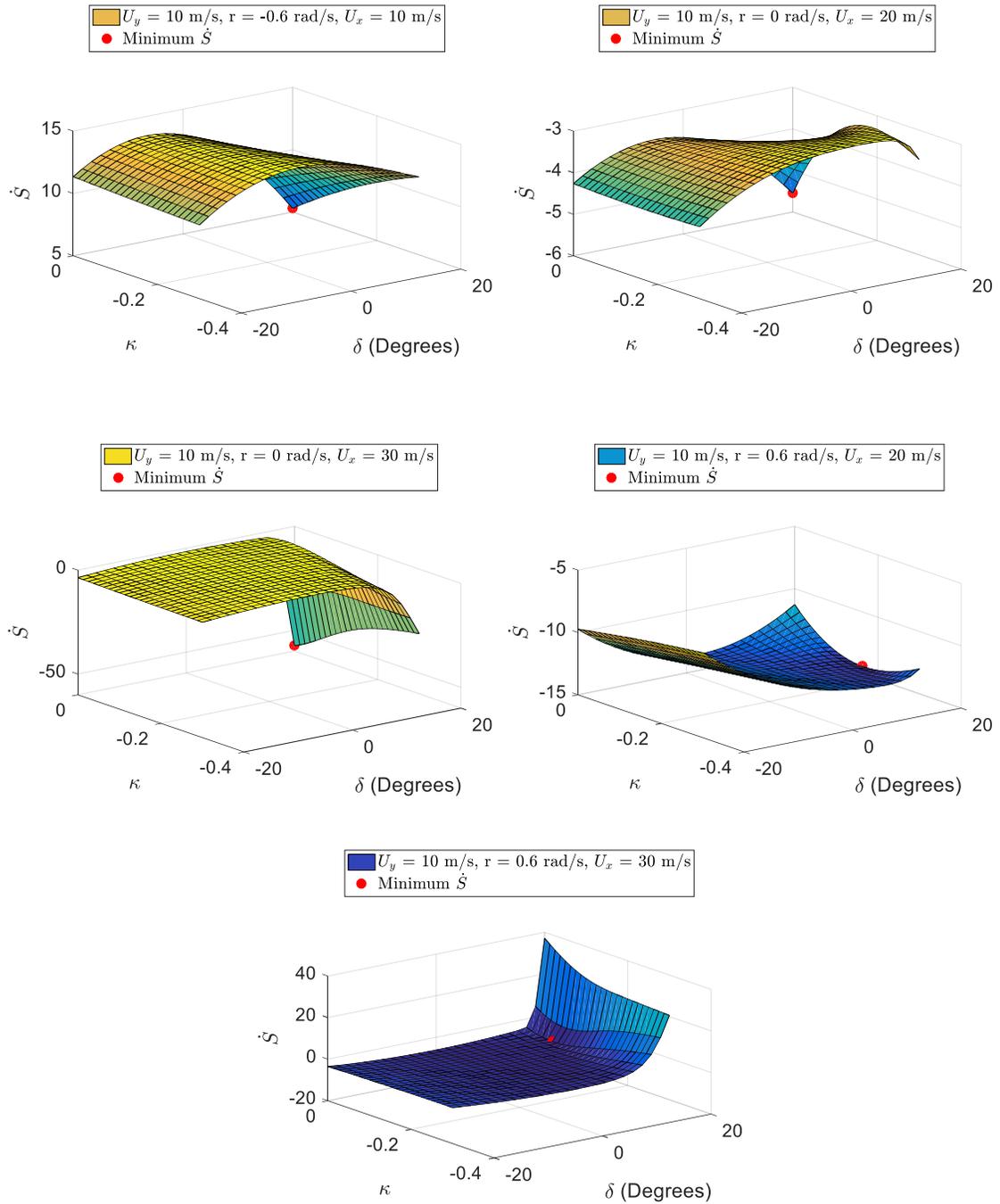


Figure 25. Surface plots of  $\dot{S}$  at several  $U_y$ ,  $r$ ,  $U_x$  vehicle states along with the calculated control inputs indicated in red.

The feasible regions of control, where  $S$  and  $\dot{S}$  can be oppositely signed, are also plotted to aid in visualizing the portions of the phase plane that are controllable. Given the set constraints on available braking and steering angles, Figure 26 presents a 3D phase portrait showing the feasible and infeasible regions of control for a 110 m cornering radius with  $\mu = 0.4$ . Figures 27 and 28, are slices of the 3D phase portrait which show the feasible regions of control at  $U_x = 10$  m/s and  $U_x = 20$  m/s.

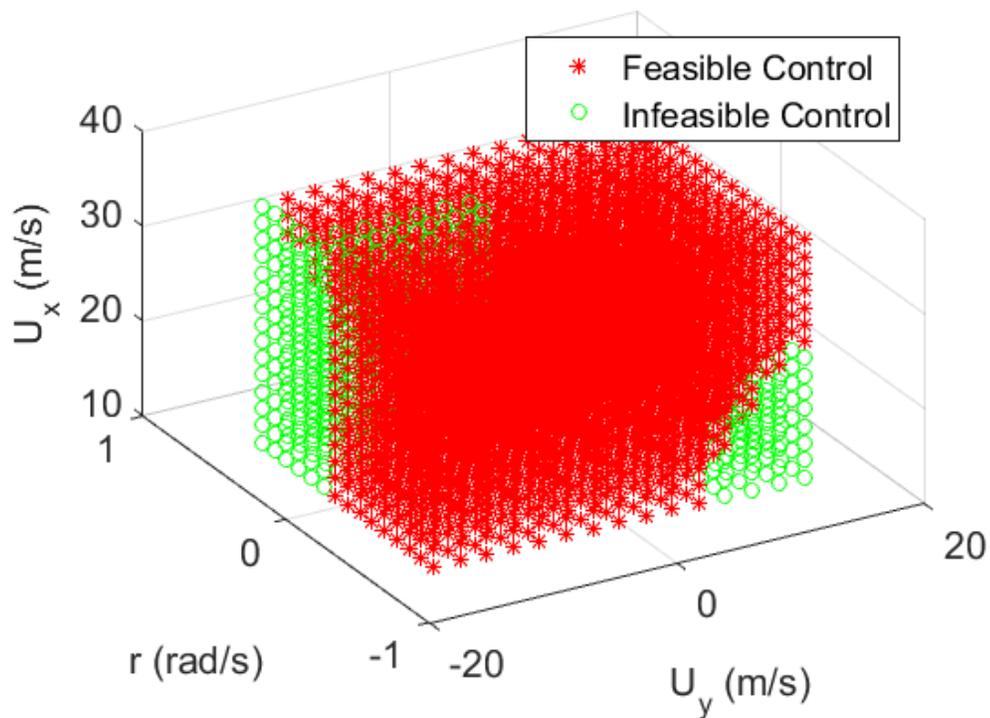


Figure 26. Feasible regions of control in the 3D phase portrait for a 110 m cornering radius,  $\mu = 0.4$ ,  $K_r = 28$ ,  $K_\beta = 50$ , and  $K_{U_x} = 0.02$ .

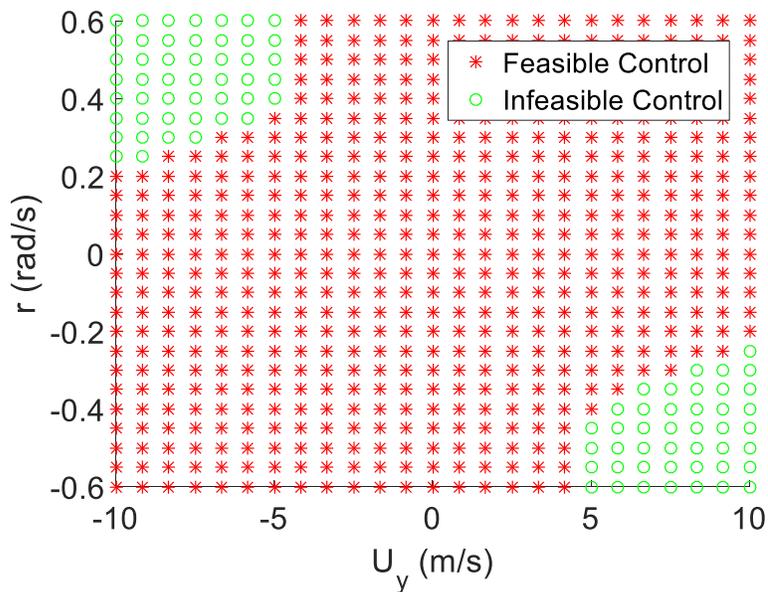


Figure 27. Slice of 3D phase portrait, showing feasible regions of control at  $U_x = 10$  m/s, for  $R = 110$  m,  $\mu = 0.4$ ,  $K_r = 28$ ,  $K_\beta = 50$ , and  $K_{U_x} = 0.02$ .

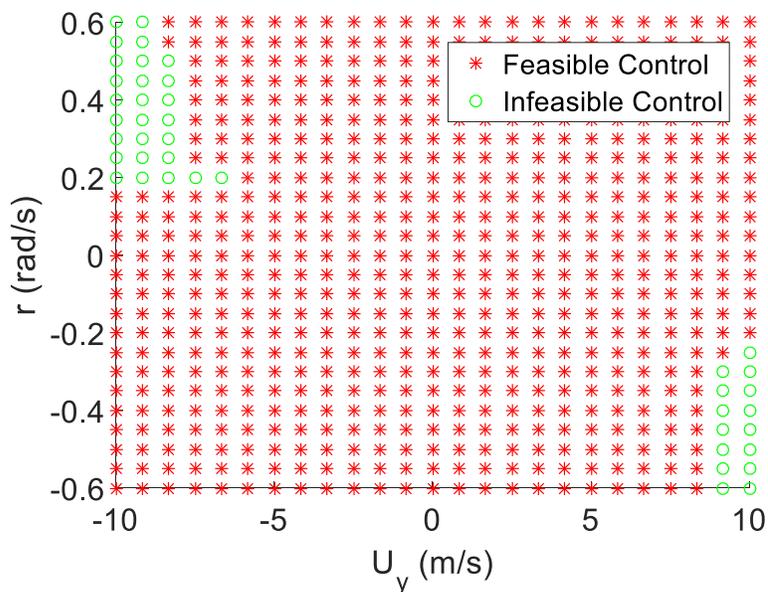


Figure 28. Slice of 3D phase portrait, showing feasible regions of control at  $U_x = 20$  m/s, for  $R = 110$  m,  $\mu = 0.4$ ,  $K_r = 28$ ,  $K_\beta = 50$ , and  $K_{U_x} = 0.02$ .

A large portion of the phase portrait is characterized by feasible control. However, there are small regions where the controller is not able to assist the driver. These regions correspond to regions of large yaw and lateral velocities, where vehicles tend to slide out. The controllable and uncontrollable regions in Figures 27 and 28 are similar to the stable and unstable regions presented in the 2D phase portraits by Inagaki. However, Figure 26 presents the closed-loop stabilizable regions; whereas, the 2D phase portraits by Inagaki only show the open-loop stable regions. It is important to note that regions of feasible control are regions where the controller can drive the vehicle closer to the sliding surface. It is not guaranteed, however, that the controller is able to drive the vehicle to the sliding surface within the time or lane width available.

Following selection of the optimized control inputs, the global position of the vehicle is determined. In simulation, the global positions are found using an Euler approximation, where the global position derivatives are described by equations 33-35.

$$\dot{\psi} = r \quad (\text{Eq. 33})$$

$$\dot{N} = U_x \cos \psi - U_y \sin \psi \quad (\text{Eq. 34})$$

$$\dot{E} = -U_y \cos \psi - U_x \sin \psi \quad (\text{Eq. 35})$$

where  $\psi$  is the vehicle heading, N is the northern component of the vehicle position, and E is the eastern component of the vehicle position. In practice, the global states are measured with a GPS system or another localization technique. The global states of the

vehicle at the subsequent time step are then fed back into the controller, closing the loop on the design.

## 7 Simulated Results

Validation of the hierarchical sliding surface controller is accomplished by comparing its respective simulation results to those of a baseline controller, to gain a sense of how well the designed sliding surface controller performs. The expectation is that the combined steering and braking controller performs better than the baseline controller. Furthermore, the vehicle response across various friction coefficients, vehicle speeds, and cornering radii are simulated to demonstrate robustness of the hierarchical sliding surface controller. Finally, the sequence of steering and braking inputs applied to maintain control during cornering is compared to those of trail braking and the last possible safe braking point is presented.

### 7.1 Baseline Controller Results

The chosen baseline controller uses proportional control with a simple look-ahead to represent a simple model of a human driver tracking a roadway. The baseline controller only involves a steering input (no braking) and responds to lateral error and heading error.

To develop this model, the position of the car is represented in terms of its coordinates relative to a road, as shown in Figure 29 and equations 36-38 below.

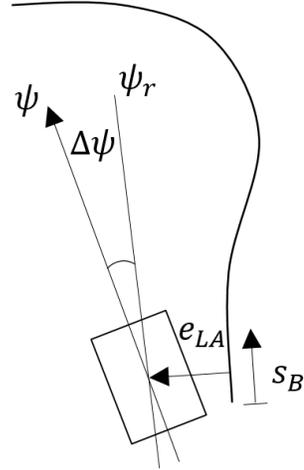


Figure 29. Position of the vehicle relative to the roadway path.

$$\dot{s}_B = U_x \cos \Delta\psi - U_y \sin \Delta\psi \quad (\text{Eq. 36})$$

$$\dot{e}_{LA} = U_y \cos \Delta\psi + U_x \sin \Delta\psi \quad (\text{Eq. 37})$$

$$\Delta\dot{\psi} = \dot{\psi} - \dot{\psi}_r = r - \frac{\dot{s}}{R_{Road}} \quad (\text{Eq. 38})$$

where  $s_B$  is the distance along the path,  $e_{LA}$  is the lateral distance from the vehicle CG to the path,  $R_{Road}$  is the road radius, and  $\Delta\psi$  is the heading difference between the vehicle and roadway. Based on the lateral and heading error, the steering input necessary to track the road can be determined from equation 39, where  $K_e$  weights the total lateral error.

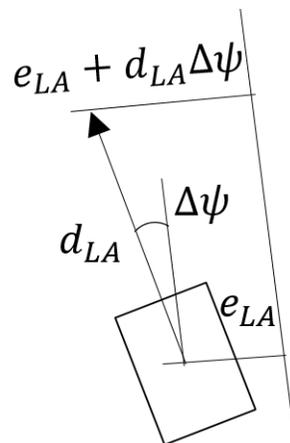


Figure 30. Position of the vehicle relative to the roadway path.  $d_{LA}$  represents the look-ahead distance, 10 m.

$$\delta = -K_e(e_{LA} + d_{LA}\Delta\psi) \quad (\text{Eq. 39})$$

The baseline controller is assessed under the following conditions shown in Table 4.

Table 4. Baseline Controller Simulation Inputs.

Baseline Controller - Inputs for Simulations								
	<b>R</b>	$U_{x,initial}$	$U_{y,initial}$	$r_{initial}$	$d_{LA}$	$\kappa_e$	$\mu$	$U_{x,max}$
	<b>m</b>	<b>m/s</b>	<b>m/s</b>	<b>rad/s</b>				<b>m/s</b>
<b>Baseline Controller</b>	110	28	0	0	10	0.2	0.3	18.0
							0.4	20.8
							0.7	27.5
							0.9	31.2

The vehicle position in the world with the baseline controller is shown in Figure 31. The  $\mu = 0.9$ ,  $U_{x,max} = 31.2$  m/s trajectory tracks the desired roadway very closely. The  $\mu = 0.7$ ,  $U_{x,max} = 27.5$  m/s trajectory begins to exhibit some lateral error, but still follows the desired corner moderately well. However, trajectories with low coefficients

of friction and a  $U_{x,max}$  well below 28 m/s ( $\mu = 0.3$  and  $\mu = 0.4$ ) remain far from reaching the roadway. Figure 31 shows that a steering-only controller works well for trajectories with initial conditions within the stable operating bounds. As expected, steering alone is insufficient to stabilize a vehicle outside of the bounds illustrated in Figure 17.

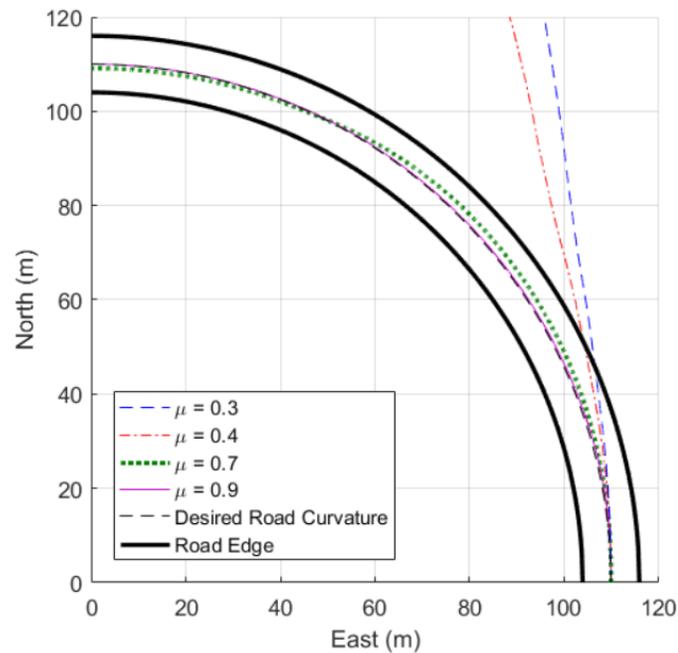


Figure 31. Vehicle response with baseline controller for  $R = 110$  m,  $U_{x,initial} = 28$  m/s,  $U_{y,initial} = 0$  m/s,  $r_{initial} = 0$  rad/s,  $K_e = 0.2$ ,  $d_{LA} = 10$ .

## 7.2 Sliding Surface Controller Results

The vehicle response with the hierarchical sliding surface control is simulated, using the code shown in Appendix B, and compared to that of the baseline controller. Simulations with various friction coefficients, initial longitudinal speeds, and cornering radii are performed to demonstrate the robustness and effectiveness of the sliding surface

controller across a wide range of parameters. Gains are chosen to: minimize overshoot and keep the vehicle within a standard lane width of 12 ft. and drive the vehicle to the desired corner. The five gains in the control structure are determined by trial and error until gains which result in desirable performance are found.

An in-depth analysis is performed to understand the effects of varying each gain. The controller gains are independently varied to understand how each affects tracking and overshoot. Increasing  $K_t$  and  $K_r$  gains pulls the vehicle trajectory in tighter and centers it about the roadway, improving tracking. However, large  $K_t$  and  $K_r$  gains led to large overshoot which is not ideal for performance. Increasing both  $K_\beta$  and  $K_{d_{LA}}$ , which has the opposite effect of increasing  $K_t$  and  $K_r$  gains, results in less desirable tracking and reduced overshoot. The longitudinal error gain,  $K_{U_x}$ , drives the longitudinal velocity near the boundary of instability. Therefore, increasing the  $K_{U_x}$  gain leads to both worse tracking and larger overshoot. In contrast, increasing the  $K_{U_x}$  gain reduces chattering. Sometimes there are two sets of controller inputs that result in the same minimum  $\dot{S}$ . Adding a  $K_{U_x}$  gain forces the controller to always select the controller input that results in the maximum  $U_x$ , which in turn reduces chattering. Table 5 summarizes the effect of each controller gain on tracking performance and overshoot. All the gains in the following simulations are balanced so that the controller tracks the roadway with little overshoot.

Table 5. Effect of each controller gain on tracking and overshoot.

Gain	Tracking	Overshoot
Increasing $K_\tau$	Better	Larger
Increasing $K_r$	Better	Larger
Increasing $K_\beta$	Worse	Smaller
Increasing $K_{d_{LA}}$	Worse	Smaller
Increasing $K_{U_x}$	Worse	Larger

The sliding surface controller is assessed under the conditions shown in Table 6 for comparison to the baseline controller. The chosen gains are displayed in Table 7.

Table 6. Sliding Surface Controller Simulation Inputs.

Sliding Surface Controller - Inputs for Simulations						
	<b>R</b>	$U_{x,initial}$	$U_{y,initial}$	$r_{initial}$	Gains	Friction Factor
	<b>m</b>	<b>m/s</b>	<b>m/s</b>	<b>rad/s</b>		
<b>Sliding Surface Controller</b>	110	28	0	0	Table 7	0.4, 0.9

Table 7. Hand-tuned controller gains.

Gain	Value	Gain	Value	Gain	Value
$K_r$	28	$K_\tau$	52	$K_{U_x}$	0.02
$K_\beta$	50	$\kappa_{d_{LA}}$	1560		

It is worth comparing the inputs used for the simulation to the road radius design guidelines for various vehicle speeds. Road radius design guidelines for highways, given various vehicle speeds, are shown in Table 8.

Table 8. Comparison to road radius design guidelines for highways ["Horizontal..."].

Curvature of High Speed Highways & Connecting Roadways (6% Superelevation)				Selected Simulation Values	
Design Speed (m/s)	mph	Usual Min. Radius of Curve (m)	Absolute Min. Radius of Curve (m)	Radius (m)	Speed (m/s)
19.4	43.4	236	184		
22.2	49.7	323	252		
25.0	55.9	521	336		
<b>27.8</b>	<b>62.2</b>	<b>725</b>	<b>437</b>	<b>110</b>	<b>28</b>
30.6	68.5	930	560		
33.3	74.5	1142	756		

1 m/s = 2.237 mph

The scenario selected for the simulation,  $U_{x,initial} = 28$  m/s, is shown in comparison to the design guidelines for highways at a comparative vehicle speed. The road radius for the simulation is far more severe than the recommended guidelines to better correlate to an emergency cornering maneuver.

The vehicle position as it travels through the corner for  $\mu = 0.4$  and  $\mu = 0.9$  is shown in Figure 32. A comparison between the controllers, seen in Figure 31 and Figure 32, shows that the designed sliding surface controller is more effective than the baseline controller. The vehicle trajectory of particular interest is  $\mu = 0.4$ , as the lowest friction

coefficient is most difficult to handle. This trajectory exhibited some oscillation, but successfully remained within the standard lane width and eventually converged to the desired path. For a 110 m cornering radius and  $\mu = 0.4$ , the maximum stable operating longitudinal velocity is 20.8 m/s, well below the initial starting velocity of the vehicle. Therefore, the sliding surface controller is able to stabilize the vehicle with initial states outside the stable operating bounds while maintaining the desired path. The design guidelines in Table 8 recommend a minimum cornering radius of 437 m for a speed of 27.8 m/s. The sliding surface controller is effective in controlling the vehicle at 28 m/s at a much smaller radius, 110 m, performing well within the curvature guidelines for highways and connecting roadways.

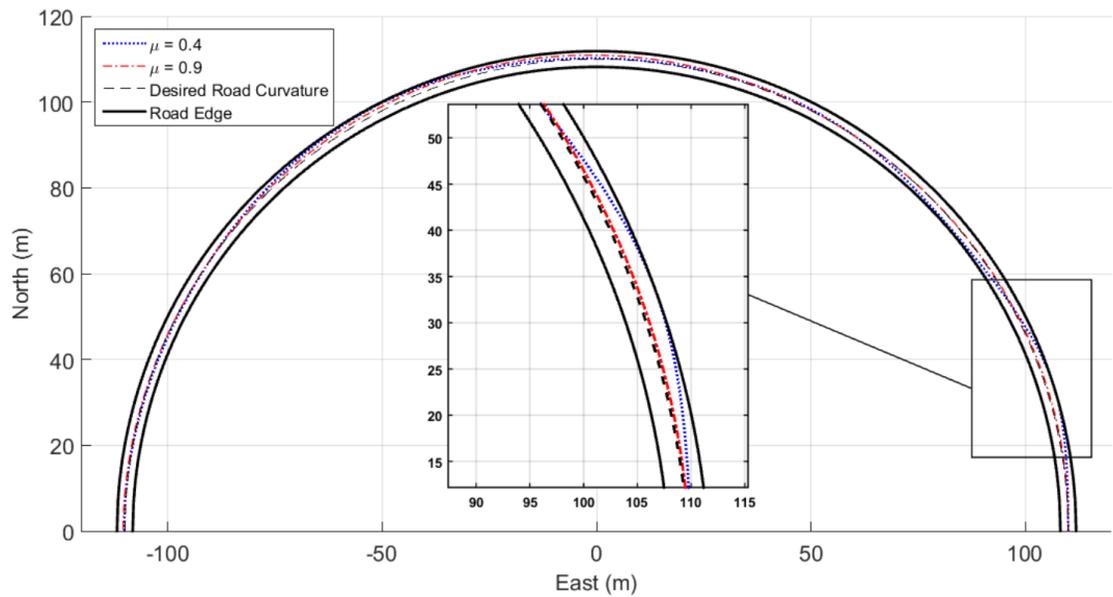


Figure 32. Vehicle response for  $\mu = 0.4$  and  $\mu = 0.9$  with sliding surface controller in place.  $U_{x,initial} = 28$  m/s,  $U_{y,initial} = 0$  m/s,  $r_{initial} = 0$  rad/s, and  $R = 110$  m.

As vehicles are subject to sudden changes in environmental factors and disturbances, the vehicle response across various friction coefficients, vehicle speeds, and cornering radii are simulated to demonstrate robustness. The range of friction coefficients correlates to a variety of road conditions. Appendix A contains a table of friction coefficients for various road surfaces and conditions. The lateral error from the lane centerline is compared while varying each parameter (i.e.  $\mu$ ,  $U_{x,initial}$ ,  $R$ ). The scenarios considered are summarized in Table 9.

Table 9. Sliding Surface Robustness Simulation Inputs.

<b>Sliding Surface Robustness Assessment - Inputs for simulations</b>						
	<b>R</b>	$U_{x,initial}$	$U_{y,initial}$	$r_{initial}$	<b>Gains</b>	<b>Friction Factor</b>
Fig. 33	110	28	0	0	Table 7	0.3, 0.4, 0.5, 0.9
Fig. 34	110	22, 25, 28, 30	0	0	Table 7	0.4
Fig. 35	95, 110, 125	28	0	0	Table 7	0.4

Figures 33-35 show that the controller is able to keep the vehicle within the lane width across a wide variation in each parameter. However, there is a critical point for each parameter at which the controller is no longer able to stabilize the vehicle within the lane width, given the set of gains selected for the simulation. For example, lateral error increases with decreasing friction coefficient, increasing initial longitudinal velocity, and decreasing corner radius. Based on the simulation results, the designed controller is robust and effective in extending the range of conditions in which the vehicle can be stabilized and kept on the roadway.

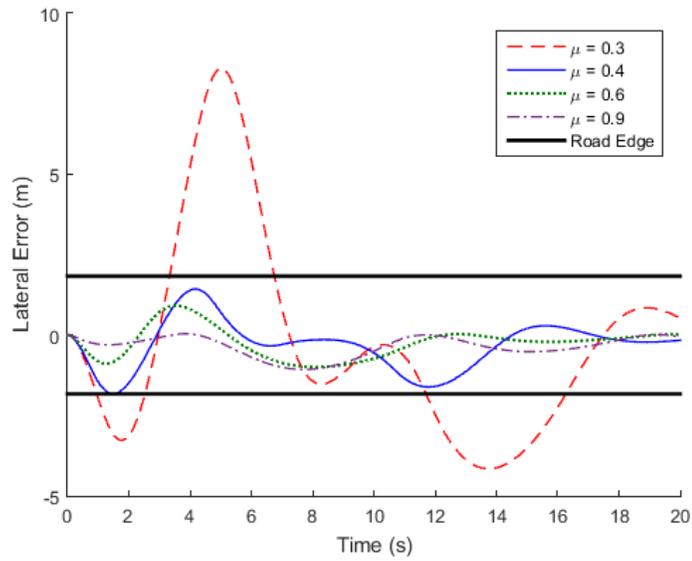


Figure 33. Vehicle lateral error from roadway centerline for various friction coefficients.  $U_{x,initial} = 28$  m/s,  $U_{y,initial} = 0$  m/s,  $r_{initial} = 0$  rad/s, and  $R = 110$  m.

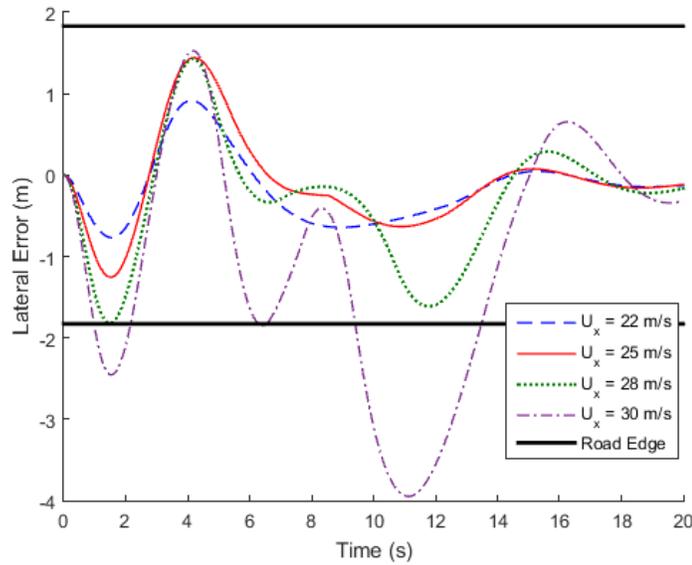


Figure 34. Vehicle lateral error from roadway centerline for various initial longitudinal velocities.  $\mu = 0.4$ ,  $R = 110$  m,  $U_{y,initial} = 0$  m/s, and  $r_{initial} = 0$  rad/s.

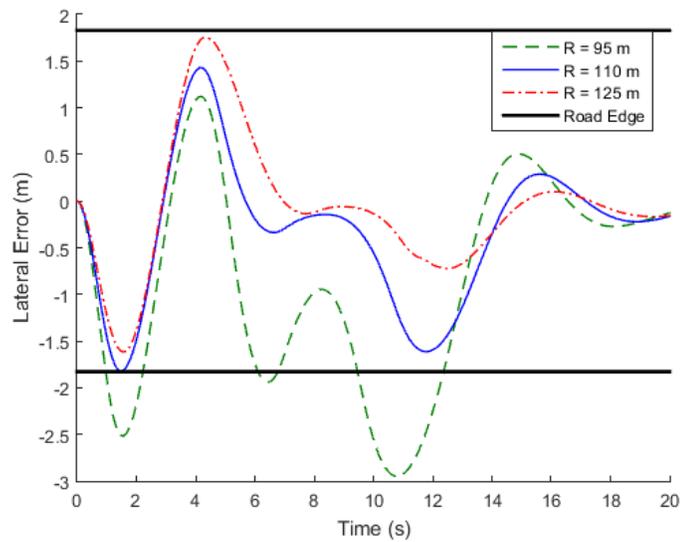


Figure 35. Vehicle lateral error from roadway centerline for various cornering radii.  $\mu = 0.4$ ,  $U_{x,initial} = 28$  m/s,  $U_{y,initial} = 0$  m/s, and  $r_{initial} = 0$  rad/s.

The sequence of steering and braking inputs applied to maintain control during cornering is also examined. The braking and steering inputs corresponding to  $\mu = 0.4$ ,  $R = 110$  m, and  $U_{x,initial} = 28$  m/s are shown in Figure 36. This figure reveals that a complex sequence of braking and steering inputs is required to maintain stability during critical cornering scenarios. The first phase, between 0 and approximately 0.5 s, consists primarily of hard braking with steering corrections, presumably to maintain yaw tracking. Near 1s, the braking and steering inputs are altered in coordination, though, unlike what occurs in trail braking, countersteering is applied. Finally, the brakes are released and the controller provides steering around the corner.

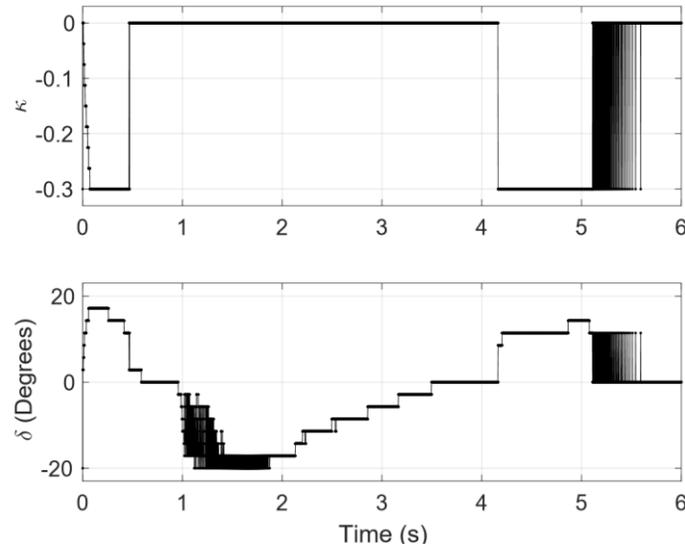


Figure 36. Sequence of steering and braking inputs for a vehicle navigating a 110 m corner with hierarchical sliding surface control.

Figure 36 shows one example of a braking and steering sequence. A given braking and steering sequence is dependent on the vehicle's initial states. The sequence presented in Figure 36 is specific to a vehicle with  $\mu = 0.4$ ,  $R = 110$  m, and  $U_{x,initial} = 28$  m/s. Different initial conditions yield a different braking and steering sequence. Additionally, the steering and braking inputs must be performed over a very short period of time. Human reaction time is very slow and the sequence of inputs is not intuitive, making replication of these inputs very challenging to impossible for a human driver.

In the previous discussion of various control approaches, chattering was presented as a possible disadvantage of sliding surface control. Small chattering is expected as the vehicle approaches the sliding surface and is acceptable so long as it does not occur for a long period of time. The sequence of braking and steering inputs in Figure 36 show that the designed controller does involve some chatter. However, the chatter occurs between

1-2 s and 5-5.5 s, both of which are short time spans and correspond to regions where  $S$  is close to 0. Furthermore, the goal of this controller is to get the vehicle onto the sliding surface at which point it is assumed the driver will be able to keep the vehicle within safe operating bounds without the help of the controller. This vehicle has achieved safe cornering conditions at 5 s so it is expected that the driver is able to take over at this point, eliminating all chatter after 5 s.

### 7.3 Minimum Brake Distance

The last possible braking point where a controller can safely negotiate the corner is also considered. To perform this analysis, a constant maximum deceleration is assumed to estimate the minimum brake distance. Using constant acceleration equations, the minimum braking distance is described by equation 40.

$$\Delta x_{min} = \frac{U_{x,max}^2 - U_{x,initial,critical}^2}{2a} \quad (\text{Eq. 40})$$

where  $U_{x,initial,critical}$  is the critical initial longitudinal velocity above which the controller can no maintain the vehicle on the vehicle back onto the roadway with the selected gains.  $U_{x,max}$  is the maximum stable steady state cornering velocity, given in equation 23, and  $a$  is the constant deceleration. The maximum deceleration, derived in equations 41-44, is based on the maximum frictional force and is estimated as  $\mu g$ .

$$\sum F = ma \quad (\text{Eq. 41})$$

$$F_{Friction} = \mu N = ma \quad (\text{Eq. 42})$$

$$\mu mg = ma \quad (\text{Eq. 43})$$

$$a = \mu g \quad (\text{Eq. 44})$$

The critical speed,  $U_{x,initial,critical}$ , is estimated using the gains presented in Table 7 for various friction coefficients and cornering radii. The minimum braking distance for several example scenarios are presented in Table 10. As a road becomes slicker or the cornering radius is decreased, the minimum brake distance required increases. The minimum braking distance is each of these scenarios is not small; therefore, early identification of the last possible safe braking point is crucial.

Table 10. Minimum braking distance for various  $\mu$  and R.

$\mu$	R (m)	$U_{x,initial,critical}$ (m/s)	$U_{x,max}$ (m/s)	$\Delta x_{min}$ (m)
0.4	95	26.0	19.3	38.7
0.4	110	28.0	20.8	44.8
0.4	125	29.7	22.1	50.2
0.3	110	24.3	18.0	45.3
0.6	110	33.1	24.4	42.5

The minimum brake distance could be used in conjunction with a vehicle-to-infrastructure communication system capable of identifying and providing warning of dangerous upcoming cornering conditions. For example, an on-board controller could predict the last possible point at which a controller would need to assist the driver to maintain safe lane tracking. In the future, this information could also enable safer higher speed autonomous vehicle operation.

## 8 Conclusion

Increased use of vehicle control systems has led to a significant decrease in accidents. While most prior work in critical cornering has focused on use of a single actuator, the work presented in this thesis suggests that a controller based on combined steering and braking is more effective. A commonly used 2D phase portrait tool, enabling the effects of steering to be visualized, is expanded to a 3D phase portrait tool, enabling the impact of both braking, as well as steering, to be examined. Data obtained during experimental cornering scenarios indicate this 3D phase portrait tool is effective in simulating the vehicle dynamics. The tool enables visualization of stable operating regions and informs the selection of a sliding surface control approach most appropriate for critical cornering scenarios.

The hierarchical sliding surface controller presented demonstrates that control inputs can be optimized to significantly extend the range of corner entry speeds for which a vehicle can stably track the road. This controller is intended to offer non-intrusive assistance to drivers as vehicles approach unfavorable operating conditions and utilize maximum handling capabilities to safely maneuver the vehicle to remain on the road. The optimization built into the sliding surface control provides complex control action coordinating the vehicle actuators in a manner above the skill level of an average driver. The controller is able to stabilize vehicle states outside of the defined open-loop stable bounds and maintain the vehicle on the roadway within the lane width, for a wide variety of reasonable operating conditions, demonstrating robustness. The driver assistance

offered by the controller extends the safe corner entry speed envelope and improves vehicle safety in critical cornering scenarios. Further, the ability to determine the last possible braking point for a given cornering scenario enables timely intervention of a controller to avert an unsafe cornering event if coupled with a warning system capable of identifying and providing notification of upcoming dangerous cornering conditions.

## 9 Bibliography

- "A QUICK INTRODUCTION TO SLIDING MODE CONTROL AND ITS APPLICATIONS." Department of Electrical and Electronic Engineering - University of Cagliari, Web. 1 June 2016.
- Adireddy, G., Shim, T., Rhode, D., and Asgari, J., 2014. "Combined wheel torque and steering control based on model predictive controller using a simplified trye model". *Int. J. Vehicle Design*, 65, pp. 31–51.
- "An Introduction To Control Systems." <http://www.facstaff.bucknell.edu/>. Web. 02 June 2016.
- Arkun, Y., and E. Zafiriou. "Model Predictive Control: Theory and Practice." *Model Based Process Control: Proceedings of the IFAC Workshop, Atlanta, Georgia, USA, 13-14 June, 1988*. By T. J. McAvoy. Vol. 82.: Elsevier, 2014. Print.
- Barzamini, R., and Shafiee, M., 2011. "A new sliding mode controller for tcp congestion control". In *Telecommunications Forum (TELFOR), 2011 19th, IEEE*, pp. 214–217.
- Beal, Craig Earl. *Applications of Model Predictive Control to Vehicle Dynamics for Active Safety and Stability*. Diss. Stanford U, 2011. Print.
- Fricke, Lynn B. *Traffic Accident Reconstruction*. Evanston, IL: Northwestern U Traffic Institute, 1990. Print.

- Glover, K., Sefton, J., & McFarlane, D. C. (1992). A tutorial on loop shaping using H-infinity robust stabilisation. *Transactions of the Institute of Measurement and Control*, 14(3), 157-168.
- Gorinevsky, Dimitry. "Lecture 14 - Model Predictive Control, Part 1." EE392m: Control Engineering in Industry. Stanford University. Web. 1 June 2016.
- Helton, J. W., & James, M. R. (1999). Extending  $H_\infty$  control to nonlinear systems: control of nonlinear systems to achieve performance objectives. Society for Industrial and Applied Mathematics.
- Honeywell, Desborough. "PID Control." *Process Automation Handbook* (2000): 155-63. [www.cds.caltech.edu](http://www.cds.caltech.edu). Caltech Division of Engineering & Applied Science. Web. 1 June 2016.
- "Horizontal Alignment." *Roadway Design Manual*. Texas Department of Transportation, Oct. 2014. Web. 01 May 2017.
- Inagaki, S., Kshiro, I., and Yamamoto, M., 1994. "Analysis on vehicle stability in critical cornering using phase-plane method". In *Proceedings of the International Symposium on Advanced Vehicle Control*, Arnhem.
- Jaganmohan Reddy, Y. "Chapter 19: Advanced Process Control Systems." *My Library My History Books on Google Play Industrial Process Automation Systems: Design and Implementation*. By B. R. Mehta.: Butterworth-Heinemann, 2014. Print.
- Janardhanan, S. "Sliding Mode Control – An Introduction." *Documents.mx*. Indian Institute of Technology Delhi. Web. 1 June 2016.

- Kritayakirana, K., and Gerdes, J., 2012. "Autonomous vehicle control at the limits of handling". *Int. J. Vehicle Autonomous Systems*, 10, pp. 271–296.
- Kwak, B., and Park, Y., 2001. "Robust vehicle stability controller based on multiple sliding mode control". In *SAE Technical Paper*, SAE International.
- Liu, Cejun, and Tony Jianqiang Ye. "Run-Off-Road Crashes: An On-Scene Perspective." U.S. Department of Transportation, National Highway Safety Traffic Administration, July 2011. Web. 24 Apr. 2017.
- Mitchell, W., Schroer, R., and Grisez, D., 2004. "Driving the traction circle". In *Proceedings of the 2004 SAE Motorsports Engineering Conference and Exhibition*.
- Nozaki, H., 2005. "Compromise between sensing of critical cornering and drift control". In *SAE Technical Paper*, SAE International.
- Nozaki, H., and Ueda, Y., 2007. "Investigation of a new steering method to improve control during cornering". In *SAE Technical Paper*, SAE International.
- Skogestad, S., & Postlethwaite, I. (2007). *Multivariable feedback control: analysis and design* (Vol. 2). New York: Wiley.
- Toroyan, T., 2015. *Global status report on road safety*.  
[http://www.who.int/violence\\_injury\\_prevention/road\\_safety\\_status/2015/en/](http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/).
- Velenis, E., Tsiotras, P., and Lu, J., 2008. "Optimality properties and driver input parameterization for trail-braking cornering". *European Jnl. Of Control*, 14(4), pp. 308–320.

Yoshino, T., and Nozaki, H., 2014. "Effect of Direct Yaw Moment Control Based on Steering Angle Velocity and Camber Angle Control". In SAE Technical Paper, SAE International.

## Appendices

### Appendix A

Table 11. Friction coefficients for various surfaces and road conditions [Fricke 1990].

<b>COEFFICIENTS OF FRICTION OF VARIOUS ROADWAY SURFACES</b>								
<b>DESCRIPTION OF ROAD SURFACES</b>	<b>DRY</b>				<b>WET</b>			
	<b>Less than 30 mph</b>		<b>More than 30 mph</b>		<b>Less than 30 mph</b>		<b>More than 30 mph</b>	
	<b>From</b>	<b>To</b>	<b>From</b>	<b>To</b>	<b>From</b>	<b>To</b>	<b>From</b>	<b>To</b>
<b>PORTLAND CEMENT</b>								
New, Sharp	0.83	1.20	0.70	1.00	0.50	0.80	0.40	0.75
Traveled	0.60	0.80	0.60	0.75	0.45	0.70	0.45	0.65
Traffic Polished	0.55	0.75	0.50	0.65	0.45	0.65	0.45	0.60
<b>ASPHALT or TAR</b>								
New, Sharp	0.80	1.20	0.65	1.00	0.50	0.80	0.45	0.75
Traveled	0.60	0.80	0.55	0.70	0.45	0.70	0.40	0.65
Traffic Polished	0.55	0.75	0.45	0.65	0.45	0.65	0.40	0.60
Excess Tar	0.50	0.60	0.35	0.60	0.30	0.60	0.25	0.55
<b>GRAVEL</b>								
Packed, Oiled	0.55	0.85	0.50	0.80	0.40	0.80	0.40	0.60
Loose	0.40	0.70	0.40	0.70	0.45	0.75	0.45	0.75
<b>CINDERS</b>								
Packed	0.50	0.70	0.50	0.70	0.65	0.75	0.65	0.75
<b>ROCK</b>								
Crushed	0.55	0.75	0.55	0.75	0.55	0.75	0.55	0.75
<b>ICE</b>								
Smooth	0.10	0.25	0.07	0.20	0.05	0.10	0.05	0.10
<b>SNOW</b>								
Packed	0.30	0.55	0.35	0.55	0.30	0.60	0.30	0.60
Loose	0.10	0.25	0.10	0.20	0.30	0.60	0.30	0.60

## Appendix B

### Sliding Surface Controller Code

```

% Sliding Surface Controller Code
% By Christina Boyd

% This code computes the optimal braking and steering control inputs to drive the vehicle dynamics to the sliding
surface.
% This code uses the 4 wheel braking model.

clc
clear all
close all

% Single Initial Condition (IC)
Uy = 0;
r = 0;
Ux = 34.1;

% Grid of ICs Points
UyBound = [-30 30];
rBound = [-2.5 0.72];
UxBound = [2 40];

[mu, mu_s, mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr, ~] = VehicleConstants4Wheel_7_19_16(1);
Trailbraking_Radius_Des = 110;

mu = [0.6];
mu_s = mu;

% Lower and upper bounds of optimization inputs
BrakingType = '4WheelBraking';

switch BrakingType
case 'DiffBraking'
    lb = [-0.3 -20*pi/180]; % Kappa lb, Delta lb
    ub = [0 20*pi/180]; % Kappa ub, Delta ub

case '4WheelBraking'
    lb = [-0.3 -0.3 -0.3 -0.3 -20*pi/180]; % Kappa lb (4), Delta lb
    ub = [0 0 0 0 20*pi/180]; % Kappa ub (4), Delta ub
otherwise
end

% Evaluate at 4 grid corners/endpoints (Calculate S dot)
kappa_TestPoints = linspace(lb(1),ub(1),9);
delta_TestPoints = linspace(lb(end),ub(end),15);

% Choose r and/or Beta control and with/without tracking
Control = 'All'; % Choose 'r' or 'r and B', or 'All' - All means r, B, and Ux cost terms included
Tracking = 'Yes'; % Choose 'No' or 'Yes'

```

```

%% Euler's Streamline
dt = 0.0005; % seconds
oo = 1;
t = (0:dt:20)'; % s

for ggg = 1:length(Trailbraking_Radius_Des)
    for fff = 1:length(mu)

        Ux_max = sqrt(mu(fff)*g*Trailbraking_Radius_Des(ggg));

        livePlots = 0;
        waitbar((fff-1)/length(mu),h)
        for i = 1:length(Uy)
            for j = 1:length(r)
                for k = 1:length(Ux)

                    clearvars -except a b BCostFactor br BrakingType Caf Car Center CenterEastLocation Control ...
                        Cxf Cxr d del delta_TestPoints derivBEndPoints derivrEndPoints derivUxEndPoints
                        derivUyEndPoints...
                        dt d_LA EulerStreamDynamics EulerStreamInputs EulerStreamWorld fff fileName Fzf Fzr...
                        g ggg GainTB h hIplots i index indexLocation InnerEastLocation Izz j jj k kappa_TestPoints...
                        L LaneWidth LateralERROR lb livePlots mass...
                        minmaxLocI minmaxSdotVal mu mu_s N nowStr numPoints nPts oo OuterEastLocation
                        PositionRelativeToRoad
                        pp PropGainSteer r...
                        rBound rCostFactor RoadCenter RoadEdge_Inner RoadEdge_Outer t...
                        tEulerStream Tracking Trailbraking_Radius_Des ub Ux UxBound Ux_max Uy UyBound x_initial...
                        x_optimized y_optimized y_optimized_Max Gains

                    % Preallocate Vectors
                    delta_Global_Max(1:length(t)) = NaN;
                    East_dot(1:length(t)) = NaN;
                    Heading_dot(1:length(t)) = NaN;
                    North_dot(1:length(t)) = NaN;
                    S_dot(1:length(t)) = NaN;
                    e_dot(1:length(t)) = NaN;
                    kappalf_Global_Max(1:length(t)) = NaN;
                    kappalr_Global_Max(1:length(t)) = NaN;
                    kapparf_Global_Max(1:length(t)) = NaN;
                    kapparr_Global_Max(1:length(t)) = NaN;
                    rCost(1:length(t)) = NaN;
                    Road_Radius_Actual(1:length(t)) = NaN;
                    Road_Radius_Actual2(1:length(t)) = NaN;
                    S(1:length(t)) = NaN;
                    Trailbraking_Radius(1:length(t)+1) = NaN;
                    xdot(1:length(t)) = NaN;
                    ydot(1:length(t)) = NaN;
                    zdot(1:length(t)) = NaN;
                    y_optimized_Global_Max(1:length(t)) = NaN;
                    Lateral_error(1:length(t)) = NaN;
                    Curvature_Error(1:length(t)) = NaN;
                    RadiusOfCurvature(1:length(t)) = NaN;

                    EWorld(1:length(t)+1) = NaN;
                    HWorld(1:length(t)+1) = NaN;
                    NWorld(1:length(t)+1) = NaN;
                    SWorld(1:length(t)+1) = NaN;
                    eWorld(1:length(t)+1) = NaN;
                    ArcAngle(1:length(t)+1) = NaN;
                end
            end
        end
    end
end

```

```

xWorldPositionOnCurve(1:length(t)+1) = NaN;
tangent_slope(1:length(t)+1) = NaN;
RoadHeading(1:length(t)+1) = NaN;
DeltaHWorld(1:length(t)+1) = NaN;
x(1:length(t)+1) = NaN;
y(1:length(t)+1) = NaN;
z(1:length(t)+1) = NaN;

% Starting Position
x(1) = Uy(i);    % Uy
y(1) = r(j);    % r
z(1) = Ux(k);    % Ux

% World Frame
NWorld(1) = 0;
EWorld(1) = Trailbraking_Radius_Des(ggg);    % m
HWorld(1) = 0;

SWorld(1) = 0;
eWorld(1) = 0;
DeltaHWorld(1) = 0;

RadiusOfCurvature(1) = Trailbraking_Radius_Des(ggg);
Trailbraking_Radius(1) = Trailbraking_Radius_Des(ggg);

% A single starting point at the center of the kappa and delta limits

switch BrakingType
case 'DiffBraking'
    x_initial = [(lb(1)+ub(1))/2,(lb(end)+ub(end))/2];    % [kappa, delta]

case '4WheelBraking'
    x_initial = [(lb(1)+ub(1))/2,(lb(end)+ub(end))/2];    % [kappa, delta]
otherwise
    error('BrakingType not defined');
end
pp = 1;
while pp <= length(t) && (x(pp) >= min(UyBound) && x(pp) <= max(UyBound) && y(pp) >=
    min(rBound) && y(pp) <= max(rBound) && z(pp) >= min(UxBound) && z(pp) <=
    max(UxBound))

clear y_optimized_Max

switch Tracking
case 'Yes'
    if Trailbraking_Radius_Des(ggg) > 0
        Center = [0 0];
        Road_Radius_Actual(pp) = sqrt( (NWorld(pp)-Center(2))^2 + (EWorld(pp)-Center(1))^2);
    else
        Center = [Trailbraking_Radius_Des(ggg)*2 0];
        Road_Radius_Actual(pp) = -sqrt( (NWorld(pp)-Center(2))^2 + (EWorld(pp)-Center(1))^2);
    end

case 'No'
    Road_Radius_Actual(pp) = Trailbraking_Radius_Des(ggg);
otherwise
    error('Tracking not defined as Yes or No');
end

```

```

% Incorporate look-ahead into the tracking feedback
PropGainSteer = 0.1;

Gains = [52,1560*z(pp)*dt,28,50,0.02];

GainTB = Gains(1);
d_LA = Gains(2);      % m
rCostFactor = Gains(3);
BCostFactor = Gains(4);
UxCostFactor = Gains(5);
Lateral_error(pp) = (Trailbraking_Radius_Des(ggg) - Road_Radius_Actual(pp));

Road_Radius_Actual2(pp) = (Trailbraking_Radius_Des(ggg) - (eWorld(pp) +
    d_LA*DeltaHWorld(pp)));

% Look-ahead: Propagate motion due to d_LA (curved d_LA path)
theta_Heading_dLA(pp) = HWorld(pp) + pi;

% Compute Radius of Curvature based on circle sitting tool
nPts = 10;

if pp >= nPts

    xSampleCircPts = EWorld(pp-nPts+1:pp);
    ySampleCircPts = NWorld(pp-nPts+1:pp);
    [x_center_inputRadius(pp),y_center_inputRadius(pp),RadiusOfCurvature(pp),~] =
        circfit2(xSampleCircPts,ySampleCircPts);

else
    RadiusOfCurvature(pp) = sqrt(x(pp).^2 + z(pp).^2) ./ y(pp);
    x_center_inputRadius(pp) = EWorld(pp) + RadiusOfCurvature(pp)*cos(theta_Heading_dLA(pp));
    y_center_inputRadius(pp) = NWorld(pp) + RadiusOfCurvature(pp)*sin(theta_Heading_dLA(pp));
end

theta_current(pp) = atan2(( NWorld(pp) - y_center_inputRadius(pp)), ( EWorld(pp) -
    x_center_inputRadius(pp)));

theta_dLA(pp) = d_LA/RadiusOfCurvature(pp);
theta_total_dLA(pp) = theta_current(pp) + theta_dLA(pp);

x_dLA(pp) = x_center_inputRadius(pp) + abs(RadiusOfCurvature(pp))*cos(theta_total_dLA(pp));
y_dLA(pp) = y_center_inputRadius(pp) + abs(RadiusOfCurvature(pp))*sin(theta_total_dLA(pp));

error_dLA_Curved(pp) = -Trailbraking_Radius_Des(ggg) + sqrt(x_dLA(pp)^2 + y_dLA(pp)^2);
Curvature_Error(pp) = (1/Trailbraking_Radius_Des(ggg) - 1/(Trailbraking_Radius_Des(ggg) +
    error_dLA_Curved(pp)));

Trailbraking_Radius(pp+1) = 1/(1/Trailbraking_Radius_Des(ggg) + GainTB*(Curvature_Error(pp)));

% Sliding Surface Cost Factors
rCost(pp) = y(pp) - z(pp)/Trailbraking_Radius(pp+1);
UxCost(pp) = (z(pp) - Ux_max);

% Increasing rCostFactor leads to large oscillation, but improves tracking
% Increasing BCostFactor leads to worse tracking performance, but reduces oscillation

switch Control
case 'r'
    S(pp) = rCostFactor*rCost(pp);

```



```

        case 'All'
            y_optimized_Max(br,del) = (2*rCostFactor*rCost(pp)*(derivrEndPoints -
derivUxEndPoints / Trailbraking_Radius(pp+1)) + 2*BCostFactor*(x(pp)/z(pp))*(derivBEndPoints) +
2*UxCostFactor*UxCost(pp)*derivUxEndPoints); % Equivalent to S dot (sliding surface)
            otherwise
                error('Control not defined as r, r and B, or All');
            end
        end
    end
end

% Find location of the best control inputs
if S(pp) > 0
    % Find global minimum
    [minmaxSdotVal, minmaxLoci, minmaxLocj] = minmax_matrix(y_optimized_Max,'min');
else
    % Find global maximum
    [minmaxSdotVal, minmaxLoci, minmaxLocj] = minmax_matrix(y_optimized_Max,'max');
end

% Optimize at best grid point
x_initial = [kappa_TestPoints(minmaxLoci) delta_TestPoints(minmaxLocj)];

% Best control inputs
x_optimized = x_initial;
y_optimized = minmaxSdotVal;

% Pull the inputs from the grid search result
kappalf_Global_Max(pp) = x_optimized(1);
kapparf_Global_Max(pp) = x_optimized(1);
kappalr_Global_Max(pp) = x_optimized(1);
kapparr_Global_Max(pp) = x_optimized(1);

delta_Global_Max(pp) = x_optimized(2);
y_optimized_Global_Max(pp) = y_optimized;

case 'DiffBraking'
    for ll = 1:length(kappa_TestPoints)
        for mm = 1:length(delta_TestPoints)
            if rCost(pp) >= 0 % actuate right front brake
                [derivUyEndPoints, derivrEndPoints, derivUxEndPoints, derivBEndPoints] =
FourWheelNonlinDynWithGrad(x(pp), y(pp), z(pp), delta_TestPoints(mm), [0, kappa_TestPoints(ll), 0, 0], mu(ff),
mu_s(ff), mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr, Trailbraking_Radius(pp+1));
                switch Control
                    case 'r'
                        y_optimized_Max(ll,mm) = rCostFactor*(derivrEndPoints - derivUxEndPoints /
Trailbraking_Radius(pp+1)); % Equivalent to S dot
                    case 'r and B'
                        y_optimized_Max(ll,mm) = (2*rCostFactor*rCost(pp)*(derivrEndPoints -
derivUxEndPoints / Trailbraking_Radius(pp+1)) + 2*BCostFactor*(x(pp)/z(pp))*(derivBEndPoints)); % Equivalent to
S dot
                    case 'All'
                        y_optimized_Max(ll,mm) = (2*rCostFactor*rCost(pp)*(derivrEndPoints -
derivUxEndPoints / Trailbraking_Radius(pp+1)) + 2*BCostFactor*(x(pp)/z(pp))*(derivBEndPoints) +
2*UxCostFactor*UxCost(pp)*derivUxEndPoints); % Equivalent to S dot
                    otherwise
                        error('Control not defined as r, r and B, or All');
                end
            end
        end
    end
end

```

```

else % actuate left rear brake
    [derivUyEndPoints, derivrEndPoints, derivUxEndPoints, derivBEndPoints] =
FourWheelNonlinDynWithGrad(x(pp), y(pp), z(pp), delta_TestPoints(mm), [0, 0, kappa_TestPoints(l1), 0], mu(ff),
mu_s(ff), mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr, Trailbraking_Radius(pp+1));
    switch Control
    case 'r'
        y_optimized_Max(l1,mm) = -rCostFactor*(derivrEndPoints - derivUxEndPoints /
Trailbraking_Radius(pp+1)); % Equivalent to S dot
    case 'r and B'
        y_optimized_Max(l1,mm) = (2*rCostFactor*rCost(pp)*(derivrEndPoints -
derivUxEndPoints / Trailbraking_Radius(pp+1)) + 2*BCostFactor*(x(pp)/z(pp))*(derivBEndPoints)); % Equivalent to
S dot
    case 'All'
        y_optimized_Max(l1,mm) = (2*rCostFactor*rCost(pp)*(derivrEndPoints -
derivUxEndPoints / Trailbraking_Radius(pp+1)) + 2*BCostFactor*(x(pp)/z(pp))*(derivBEndPoints) +
2*UxCostFactor*UxCost(pp)*derivUxEndPoints); % Equivalent to S dot
    otherwise
        error('Control not defined as r, r and B, or All');
    end
end
end
end

% Find location of the best control inputs
if S(pp) > 0
    % Find global minimum
    [minmaxSdotVal, minmaxLocI, minmaxLocJ] = minmax_matrix(y_optimized_Max,'min');
else
    % Find global minimum
    [minmaxSdotVal, minmaxLocI, minmaxLocJ] = minmax_matrix(y_optimized_Max,'max');
end

% Optimize at best grid point
x_initial = [kappa_TestPoints(minmaxLocI) delta_TestPoints(minmaxLocJ)];

% Best control inputs
x_optimized = x_initial;
y_optimized = minmaxSdotVal;

% Pull the inputs from the grid search result
if rCost(pp)>=0 % actuate front brake
    kappalf_Global_Max(pp) = 0;
    kapparf_Global_Max(pp) = x_optimized(1);
    kappalr_Global_Max(pp) = 0;
    kapparr_Global_Max(pp) = 0;

    delta_Global_Max(pp) = x_optimized(2);
    y_optimized_Global_Max(pp) = y_optimized;
else % actuate rear brake
    kappalf_Global_Max(pp) = 0;
    kapparf_Global_Max(pp) = 0;
    kappalr_Global_Max(pp) = x_optimized(1);
    kapparr_Global_Max(pp) = 0;

    delta_Global_Max(pp) = x_optimized(2);
    y_optimized_Global_Max(pp) = y_optimized;
end
otherwise

```

```

        error('BrakingType not defined');

    end

    [xdot(pp), ydot(pp), zdot(pp)] = FourWheelNonlinDynWithGrad(x(pp), y(pp), z(pp),
    delta_Global_Max(pp), [kappalf_Global_Max(pp), kapparf_Global_Max(pp), kappalr_Global_Max(pp),
    kapparr_Global_Max(pp)], mu(fff), mu_s(fff), mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr,
    Trailbraking_Radius(pp+1));

    % Next Time Step
    x(pp+1) = x(pp) + xdot(pp)*dt;    % Uy
    y(pp+1) = y(pp) + ydot(pp)*dt;    % r
    z(pp+1) = z(pp) + zdot(pp)*dt;    % Ux

    % Compute world Derivatives
    Heading_dot(pp) = y(pp);
    North_dot(pp) = z(pp)*cos(HWorld(pp)) + x(pp)*sin(HWorld(pp));
    East_dot(pp) = -x(pp)*cos(HWorld(pp)) - z(pp)*sin(HWorld(pp));

    S_dot(pp) = z(pp)*cos((pp)) - x(pp)*sin(DeltaHWorld(pp));
    e_dot(pp) = x(pp)*cos(DeltaHWorld(pp)) + z(pp)*sin(DeltaHWorld(pp));

    % Plot where vehicle is in the world frame
    NWorld(pp+1) = NWorld(pp) + North_dot(pp)*dt;
    EWorld(pp+1) = EWorld(pp) + East_dot(pp)*dt;
    HWorld(pp+1) = HWorld(pp) + Heading_dot(pp)*dt;

    SWorld(pp+1) = SWorld(pp) + S_dot(pp)*dt;
    eWorld(pp+1) = eWorld(pp) + e_dot(pp)*dt;

    % Use current world position to find the road heading
    ArcAngle(pp+1) = atan2(NWorld(pp+1),EWorld(pp+1));
    xWorldPositionOnCurve(pp+1) = Trailbraking_Radius_Des(ggg)*cos(ArcAngle(pp+1));
    tangent_slope(pp+1) = 1/2*((Trailbraking_Radius_Des(ggg)^2 - xWorldPositionOnCurve(pp+1)^2)^(-
    1/2))*(-2*xWorldPositionOnCurve(pp+1));

    if EWorld(pp+1)>=0
        RoadHeading(pp+1) = atan2(-tangent_slope(pp+1),-1) - (pi/2);
    else
        RoadHeading(pp+1) = atan2(-tangent_slope(pp+1),-1) + 2*pi - (pi/2);
    end

    DeltaHWorld(pp+1) = HWorld(pp+1) - RoadHeading(pp+1);

    pp = pp+1;
end

% Save states from current vehicle trajectory
tEulerStream{1,oo} = t;
EulerStreamDynamics{1,oo} = [x(1:end-1)', y(1:end-1)', z(1:end-1)'];
EulerStreamInputs{1,oo} = [kappalf_Global_Max' kapparf_Global_Max' kappalr_Global_Max'
kapparr_Global_Max' delta_Global_Max' Trailbraking_Radius(1:(end-1))'];

```

```

        EulerStreamWorld{1,oo} = [EWorld(1:end-1)', NWorld(1:end-1)', RadiusOfCurvature(1:end)',
Road_Radius_Actual'];
        PositionRelativeToRoad{1,oo} = [SWorld(1:end-1)', eWorld(1:end-1)', DeltaHWorld(1:end-1)'];
        LateralERROR{1,oo} = Lateral_error;

        oo = oo + 1;

        if livePlots
            figure(1)
            plot3(x', y', z')
        end

    end
end
end
end
end
end

```

## Four Wheel Vehicle Dynamics - Equations of Motion

% By Christina Boyd

```

function [derivUy, derivvr, derivUx, derivB] = FourWheelNonlinDynWithGrad(Uy, r, Ux, delta, kappa, mu, mu_s,
mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr, Trailbraking_Radius)

```

```

[n,m,l] = size(Uy); %Note: size(Uy) = size(r) = size(Ux) due to meshgrid

```

```

Klf = kappa(1); % left, front
Krf = kappa(2);
Klr = kappa(3);
Krr = kappa(4);

```

```

%alpha matrix, front and rears
alphalf = atan((Uy+a*r)/(Ux - d*r))-delta;
alpharf = atan((Uy+a*r)/(Ux + d*r))-delta;
alphalr = atan((Uy-b*r)/(Ux - d*r));
alpharr = atan((Uy-b*r)/(Ux + d*r));

```

```

%Left Front Forces
f_lf = sqrt((Cxf*(Klf/(1+Klf)).^2+(Caf*(tan(alphalf)/(1+Klf)).^2);

```

```

if f_lf == 0
    Fxlf = 0;
    Fyly = 0;
else
    F_lf = (f_lf <= 3*mu*Fzf).*(f_lf - (1/(3*mu*Fzf))*(2-mu_s/mu).*f_lf.^2+(1/(9*mu^2*Fzf^2))*(1-
(2/3)*(mu_s/mu)).*f_lf.^3);
    F_lf = F_lf+(~(f_lf <= 3*mu*Fzf))*mu_s*Fzf;

    Fxlf = Cxf*(Klf/(1+Klf)).*(F_lf/f_lf);
    Fyly = -Caf*(tan(alphalf)/(1+Klf)).*(F_lf/f_lf);
end

```

```

FxlV = Fxlf.*cos(delta) - Fylf.*sin(delta);
FylV = Fxlf.*sin(delta) + Fylf.*cos(delta);

%Right Front Forces
f_rf = sqrt((Cxf*(Krf./(1+Krf))).^2+(Caf*(tan(alpharf)./(1+Krf))).^2);

if f_rf == 0
    Fxrf = 0;
    Fyrf = 0;
else
    F_rf = (f_rf <= 3*mu*Fzf).*(f_rf - (1/(3*mu*Fzf))*(2-mu_s/mu).*f_rf.^2+(1/(9*mu^2*Fzf^2))*(1-
(2/3)*(mu_s/mu)).*f_rf.^3);
    F_rf = F_rf+(~(f_rf <= 3*mu*Fzf))*mu_s*Fzf;

    Fxrf = Cxf*(Krf./(1+Krf)).*(F_rf./f_rf);
    Fyrf = -Caf*(tan(alpharf)./(1+Krf)).*(F_rf./f_rf);
end

FxfV = Fxrf.*cos(delta) - Fyrf.*sin(delta);
FyfV = Fxrf.*sin(delta) + Fyrf.*cos(delta);

% Left Rear Forces
f_lr = sqrt((Cxr*(Klr./(1+Klr))).^2+(Car*(tan(alphalr)./(1+Klr))).^2);

if f_lr == 0
    Fxlr = 0;
    Fylr = 0;
else
    F_lr = (f_lr <= 3*mu*Fzr).*(f_lr - (1/(3*mu*Fzr))*(2-mu_s/mu).*f_lr.^2+(1/(9*mu^2*Fzr^2))*(1-
(2/3)*(mu_s/mu)).*f_lr.^3);
    F_lr = F_lr+(~(f_lr <= 3*mu*Fzr))*mu_s*Fzr;

    Fxlr = Cxr*(Klr./(1+Klr)).*(F_lr./f_lr);
    Fylr = -Car*(tan(alphalr)./(1+Klr)).*(F_lr./f_lr);
end

FxlV = Fxlr;
FylV = Fylr;

% Right Rear Forces
f_rr = sqrt((Cxr*(Krr./(1+Krr))).^2+(Car*(tan(alpharr)./(1+Krr))).^2);

if f_rr ==0
    Fxrr = 0;
    Fyrr = 0;
else
    F_rr = (f_rr <= 3*mu*Fzr).*(f_rr - (1/(3*mu*Fzr))*(2-mu_s/mu).*f_rr.^2+(1/(9*mu^2*Fzr^2))*(1-
(2/3)*(mu_s/mu)).*f_rr.^3);
    F_rr = F_rr+(~(f_rr <= 3*mu*Fzr))*mu_s*Fzr;

    Fxrr = Cxr*(Krr./(1+Krr)).*(F_rr./f_rr);
    Fyrr = -Car*(tan(alpharr)./(1+Krr)).*(F_rr./f_rr);
end

FxrV = Fxrr;
FyrV = Fyrr;

%Compute derivatives
derivUy = (FylV+FyrfV+FylrV+FyrrV)./mass - r.*Ux;

```

```

derivr = (a*(FylfV+FyrfV)-b*(FylrV+FyrrV) + d.*(FxrV+FxrV-FxlfV-FxlrV))./Izz;
derivUx = (FxlV+FxrV+FxlV+FxrV)./mass + r.*Uy;
derivB = derivUy./Ux - (Ux).^2.*derivUx.*Uy;
end

```

## Baseline Controller Code

```

% Baseline Controller Model - Steering Inputs Only
% By Christina Boyd
% This code uses the 4 wheel braking model.

clc
clear all
close all

% Single IC
numPoints = 1;

Uy = 0;
r = 0;
Ux = 28;

% Grid of ICs Points
UyBound = [-30 30];
rBound = [-0.72 0.72];
UxBound = [6 36];

[mu, mu_s, mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr, Trailbraking_Radius_Des] =
VehicleConstants4Wheel_7_19_16(1);

mu = [0.3, 0.4, 0.7, 0.9];
mu_s = mu;

%% Euler's Streamline
dt = 0.0005; % seconds
oo = 1;

t = (0:dt:10)'; % s

for fff = 1:length(mu)
% Phase Portrait with controlled steering and braking
livePlots = 0;

for i = 1:length(Uy)
for j = 1:length(r)
for k = 1:length(Ux)

clear xdot
clear ydot
clear zdot
clear NWorld
clear EWorld
clear HWorld

```

```

clear x      % Uy
clear y      % r
clear z      % Ux
clear SWorld
clear eWorld
clear DeltaHWorld
clear Heading_dot
clear North_dot
clear East_dot
clear S_dot
clear e_dot
clear Delta_H_dot
clear delta
clear Trailbraking_Radius
clear RadiusOfCurvature

% Starting Position
x(1) = Uy(i);      % Uy
y(1) = r(j);      % r
z(1) = Ux(k);     % Ux

% World Frame
NWorld(1) = 0;
EWorld(1) = Trailbraking_Radius_Des;      % m
HWorld(1) = 0;
SWorld(1) = 0;
eWorld(1) = 0;
DeltaHWorld(1) = 0;
RadiusOfCurvature(1) = NaN;
delta(1) = 0;

pp = 1;
while pp <= length(t) && (x(pp) >= min(UyBound) && x(pp) <= max(UyBound) && y(pp) >=
    min(rBound) && y(pp) <= max(rBound) && z(pp) >= min(UxBound) && z(pp) <= max(UxBound))

    Trailbraking_Radius(pp) = Trailbraking_Radius_Des;

    [xdot(pp), ydot(pp), zdot(pp), ~] = FourWheelNonlinDynWithGrad(x(pp), y(pp), z(pp), delta(pp), [0, 0, 0,
        0], mu(ff), mu_s(ff), mass, a, b, L, d, Izz, g, Cxf, Caf, Fzf, Cxr, Car, Fzr, Trailbraking_Radius(pp));

    % Next Time Step
    x(pp+1) = x(pp) + xdot(pp)*dt;      % Uy
    y(pp+1) = y(pp) + ydot(pp)*dt;      % r
    z(pp+1) = z(pp) + zdot(pp)*dt;      % Ux

    RadiusOfCurvature(pp+1) = sqrt(x(pp+1).^2 + z(pp+1).^2) ./ y(pp+1);

    % Compute world Derivatives
    Heading_dot(pp) = y(pp);
    North_dot(pp) = z(pp)*cos(HWorld(pp)) - x(pp)*sin(HWorld(pp));
    East_dot(pp) = -x(pp)*cos(HWorld(pp)) - z(pp)*sin(HWorld(pp));

    S_dot(pp) = z(pp)*cos(DeltaHWorld(pp)) - x(pp)*sin(DeltaHWorld(pp));
    e_dot(pp) = x(pp)*cos(DeltaHWorld(pp)) + z(pp)*sin(DeltaHWorld(pp));
    Delta_H_dot(pp) = y(pp) - S_dot(pp)/Trailbraking_Radius(pp);

    % Determine where vehicle is in the world frame
    NWorld(pp+1) = NWorld(pp) + North_dot(pp)*dt;
    EWorld(pp+1) = EWorld(pp) + East_dot(pp)*dt;

```

```

HWorld(pp+1) = HWorld(pp) + Heading_dot(pp)*dt;

SWorld(pp+1) = SWorld(pp) + S_dot(pp)*dt;
eWorld(pp+1) = eWorld(pp) + e_dot(pp)*dt;
DeltaHWorld(pp+1) = DeltaHWorld(pp) + Delta_H_dot(pp)*dt;

PropGainSteer = 0.2;
d_LA = 10;           % m
delta(pp+1) = -PropGainSteer*(eWorld(pp) + d_LA*DeltaHWorld(pp));

    pp = pp+1;
end

tEulerStream{1,oo} = t(1:pp-1);
EulerStreamDynamics{1,oo} = [x(1:end-1)', y(1:end-1)', z(1:end-1)'];
EulerStreamInputs{1,oo} = [delta(1:end-1)' Trailbraking_Radius'];

EulerStreamWorld{1,oo} = [EWorld(1:end-1)', NWorld(1:end-1)', RadiusOfCurvature(1:end-1)'];

PositionRelativeToRoad{1,oo} = [SWorld(1:end-1)', eWorld(1:end-1)', DeltaHWorld(1:end-1)'];

oo = oo + 1;

% If plotting during computation is desired, plot the most
% recently completed streamline on the already opened plot
if livePlots
    figure(1)
    plot3(x', y', z')
end

end
end
end

```