12-2022

Safety-Aware Longitudinal and Lateral Control of Autonomous Vehicles

Randy Guadalupe Chapa
The University of Texas Rio Grande Valley

SAFETY-AWARE LONGITUDINAL AND LATERAL CONTROL OF AUTONOMOUS VEHICLES

A Thesis

by

RANDY GUADALUPE CHAPA

Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE IN ENGINEERING

Major Subject: Electrical Engineering

The University of Texas Rio Grande Valley

December 2022

SAFETY-AWARE LONGITUDINAL AND LATERAL CONTROL OF AUTONOMOUS VEHICLES

A Thesis
by
RANDY GUADALUPE CHAPA

COMMITTEE MEMBERS

Dr. Wenjie Dong
Chair of Committee

Dr. Hasina Huq
Committee Member

Dr. Alexander Domijan
Committee Member

December 2022

ABSTRACT

Chapa, Randy G., Safety-Aware Longitudinal and Lateral Control of Autonomous Vehicles. Master of Science in Engineering (MSE), December 2022, 71 pp., 2 tables, 22 figures, references, 28 titles.

Safety is undoubtedly the most critical design requirement regarding autonomous vehicle controllers. This research considers an autonomous vehicle to keep a desired distance from the leader vehicle, as well as stay centered within the lane. To achieve this, the lateral control problem and the combined longitudinal and lateral control problem were studied. Adaptive control laws were proposed with the aid of the backstepping technique and the barrier function technique. Simulation was done to verify the effectiveness of the proposed control laws.

DEDICATION

For my friends and family, for always having faith in me.

ACKNOWLEDGMENTS

Working with Dr. Wenjie Dong has been challenging yet rewarding. Through his expertise and experience, I was able to propose some impactful work that I am proud of. I would also like to thank Dr. Hasina Huq and Dr. Alexander Domijan for always being available to listen to my concerns or questions throughout my time at UTRGV. This committee, with their vast knowledge of different areas of Electrical Engineering, I was able to challenge myself during this pursuit of the master's degree, which is all I could've hoped for, upon enrolling.

I would like to also acknowledge funding provided by the National Science Foundation CREST Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (NSF Award No. 2112650) and the NSF grant no. ECCS-2037649. The opinions expressed in this thesis are solely mine and do not necessarily represent those of the NSF.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

TABLE OF FIGURES

Page

CHAPTER I

INTRODUCTION

Safety is the most important design element given the recent boom in autonomous vehicles (AVs) and applications. Any engineering system should, in theory, be considered "safe", but a safety-critical system prioritizes safety as a key design requirement. This serves as another motivation for low-level controller strategies that guarantee a control system—whether linear or non-linear—is regarded safe. With this knowledge, it is advisable to start thinking about a specific application for vehicle control, analyze safety factors, and then offer an alternate way that has advantages over methods that have already been written about. In general, certain currently used methods can be improved to create a new way that is unique to a particular application or physical model. In another section, this is discussed in more detail.

In order to uncover useful applications of control that is assured to be safe, I thought it would be appropriate to investigate driver statistics regarding vehicle control. It turns out that lane departures account for more than 39% of crash-related fatalities and are the leading cause of fatal accidents in the United States [1]. In addition, the National Highway Transportation Safety Administration (NHTSA) estimates that up to 1.5 million incidents are the result of human error, with unintentional lane changes accounting for a significant portion of these accidents. Such a lane-keeping scenario would fall under the category of a lateral control problem. With the issue identified, the goal of this thesis work is to create a lateral control problem controller with safety as the primary design consideration. This can be achieved through the integrator backstepping

technique, which is derived from the use of a Lyapunov function and a barrier function. The next step would be to consider a combined longitudinal and lateral control problem, as this would be a better practical application of an autonomous vehicle. A later section will include technical details on how this can be accomplished. Literature has extensively discussed both longitudinal and lateral control issues, but what distinguishes this study is the use of integrator backstepping. Once the appropriate controller has been created, this effort will be a distinctive addition to current techniques.

1.1 Literature Review

1.1.1 Safety Guarantees of a System

It is best to first define safety before diving into specifics of safety-guaranteed controllers that are described in the literature. The concept of safety was first discussed in [2] by Leslie Lamport in 1977, and once again articulated in [3]. Elegantly, it is proclaimed that safety requires for "bad" things to be avoided, while liveness requires that "good" things will eventually occur. A relevant liveness property that closely aligns with control systems is asymptotic stability, in which an asymptotically stable equilibrium point will eventually be reached. Fig. 1.1 contains the most elementary example of system stability, and we can think of reaching the stable equilibrium point as our "good" objective.

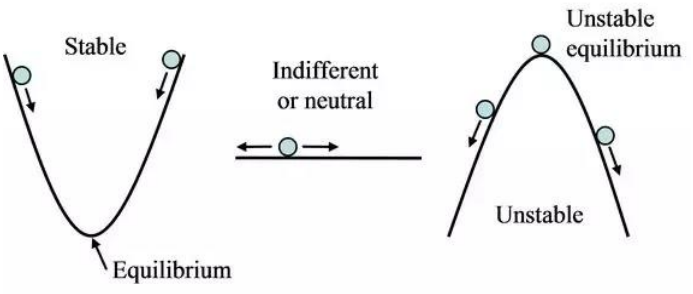

Figure 1.1 System Stability Ramp

As for a safety property, invariance is a common example, meaning any trajectory starting inside an invariant set will never reach the complement of the set. This further proves the idea of "bad" things never happening or being reached. Recall that the complement of a set is everything that isn't the original set itself, and a visual is shown in Fig. 1.2.

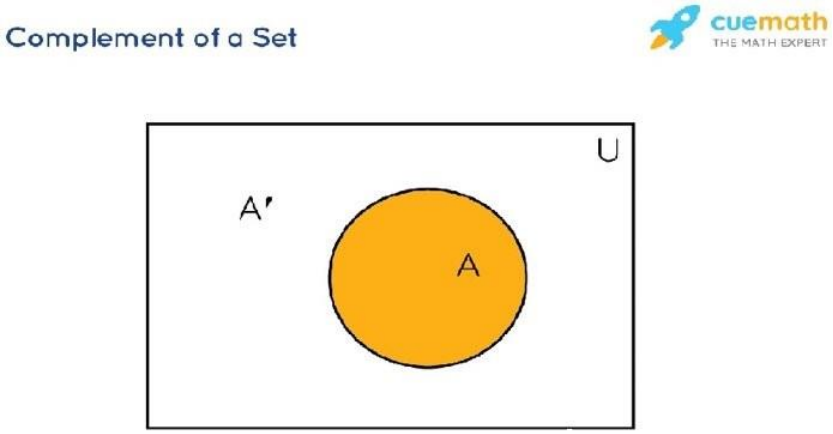

Figure 1.2 Complement of a Set

Let's say the set *A* is invariant, then any trajectory that starts in *A*, will guarantee to stay within that set, never reaching the complement. With the identification of invariance as a safety

property, and asymptotic stability as liveness, it is evident that safety has not been a focal point in control theory throughout history. This is true, mainly through existing Lyapunov functions that stabilize systems and will be further examined later in the next section.

1.1.2 Quadratic Program Controller

In order to achieve a liveness objective, together with a safety objective, a quadratic program (QP) is often the preferred choice. One such example is to implement an adaptive cruise control (ACC) experiment, that utilizes a QP controller to satisfy stability and safety constraints [4]. The ACC example is an ideal experiment to unify the liveness and safety objectives and is used in a vast number of works such as [5], [6] and [7]. ACC proves to be useful because we associate two objectives for a following vehicle to abide to: maintain a safe distance between the lead vehicle and to maintain a desired velocity. One can easily see that the minimum distance to be enforced satisfies a safety objective, and a desired velocity represents the liveness objective. The challenge becomes prioritizing one objective over the other, intuitively it is known that the safety objective should always be satisfied before the liveness objective should be considered. Most works classify the safety constraints as hard constraints, and the liveness as soft constraints.

Optimal controllers are the preferred method for the QP problems, as the target is to minimize the control input for the system in question. This proves to be useful, as in a practical scenario, the less energy that is needed to change the input, the better the performance of the system. Think of the ACC example, if the control input is a steering angle or a steering wheel force, then the objective would be to minimize the change of the angle or force, as a drastic change would impact the driver comfort. Driver comfort can be seen as the stability objective, which can be overlooked if the steering angle must undergo a drastic change to avoid a collision.

Another popular method that has been explored in literature is model predictive control (MPC). MPC considers a prediction horizon, to map out potential trajectories of a system, and determine which is best to follow. A basic visual for a discrete MPC design for better understanding can be seen in Fig. 1.3.

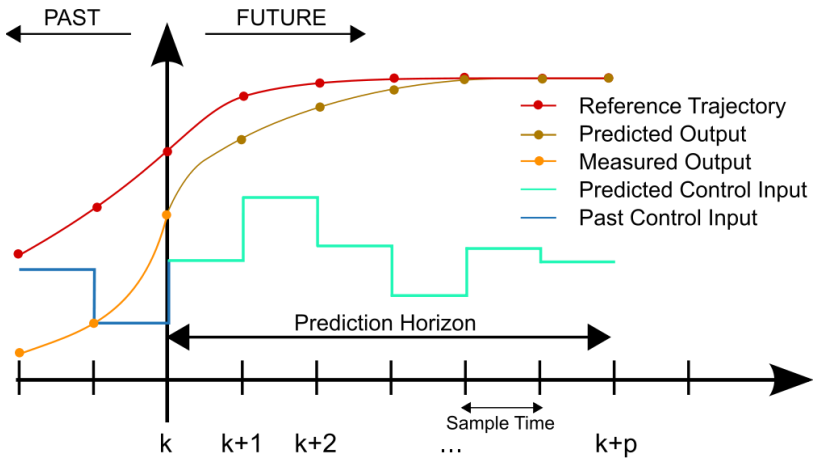

Figure 1.3 Model Predictive Control Scheme

From the figure, the designer selects a control horizon to consider, that provides one or several possible control inputs to follow. Finding the proper prediction horizon can be troubling, as selecting a large horizon can be computationally expensive, while a short horizon may not be able to account for an undesirable obstacle later in the trajectory. One can see how MPC could be used for driving scenarios, if the objective is to lane keep or to maintain travel behind a lead vehicle. Examples of MPC in literature include [8], [9], [10], [11] and [12] which also happens to incorporate the control Lyapunov function, which will be discussed in more detail in the next section. Overall, QP controllers have a broad use for control laws, and seem to be an ideal candidate for our design to test a lane keeping example.

1.1.3 Reinforcement Learning

Dynamic driving environments requires safe decision making, and one method that has emerged recently to handle such environments are reinforcement learning (RL) strategies. To understand how a generic RL system works, a visual is included in Fig. 1.4.

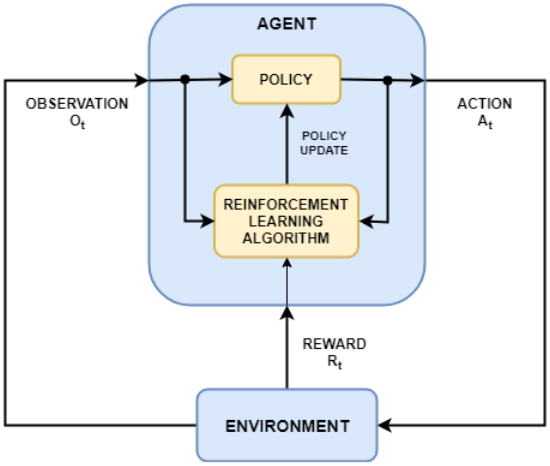

Figure 1.4 Reinforcement Learning Algorithm

Essentially, RL methods can adjust to a dynamic environment, by enforcing a policy and an algorithm that derives an action, based on an initial observation in a system. To better understand with an example, let's say the distance between a lead and following vehicle is small. The proper action would be to record this distance and call for an action that increases the distance, to the safe desired amount. A result of choosing the best action is a reward, and the main challenge when it comes to RL is deriving the best reward function. This is just an elementary introduction to RL and much more technical details can be found in [13], [14] and [15]. Several authors did an extensive review of existing RL approaches in [16]. For RL to address safety objectives, a proposed policy would need to adjust the algorithm as needed, in [17] a penalty is considered

that aims to avoid undesirable outcomes. In [18], the idea of Markov decision process (MDP) is implemented in the RL design. MDP is popular for optimization techniques, that computes solutions via dynamic programming.

Normally, a RL model can function in unknown environments and conditions but works such as [19] and [20] incorporate known, initial conditions to benefit the model's learning curve. This drastically improves design functions and having a baseline for certain interactions between the agent and the environment proves to be useful. To reiterate, RL methods have shown promising results for autonomous safety applications but are technically challenging for a simple lateral control problem like lane keeping. Future work could call for RL techniques, especially if configured with longitudinal control as well.

1.2 Thesis Content

This work will first consider an error based dynamic model from [1] for a lane keeping control problem. Next, entry level controllers are designed and compared to stabilize the dynamics. From there, safety requirements can be put in place, while additionally applying a quadratic programmable controller from existing literature. The QP controller uses the CLF and CBF as objective constraints. Finally, the integrator backstepping controller solution will stabilize the dynamics, and ensure that the vehicle stays within the lane, this will be the safety guarantee. Once the lateral control problem is successfully solved and proven to be a useful option, a combined longitudinal and lateral control will then be introduced. Again, the integrator backstepping controller will be designed to fit this model, and simulations will be conducted in MATLAB Simulink to verify the effectiveness.

Chapter I includes a brief analysis of existing safety guarantees in literature, for either lateral control, longitudinal control or both controlled together.

In Chapter II, the lateral dynamics are introduced, and the steady state formula is written. This equation will be the base of the work, with the goal being to stabilize and set some objectives of a lane keeping problem.

Chapter III will introduce some preliminary results on the dynamics from chapter II, using feedback control laws derived from two different methods, enforcing stability on the system.

Next, chapter IV considers an additional challenge by considering system dynamic uncertainties. This is done with the help of an integrator backstepping control law. Stability and safety objectives are to be met.

Chapter V includes a quadratic programmable controller, which not only enforces safety and stability objectives, but minimizes the control input.

Chapter VI will introduce the combined longitudinal and lateral control problem, and the proposed integrator backstepping controller will be designed to fit the model. Simulation is done to verify the effectiveness of the theory.

Finally, a brief conclusion will be in chapter VII, detailing the advantages of the proposed unified controller along with future work.

1.3 Thesis Contribution

The intention of this thesis is to contribute to the existing methods of safety critical controllers for autonomous vehicle research. A combined longitudinal and lateral control

problem will be solved through the integrator backstepping controller, which is derived from a Lyapunov function together with a barrier function. At first, a lateral control problem is discussed, then after it is compared to other methods in literature, the combined model will be introduced. For students that are interested in control systems and/or autonomous vehicle applications, the hope is that this work can provide a great introduction, as extensive technical knowledge isn't required to make sense of the results.

CHAPTER II

LATERAL CONTROL DYNAMICS

To reiterate, the first goal is to propose a lateral control system which guarantees safety. Basic lateral driving maneuvers include lane keeping and lane change. The corresponding control tasks are path following and trajectory tracking. For path following, the path reference is independent of time. Trajectory tracking considers time as a constraint of the reference path. This makes sense as a path can change over time in trajectory tracking if an obstacle were to appear or if a preceding vehicle is being closed in on. Lane keeping is an example of a path following task, as the goal is to stay within the lane, while continuing to follow the road and its curvature. Lane changes, however, would have to consider that the trajectory can change over time, and if an obstacle is in the path, then to avoid it, a lane change is required. Lane keeping alone is an easier task to handle; longitudinal control will also be considered later.

Longitudinal control has been researched heavily over the past few decades, with the main application tested on being cruise control systems. Beyond the scope of standard cruise control, adaptive cruise control (ACC) has been a great tool to prove the usefulness of proposed control theory works in literature. Like standard cruise control, ACC maintains a velocity that the user likes, but when a preceding vehicle is approached, the velocity is reduced, with the goal of maintaining a minimum distance between the vehicles. Interestingly, this is exactly how the

CBF, and CLF constraint should operate under a QP controller, with the minimum distance being the hard requirement, and the suggested velocity being the soft requirement.

2.1 Summary of Kinematic Model Equations

To begin a lateral control problem, a widely used model is needed to derive dynamics from, and for the entirety of the works done so far, we refer to the popular bicycle model in Fig 2.1 from the Rajamani controls textbook [1].

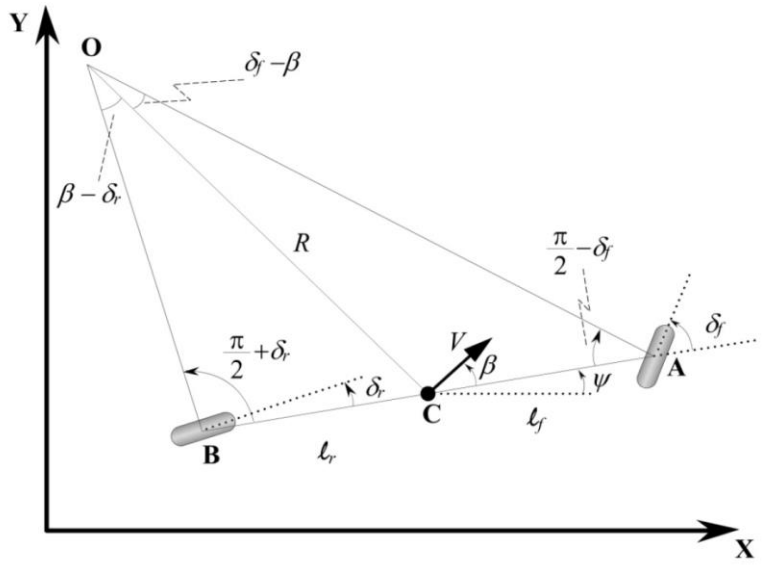

Figure 2.1 Bicycle Kinematics

In this model, the two left and right front wheels are to be represented by a single wheel, noted at point A. The same goes for the two rear wheels, represented at point B. Steering angles for the front wheels and rear wheels are noted as δ_f and δ_r respectively. In our case, only the front wheels can be steered, so δ_r is set to zero and the steering wheel angle will just be referred to as δ. Center of gravity (c.g.) of the vehicle is noted at point C. ℓ_f and ℓ_r represent the distances of

points A and B from point C, respectively. The wheelbase of a vehicle is the length from the center of the front wheels to the center of the rear wheels, and in this case, it is represented by $L = \ell_f + \ell_r$.

Regarding motion, the coordinates required to describe the vehicle's motion are *X*, *Y* and ψ. (*X*, *Y*) are inertial coordinates of the c.g. location, and ψ is the orientation of the vehicle. Velocity at the c.g. is denoted by *V* and an angle β is created with longitudinal axis of the vehicle. β is known as the slip angle. With the variables defined, now the kinematic model equations can be written, but it is only effective for applications that don't include external forces. They are noted here however, to better understand the dynamic equations next. The equations of motion are given by:

$$\dot{X} = Vcos(\psi + \beta), \tag{2.1}$$

$$\dot{Y} = Vsin(\psi + \beta), \tag{2.2}$$

$$\dot{\psi} = \frac{Vcos(\beta)}{\ell_f + \ell_r} tan(\delta), \tag{2.3}$$

$$\beta = \tan^{-1}\left(\frac{\ell_f tan(\delta) + \ell_r tan(\delta)}{\ell_f + \ell_r}\right). \tag{2.4}$$

The inputs for this model are velocity V and the steering wheel angle δ. It is worth noting that V can be constant, a time varying function or can be derived from a longitudinal vehicle model. For this study, V is assumed to be constant, and will be defined later in this chapter.

2.2 Bicycle Model of Lateral Vehicle Dynamics

Once the vehicle is at a higher speed, the assumption that each wheel's velocity is in the same direction of the respective wheel cannot be made. This would call for a dynamic model for

lateral vehicle motion. The canonical "bicycle" model with two degrees of freedom is chosen to represent the dynamics, seen in Fig. 2.2.

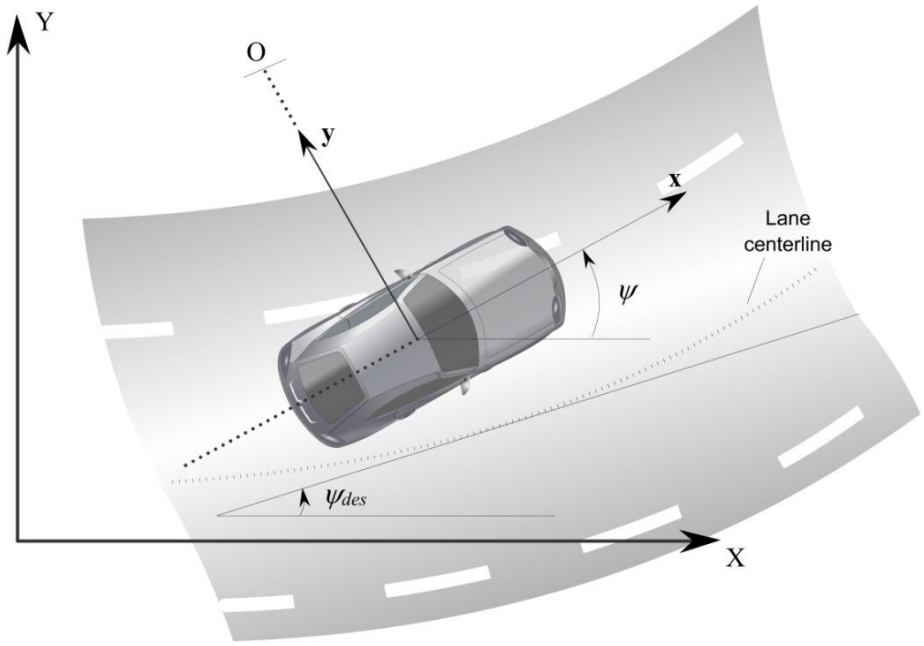

Figure 2.2 Lateral Vehicle Dynamics

The two degrees of freedom are represented by vehicle yaw angle ψ and lateral position y. The vehicle's center of rotation is notated by the point O, and lateral position is measured along the vehicle's lateral axis to that point. Measuring the vehicle yaw angle ψ is done with respect to the global *X* axis. V_x then represents the longitudinal velocity at c.g. of the vehicle.

2.2.1 State Space Model – Lateral Motion

Applying Newton's second law of motion along the global *Y* axis gives,

$$ma_y = F_{yf} + F_{yr}, \tag{2.5}$$

where $a_y = \left(\frac{d^2y}{dt^2}\right)_{intertial}$ denotes inertial acceleration of the vehicle from c.g. in the direction of the global *Y* axis. F_{yf} represents the front lateral tire force and F_{yr} represents the rear lateral tire force. a_y is computed from the terms acceleration $\ddot{y}$ along the global *Y* axis, and centripetal acceleration $V_x\dot{\psi}$. This leads to

$$a_y = \ddot{y} + V_x\dot{\psi}. \tag{2.6}$$

By substituting Eq. (2.6) into Eq. (2.5), an equation for lateral motion of the vehicle is defined as

$$m\left(\ddot{y} + V_x\dot{\psi}\right) = F_{yf} + F_{yr}. \tag{2.7}$$

Yaw dynamics about the global *Z* axis is defined as

$$I_z\ddot{\psi} = \ell_f F_{yf} - \ell_r F_{yr}, \tag{2.8}$$

where ℓ_f and ℓ_r are front tire and rear tire distances respectively, from c.g. of the vehicle.

Next, lateral tire force of the front wheels can be written as

$$F_{yf} = 2C_{af}\left(\delta - \theta_{Vf}\right), \tag{2.9}$$

where C_{af} is the constant cornering stiffness of each front tire, δ is the steering angle and θ_{Vf} is the front tire velocity angle. 2 comes from the pair of front wheels.

Lateral tire force for the rear wheels is then given as

$$F_{yr} = 2C_{ar}(-\theta_{Vr}), \tag{2.10}$$

where similarly, C_{ar} is the constant cornering stiffness of the rear tires, and θ_{Vr} is the rear velocity angle of the tire. As a clarification, this model is for a front wheel drive vehicle, therefore a rear steering angle is not considered. To calculate θ_{Vf} and θ_{Vr}, we use the relationships given from

$$tan\left(\theta_{Vf}\right) = \frac{V_y + \ell_f\dot{\psi}}{V_x}. \tag{2.11}$$

$$tan(\theta_{Vr}) = \frac{V_y - \ell_r \dot{\psi}}{V_x}. \tag{2.12}$$

By using small angle approximations, Eq. (2.11) and Eq (2.12) can be rewritten as

$$\theta_{Vf} = \frac{\dot{y} + \ell_f \dot{\psi}}{V_x}. \tag{2.13}$$

$$\theta_{Vr} = \frac{\dot{y} - \ell_f \dot{\psi}}{V_x}. \tag{2.14}$$

Finally, through substitution in Eqs. (2.7) and (2.8), the state space model can be written as

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{af}+2C_{ar}}{mV_x} & 0 & -V_x - \frac{-2C_{af}\ell_f + 2C_{ar}\ell_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2C_{af}\ell_f - 2C_{ar}\ell_r}{I_z V_x} & 0 & -\frac{2C_{af}\ell_f{}^2 + 2C_{ar}\ell_r{}^2}{I_z V_x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2C_{af}\ell_f}{I_z} \end{bmatrix} \delta. \tag{2.15}$$

With this system, the lateral position y and the yaw angle ψ can be monitored, for an application such as lane keeping. However, there is a more intuitive model, that can be directly used as a lane keeping problem, and that will be introduced in the next section.

2.2.2 Dynamics with Respect to Road

One way to develop a steering control system is to consider a set of dynamics with state variables in terms of position and orientation error, with respect to a given road. First, e_1 is defined as the distance of the c.g. of the vehicle from the center line of the lane. e_2 is the orientation error of the vehicle with respect to the road. It is assumed that the longitudinal

velocity V_x is constant, and the radius of the road is noted as R. Rate of change of desired orientation is given as:

$$\dot{\psi_{des}} = \frac{V_x}{R}. \tag{2.16}$$

Desired acceleration is written as:

$$\frac{V_x^2}{R} = V_x \dot{\psi_{des}}. \tag{2.17}$$

Now, $\ddot{e_1}$ and e_2 can be defined as follows:

$$\ddot{e_1} = \left(\ddot{y} + V_x \dot{\psi}\right) - \frac{V_x^2}{R} = \ddot{y} + V_x\left(\dot{\psi} - \dot{\psi_{des}}\right), \tag{2.18}$$

and

$$e_2 = \psi - \psi_{des}, \tag{2.19}$$

finally,

$$\dot{e_1} = \dot{y} + V_x(\psi - \psi_{des}). \tag{2.20}$$

Velocity is assumed to be constant, if it were not constant, then it would create a nonlinear, time varying model, which is not ideal for this study. This approach instead will design a linear time-invariant (LTI) model, but for continued work, a linear parameter varying (LPV) model can be considered in which velocity is the varying parameter. For a varying velocity, integrating Eq. (2.18) would obtain

$$\dot{e_1} = \dot{y} + \int V_x e_2 dt\,, \tag{2.21}$$

Next, substituting Eqs. (2.19) and (2.20) into (2.7) and (2.8) gives

$$m\ddot{e_1} = \dot{e_1}\left[-\frac{2}{V_x}C_{af} - \frac{2}{V_x}C_{ar}\right] + e_2\left[2C_{af} + 2C_{ar}\right],$$

$$+\dot{e_2}\left[-\frac{2C_{af}\ell_f}{V_x} + \frac{2C_{ar}\ell_r}{V_x}\right] + \dot{\psi_{des}}\left[-\frac{2C_{af}\ell_f}{V_x} + \frac{2C_{ar}\ell_r}{V_x}\right] + 2C_{af}\delta \tag{2.22}$$

as well as

$$I_z\ddot{e_2} = 2C_{af}\ell_1\delta + \dot{e_1}\left[-\frac{2C_{af}\ell_f}{V_x} + \frac{2C_{ar}\ell_r}{V_x}\right]$$

$$+e_2\left[2C_{af}\ell_f - 2C_{ar}\ell_r\right] + \dot{e_2}\left[-\frac{2C_{af}{\ell_f}^2}{V_x} - \frac{2C_{ar}{\ell_r}^2}{V_x}\right] \tag{2.23}$$

$$-I_z\ddot{\psi_{des}} + \dot{\psi_{des}}\left[-\frac{2C_{af}{\ell_f}^2}{V_x} - \frac{2C_{ar}{\ell_r}^2}{V_x}\right]$$

With all the given equations, the state space model, as tracking error variables is given by:

$$\frac{d}{dt}\begin{bmatrix} e_1 \\ \dot{e_1} \\ e_2 \\ \dot{e_2} \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{af}+2C_{ar}}{mV_x} & \frac{2C_{af}+2C_{ar}}{m} & \frac{-2C_{af}\ell_f+2C_{ar}\ell_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2C_{af}\ell_f - 2C_{ar}\ell_r}{I_zV_x} & \frac{2C_{af}\ell_f - 2C_{ar}\ell_r}{I_z} & -\frac{2C_{af}{\ell_f}^2 + 2C_{ar}{\ell_r}^2}{I_zV_x} \end{bmatrix}\begin{bmatrix} e_1 \\ \dot{e_1} \\ e_2 \\ \dot{e_2} \end{bmatrix}$$

$$+\begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2C_{af}\ell_f}{I_z} \end{bmatrix}\delta + \begin{bmatrix} 0 \\ \frac{-2C_{af}\ell_f+2C_{ar}\ell_r}{mV_x} - V_x \\ 0 \\ -\frac{2C_{af}{\ell_f}^2 + 2C_{ar}{\ell_r}^2}{I_zV_x} \end{bmatrix}\dot{\psi_{des}}. \tag{2.24}$$

In compact form, Eq. (2.24) can be rewritten as:

$$\dot{x} = Ax + B_1\delta + B_2\dot{\psi_{des}}, \tag{2.25}$$

with x being the state space vector containing $[e_1 \quad \dot{e_1} \quad e_2 \quad \dot{e_2}]^T$. Stabilizing the dynamics in Eq. (2.25) would satisfy the tracking objective of our steering control problem. Recall that the control input of the system is the steering angle δ in degrees. Road bank angle is not considered for this problem. Eq. (2.25) will serve as the main control problem for the rest of this work, and as mentioned previously, it isn't a widely used model in existing literature.

2.3 Summary

Finally, for reference, the summary of the dynamic model equations is included in Fig. 2.3.

Table 2.1: Summary Table of Dynamic Model Equations

Symbol	Nomenclature	Equation
x	State space vector	$x = [e_1 \quad \dot{e_1} \quad e_2 \quad \dot{e_2}]^T$
		$\dot{x} = Ax + B_1\delta + B_2\dot{\psi_{des}}$
		Matrices found in Eq. (2.24)
e_1	Lateral position error with respect to the road	$\ddot{y} + V_x(\dot{\psi} - \dot{\psi_{des}})$
e_2	Yaw angle error with respect to the road (orientation error)	$e_2 = \psi - \psi_{des}$
δ	Front wheel(s) steering angle	
$\dot{\psi_{des}}$	Desired yaw rate	$\dot{\psi_{des}} = \frac{V_x}{R}$

CHAPTER III

LANE KEEPING CONTROL

In the previous chapter, the lateral vehicle dynamics were introduced, and the primary model was broken down by each term that makes up the steady-state model. It was discussed that by stabilizing the dynamics, this would satisfy the control objective of automatic steering, e.g., lane keeping of a vehicle. This chapter will discuss and compare some control designs through state feedback. Simulations are conducted in MATLAB, with results presented near the end of this chapter.

3.1 State Feedback Control

Recall from Chapter II, that the state space model to be considered is given as

$$\dot{x} = Ax + B_1\delta + B_2\dot{\psi_{des}}, \tag{3.1}$$

with $x = [e_1 \quad \dot{e_1} \quad e_2 \quad \dot{e_2}]^T$, the objective is to ensure that the lateral position error e_1 is as small as possible.

Vehicle parameters of a sedan are included in Figure 3.1 and will be used for simulations through the entirety of this work.

Table 3.1 Parameters for MATLAB						
m = 1573	ℓ_f = 1.1	ℓ_r = 1.58	C_{af} = 80000	C_{ar} = 80000	I_z = 2873	$\dot{\psi}_{des} = V_x$ / R
x_0 = [0; 0.3; 1; 0.1]				V_x = 30 m/s	I_z = 2873	R = 1000 m

In Eq. (3.1), the matrix *A* is unstable, with two of its eigenvalues at the origin, this is proven with the MATLAB command

```
eig(A) >> eig(A)

ans =

0.0000 + 0.0000i

-6.8308 + 5.0278i

-6.8308 - 5.0278i

-0.0000 + 0.0000i
```

To stabilize the system, feedback is considered first. The preliminary state feedback law to be tested is

$$\delta = -Kx = -k_1 e_1 - k_2 e_2 - k_3 e_3 - k_4 e_4, \tag{3.2}$$

with eigenvalues of our matrix $A - BK$ placed at any location of our choosing. The system becomes closed-loop, and the resulting feedback controller is now given as

$$\dot{x} = (Ax - B_1 K)x + B_2 \dot{\psi}_{des}, \tag{3.3}$$

To compute the feedback matrix *K*, first a vector *P* containing the desired eigenvalue locations is selected. For this study, the same values are chosen as in [1], with *P* containing the vector

$[-5 - 3j \quad -5 + 3j \quad -7 \quad -10]^T$. Then the following MATLAB command is used to compute *K*

```
K = place(A, B1, P);
```

With parameters defined, and a newly configured closed-loop system from Eq. (3.3), simulations are done to check if lateral error e_1 and yaw angle error e_2 converge to a stable value, that is close to zero. First, the log of e_1 over time is shown in Fig. 3.2.

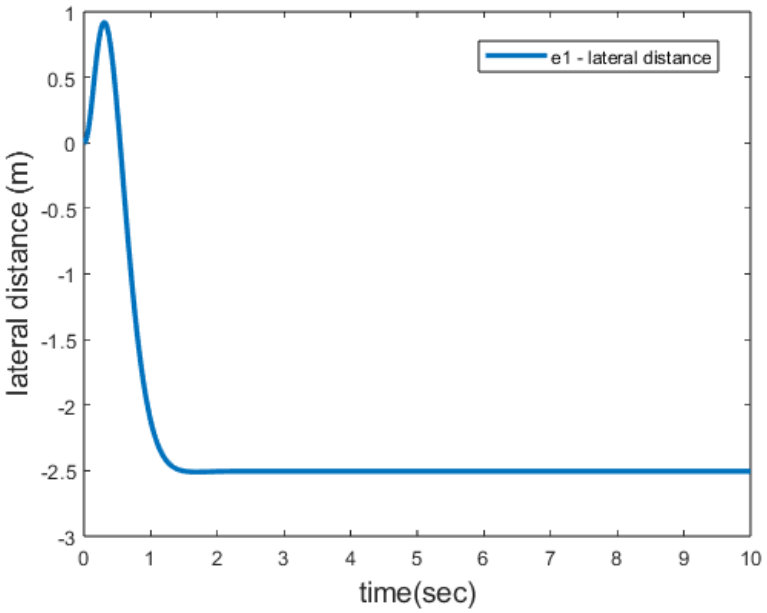

Figure 3.1 Lateral Position Error (pole placements)

From the initial condition, the error surges, then settles around -2.5 m. One might see concern that it does not converge to zero, but due to the term $B_2\dot{\psi}_{des}$ in Eq. (3.3), e_1 does not need to converge to zero. This would make sense for a generic controller, but in our case, we want this error to be less than a predefined constant. Later, this will be enforced. Next, a time history of the orientation error e_2 is shown in Fig. 3.3.

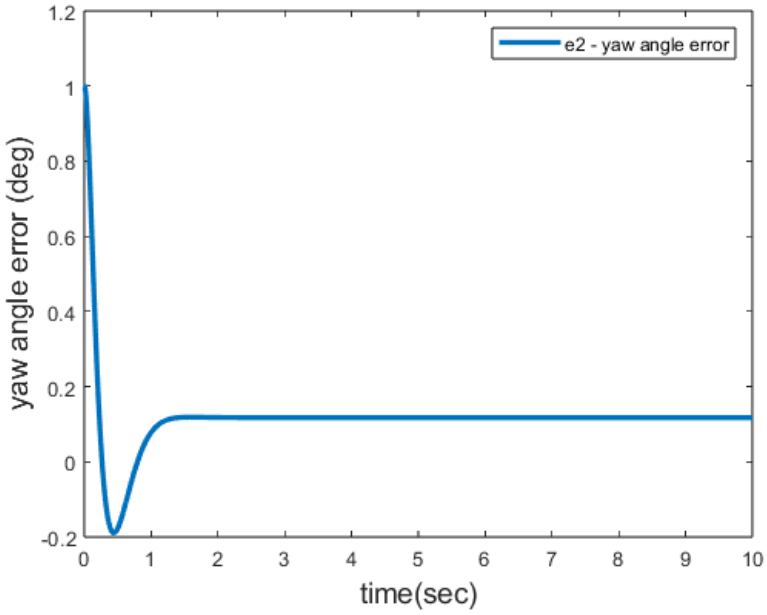

Figure 3.2 Yaw angle error (pole placements)

Once again, e_2 does not converge to zero, this is because the desired yaw rate $\dot{\psi_{des}}$ is a non-zero value. It is possible to include a feedforward term to ensure that e_1 converges to zero, but e_2 cannot converge to zero due to the curvature of the road. More detail on this reasoning can be found in Rajamani [1].

3.1.1 State Feedback via Riccati Equation

Another method of designing a state feedback controller for our system is through the use of a continuous algebraic Riccati equation (CARE). Before, we chose a feedback matrix *K* using eigenvalue pole placements. To achieve better performance, we can define a different *K* as

$$K_1 = R^{-1}{B_1}^T P, \tag{3.4}$$

where R is assumed to a real, positive definite or a real symmetric matrix [21] and P must satisfy the CARE

$$A^T P + PA - PB_1 R^{-1} {B_1}^T P + Q = 0 \tag{3.5}$$

where both R and Q are selected to be identity matrices. To solve the Riccati equation, MATLAB has the command

```
[X, K, L] = icare(A, B1, Q)
```

where X contains the unique solution matrix, K contains the state-feedback gain, and L contains the closed-loop eigenvalues. Then Eq. (3.3) can be re-calculated with the new feedback gain K.

With an alternate closed-loop system in place, the same simulations can be done for our state variables e_1 and e_2 to see any differences in performance. The same initial condition vector as before is used here. First, Fig. 3.4 contains the time history of lateral position error e_1.

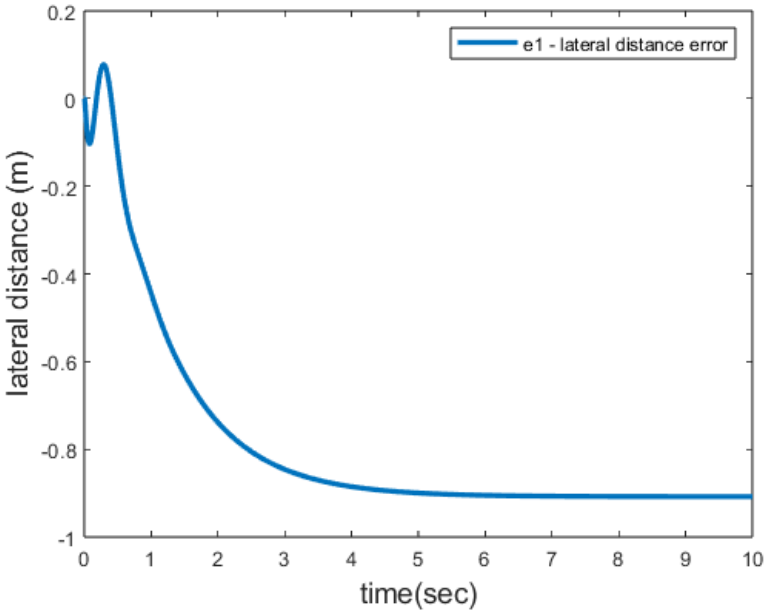

Figure 3.3 Lateral Position Error (Riccati)

It is shown that the initial overshoot around 0.5 seconds isn't as large as the pole placement method, not much greater than zero. Before, the lateral error was around -2.5m but it does converge to that point slightly faster than the Riccati method. The next chapter will conduct the same experiment, but with a predetermined maximum lateral displacement that the system must stay below.

Next, the time history of the yaw angle error e_2 is shown in Fig. 3.5.

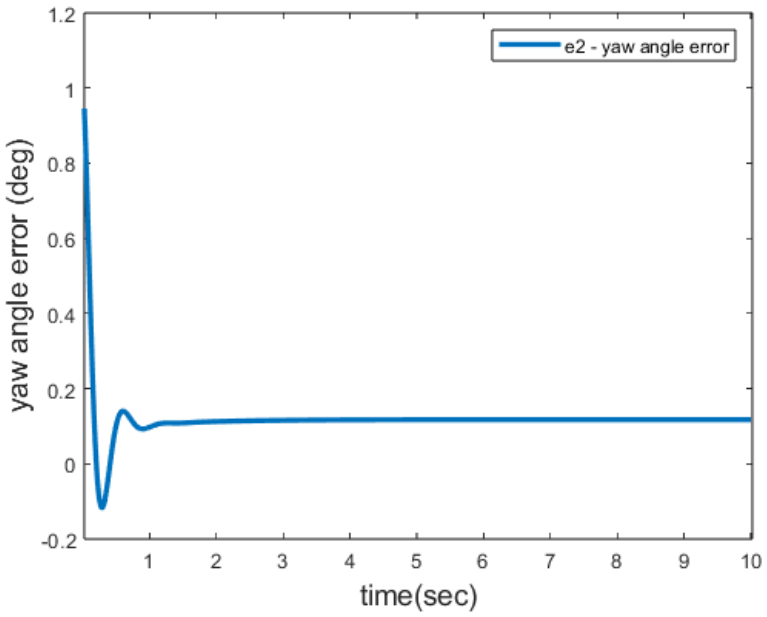

Figure 3.4 Yaw angle error (Riccati)

Interestingly, the performance is roughly the same as the pole placement method, only the initial overshoot is slightly less. For the two methods shown in this chapter, they successfully impose stability on the previously unstable system, with the errors converging to a finite number. However, one cannot say that safety is guaranteed for this system, for all time. To do so, a safety objective needs to be formulated, and this will be chosen in the next chapter.

3.2 Summary

This chapter introduced the lateral control problem to be considered for this work, first stabilizing the steady-state dynamics of Eq. (3.1). Pole placements are computed to derive a feedback law in Eq. (3.2), in which the system states converge to a finite value, as shown in the simulation Figs. (3.2, 3.3).

Next, another method of computing the feedback matrix, K is shown through the CARE method. Again, simulations are conducted, and new results are shown in Figs. (3.4, 3.5). While both methods do stabilize the system, there are no safety guarantees for all time, therefore a safety objective must be chosen to further improve on the system.

CHAPTER IV

INTEGRATOR BACKSTEPPING TECHNIQUE

Along with the surge in safety-critical research for vehicle applications, another challenge can arise through some unknown parameters. One may not know explicitly some parameters of the system dynamics. In this case, such parameters include the vehicle mass, cornering stiffness of the tires, wheelbase of the vehicle and the inertial moment balance. The only predetermined parameter that is concrete is the velocity, which has been set to 30 m/s for the previous simulations. To address this challenge, we will apply the integration backstepping (IB) design, which is detailed in [26]. First, we apply the IB to the original system dynamics and a simulation will be conducted. Next, we consider the unknown parameters, derive an estimation of the dynamics, and conduct a simulation. The results will be compared and verify if the safety requirement is met.

4.1 Integrator Backstepping

We begin by considering the system

$$\dot{\eta} = f(\eta) + g(\eta)\xi \tag{4.1}$$

$$\dot{\xi} = u \tag{4.2}$$

where $[\,\eta^T\ \xi\,]^T \in R^{n+1}$ is the state and $u \in R$ is the control input. Both functions, $f : D \to R^n$ and $g : D \to R^n$ are smooth in domain, containing $\eta = 0, f(0) = 0$. Now, state feedback

control law can be designed to stabilize the origin ($\eta = 0, \xi = 0$). First, we assume that f and g are known. This system is a cascade connection of two components, being Eq. (4.1) with an input ξ and the integrator Eq. (4.2). Cascade control considers two controllers, with the output of one controller providing a set point for another, and they are configured in a feedback loop. Eq. (5.1) can be stabilized by the feedback control law $\xi = \phi(\eta),\ \phi(0) = 0$

that is the origin of

$$\dot{\eta} = f(\eta) + g(\eta)\phi(\eta) \tag{4.3}$$

is asymptotically stable. Let the Lyapunov function $V(\eta)$ satisfies the following inequality

$$\frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\phi(\eta)] \leq -W(\eta), \qquad \forall \eta \in D \tag{4.4}$$

with $W(\eta)$ being positive definite. Now, through adding and subtracting $g(\eta)\phi(\eta)$ on the right side of Eq. (4.2), we rewrite the equations as

$$\dot{\eta} = [f(\eta) + g(\eta)\phi(\eta)] + g(\eta)[\xi - \phi(\eta)] \tag{4.5}$$

$$\dot{\xi} = u \tag{4.6}$$

Next, the change of variables is written as

$$z = \xi - \phi(\eta) \tag{4.7}$$

which results in the system

$$\dot{\eta} = [f(\eta) + g(\eta)\phi(\eta)] + g(\eta)z \tag{4.8}$$

$$\dot{z} = u - \dot{\phi} \tag{4.9}$$

All the forms of our equations can be seen in Fig. 4.1.

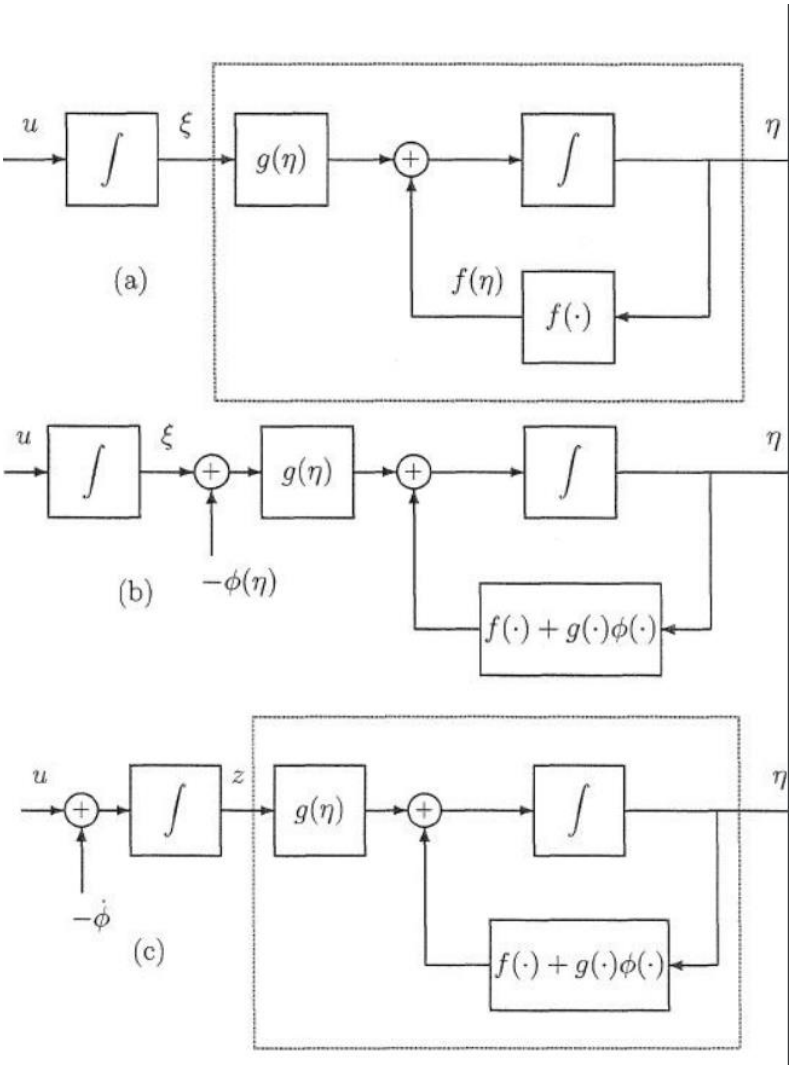

Figure 4.1 Block Diagram of Eqs. (4.1), (4.2)

In Fig. 4.1, the transition from 4.1(b) to 4.1(c) is the backstepping from - $\phi(\eta)$ to the integrator.

Now, the derivative on $\dot{\phi}$ can be computed using

$$\dot{\phi} = \frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\xi] \tag{4.10}$$

By rewriting the integrator term in Eq. (4.9) taking $v = u - \dot{\phi}$, we get a cascade connection, in the form like the original system in Eqs. (4.1, 4.2).

$$\dot{\eta} = [f(\eta) + g(\eta)\phi(\eta)] + g(\eta)z \tag{4.11}$$

$$\dot{z} = v \tag{4.12}$$

The difference in Eq. (4.11) from the original, is that asymptotically stable origin exists, at an input of zero. Next, we need to design v to stabilize the entire system. This can be done using

$$V_c(\eta, \xi) = V(\eta) + \frac{1}{2}z^2 \tag{4.13}$$

as a candidate of a Lyapunov function. Next, we obtain

$$\dot{V}_c = \frac{\partial V}{\partial \eta}[f(\eta) + g(\eta)\phi(\eta)] + \frac{\partial V}{\partial \eta}g(\eta)z + zv \tag{4.14}$$

$$<= -W(\eta) + \frac{\partial V}{\partial \eta}g(\eta)z + zv$$

choosing

$$v = -\frac{\partial V}{\partial \eta}g(\eta) - kz, \qquad k > 0 \tag{4.15}$$

now yields

$$\dot{V}_c \leq -W(\eta) - kz^2 \tag{4.16}$$

This proves that at the origin ($\eta = 0,\ z = 0$) is asymptotically stable. By $\phi(0) = 0$, it is concluded that the origin ($\eta = 0,\ \xi = 0$) is also asymptotically stable. We then can substitute for v, z and $\dot{\phi}$, to obtain a state feedback control law

$$u = \frac{\partial \phi}{\partial \eta}[f(\eta) + g(\eta)\xi] - \frac{\partial V}{\partial \eta} g(\eta) - k[\xi - \phi(\eta)] \tag{4.17}$$

It is concluded that the origin is globally asymptotically stable. These findings are summarized in the lemma below, which was written by Khalil [26].

Lemma 5.1 Consider the system Eq. (4.1) - (4.2). Let $\phi(\eta)$ be a stabilizing state feedback control law for Eq. (5.1) with $\phi(0) = 0$, and $V(\eta)$ be a Lyapunov function satisfies Eq. (4.4) for some positive definite function $W(\eta)$. Then, the state feedback control law in Eq. (4.17) stabilizes the origin of Eq. (4.1) - (4.2), with $V(\eta) + [\xi - \phi(\eta)]^2/2$ as a Lyapunov function. Moreover, if all the assumptions hold globally and $V(\eta)$ is " radially unbounded", the origin will be globally asymptotically stable.

4.2 Integrator Backstepping for Known Vehicle Dynamics

Now, we can apply this backstepping technique to our system dynamics, which are naturally unstable, and achieve the goal of stabilizing the output states. Recall, that we also require that the state lateral position error $|e_1| \leq 0.9m$.

Recall the position with respect to the road dynamics as

$$\dot{x} = Ax + B_1\delta + B_2\dot{\psi_{des}} \tag{4.18}$$

where we write the matrices using coefficients as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{a_{22}}{V_x} & a_{23} & \frac{a_{24}}{V_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{a_{42}}{V_x} & a_{43} & \frac{a_{44}}{V_x} \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 0 \\ b_{12} \\ 0 \\ b_{14} \end{bmatrix},$$

$$b_2 = \begin{bmatrix} 0 \\ \frac{b_{22}}{V_x} - V_x \\ 0 \\ \frac{b_{24}}{V_x} \end{bmatrix}.$$

We want to design a controller δ such that $|x_1| \leq c$. where we rewrite the x state vector as $[x_1 \quad x_2 \quad x_3 \quad x_4]^T$. First, we will design the controller, assuming the matrices are well known.

Case 1: Assume A, b_1 and b_2 are well known.

Define a Lyapunov function

$$V_1 = \frac{1}{2} log \frac{c^2}{c^2 - x_1^2} \tag{4.19}$$

then

$$\dot{V_1} = \frac{x_1 \dot{x_1}}{c^2 - x_1^2} = \frac{x_1 x_2}{c^2 - x_1^2} \tag{4.20}$$

We can design a stabilizing function α_1 as

$$\alpha_1 = -(c^2 - x_1^2) k_1 x_1 \tag{4.21}$$

where $k_1 > 0$, then

$$\dot{V}_1 = -k_1 x_1^2 + \frac{x_1(x_2 - \alpha_1)}{c^2 - x_1^2} \tag{4.22}$$

Let $z = x_2 - \alpha_1$, we have

$$\dot{z} = \frac{a_{22}x_2}{V_x} + a_{23}x_3 + \frac{a_{24}x_4}{V_x} + b_{12}\delta + \left(\frac{b_{22}}{V_x} - V_x\right)\dot{\psi_{des}} \tag{4.23}$$

Define a Lyapunov function as

$$V_2 = V_1 + \frac{1}{2}z^2 \tag{4.24}$$

then

$$\dot{V}_2 = \dot{V}_1 + z\dot{z}$$

$$= -k_1 x_1^2 + \frac{x_1 z}{c^2 - x_1^2} + z\left(\frac{a_{22}x_2}{V_x} + a_{23}x_3 + \frac{a_{24}x_4}{V_x} + b_{12}\delta + \left(\frac{b_{22}}{V_x} - V_x\right)\dot{\psi_{des}}\right) \tag{4.25}$$

Finally, we can choose a controller

$$\delta = b_{12}^{-1}\left(-\frac{a_{22}x_2}{V_x} - a_{23}x_3 - \frac{a_{24}x_4}{V_x} - \left(\frac{b_{22}}{V_x} - V_x\right)\dot{\psi_{des}} - \frac{x_1}{c^2 - x_1^2} - k_2 z\right) \tag{4.26}$$

where $k_2 > 0$, then

$$\dot{V}_2 = -k_1 x_1^2 - k_2 z^2 \tag{4.27}$$

this means that x_1 and x_2 converge to zero.

We can choose the input δ from Eq. (4.26) and conduct simulation, to verify our safety bounds. The first run, we choose the same initial condition as the previous chapter as

$$x_0 = [\,0\ m;\ 0.3\ m/s\ ;\ 1\ deg\ ;\ 0.1\ deg/sec]$$

The time history of lateral distance, x_1 and lateral velocity x_2 is shown in Fig. 4.2.

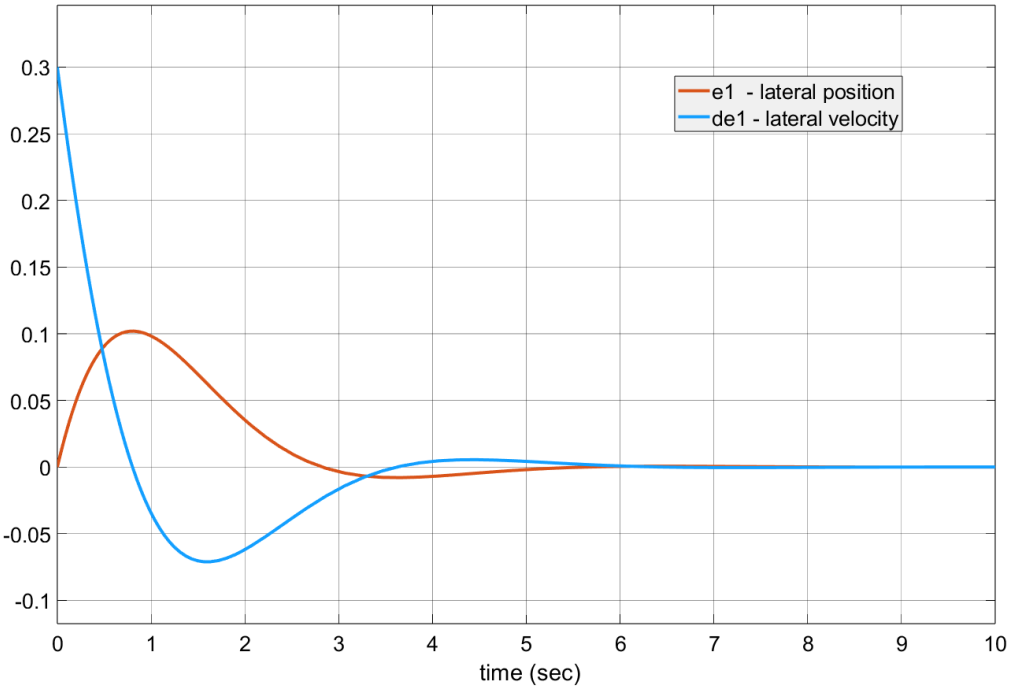

Figure 4.2 Integrator Backstepping of Known Dynamics - x_1 and x_2

The error distance manages to not only stabilize itself, but it converges back to zero, which wasn't accomplished with the previous controllers. Previously, it was said that the lateral error, will not converge to zero due to the disturbance term $B_2\dot{\psi_{des}}$. However, this control law can get lateral distance and lateral velocity (x_1, x_2) to converge to zero.

Next, we want to plot the other states, yaw angle (x_3) and yaw rate (x_4). Both are plotted in Fig. 4.3.

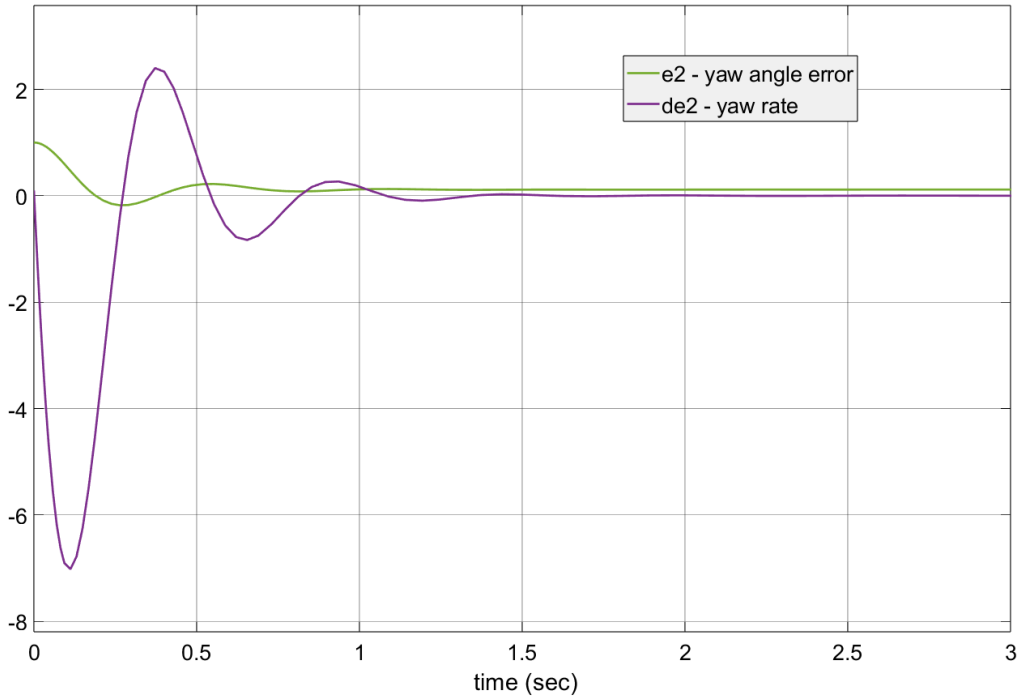

Figure 4.3 Integrator Backstepping of Known Dynamics - x_3 and x_4

Interestingly, we first see that all states except the yaw angle (x_3) converge to zero, which was not achieved in the other controllers. Now, with the integrator backstepping technique, we guarantee that x_1 and x_2 will converge to zero. Also, we know that if x_2 and x_4 are zero, then we know what x_1 and x_3 will not change.

With the integrator backstepping controller, we ensured that the lateral distance from the center of the lane is at a "perfect" zero. In a practical setting, this is ideal, as the vehicle is centered, and would in theory be at the safest location in the event of a lane drift. The next challenge is to consider some system dynamic uncertainties, in the case of a simulation in which we are uncertain regarding the system or even the environment of the study.

4.3 Integrator Backstepping with System Dynamics Uncertainties

It is reasonable to assume for future works and other works involving vehicle parameters, that uncertainties will arise. We will perform the same process as the previous section, except we will consider unknown parameters a_{ij} and b_{ij}, with the only fully known parameter we choose is $V_x = 30m/s$.

First, we design the stabilizing function α_1 as

$$\alpha_1 = -(c^2 - x_1^2)k_1 x_1 \tag{4.28}$$

where $k_1 > 0$, then

$$\dot{V}_1 = -k_1 x_1^2 + \frac{x_1(x_2 - \alpha_1)}{c^2 - x_1^2} \tag{4.29}$$

We let $z = x_2 - \alpha_1$, we have

$$\dot{z} = \frac{a_{22}x_2}{V_x} + a_{23}x_3 + \frac{a_{24}x_4}{V_x} + b_{12}\delta + \left(\frac{b_{22}}{V_x} - V_x\right)\dot{\psi_{des}} \tag{4.30}$$

Next, we define $\hat{a}_{ij}$ and $\hat{b}_{ij}$ to be estimates of a_{ij} and b_{ij}. A Lyapunov function is then defined as

$$V_2 = V_1 + \frac{1}{2}z^2 + \frac{\gamma_1}{2}\tilde{a}_{22} + \frac{\gamma_2}{2}\tilde{a}_{23} + \frac{\gamma_3}{2}\tilde{a}_{24} + \frac{\gamma_4}{2}\tilde{b}_{12} + \frac{\gamma_5}{2}\tilde{b}_{22} \tag{4.31}$$

where $\tilde{a}_{ij} = a_{ij} - \hat{a}_{ij}$ and $\tilde{b}_{ij} = b_{ij} - \hat{b}_{ij}$. Then we get

$$\dot{V}_2 = -k_1 x_1^2 + \frac{x_1 z}{c^2 - x_1^2} + z\left(\frac{a_{22}x_2}{V_x} + a_{23}x_3 + \frac{a_{24}x_4}{V_x} + b_{12}\delta + \left(\frac{b_{22}}{V_x} - V_x\right)\dot{\psi_{des}}\right)$$

$$+ \gamma_1\tilde{a}_{22}\dot{\tilde{a}}_{22} + \gamma_2\tilde{a}_{23}\dot{\tilde{a}}_{23} + \gamma_3\tilde{a}_{24}\dot{\tilde{a}}_{24} + \gamma_4\tilde{b}_{12}\dot{\tilde{b}}_{12} + \gamma_5\tilde{b}_{22}\dot{\tilde{b}}_{22} \tag{4.32}$$

Now, we can choose the controller δ as

$$\delta = \hat{b}_{12}^{-1}\left(-\frac{\hat{a}_{22}x_2}{V_x} - \hat{a}_{23}x_3 - \frac{\hat{a}_{24}x_4}{V_x} - \left(\frac{\hat{b}_{22}}{V_x} - V_x\right)\dot{\psi_{des}} - \frac{x_1}{c^2 - x_1^2} - k_2 z\right) \tag{4.33}$$

where Eq. (4.32) can be rewritten as

$$\dot{V_2} = -k_1 x_1^2 - k_2 z\text{^}2 + z\left(\frac{\tilde{a}_{22}x_2}{V_x} + \tilde{a}_{23}x_3 + \frac{\tilde{a}_{24}x_4}{V_x} + \hat{b}_{12}\delta + \left(\frac{\tilde{b}_{22}}{V_x}\right)\dot{\psi_{des}}\right)$$

$$+ \gamma_1\tilde{a}_{22}\dot{\tilde{a}}_{22} + \gamma_2\tilde{a}_{23}\dot{\tilde{a}}_{23} + \gamma_3\tilde{a}_{24}\dot{\tilde{a}}_{24} + \gamma_4\tilde{b}_{12}\dot{\tilde{b}}_{12} + \gamma_5\tilde{b}_{22}\dot{\tilde{b}}_{22} \tag{4.34}$$

The update laws are defined as

$$\dot{\hat{a}}_{22} = \gamma_1{}^{-1}\frac{zx_2}{V_x} \tag{4.35a}$$

$$\dot{\hat{a}}_{23} = \gamma_2{}^{-1}zx_3 \tag{4.35b}$$

$$\dot{\hat{a}}_{24} = \gamma_3{}^{-1}\frac{zx_4}{V_x} \tag{4.35c}$$

$$\dot{\hat{b}}_{12} = \gamma_4{}^{-1}z\delta \tag{4.35d}$$

$$\dot{\hat{b}}_{22} = \gamma_5{}^{-1}\frac{z\dot{\psi_{des}}}{V_x} \tag{4.35e}$$

which finally gives

$$\dot{V_2} = -k_1 x_1^2 - k_2 z^2 \tag{4.36}$$

This controller ensures that states x_1, x_2 and x_4 converge to zero, but the yaw angle error x_3 will not converge, due to the curvature of the road. γ_x must be greater than 0, the larger the value, the

faster the states converge to zero. For the first simulation, they will be set to 1, and their impact will be shown.

The same initial condition from before is used again also with $\gamma_x = 1$, and the history of x_1 and x_2 is shown in Fig. 4.4.

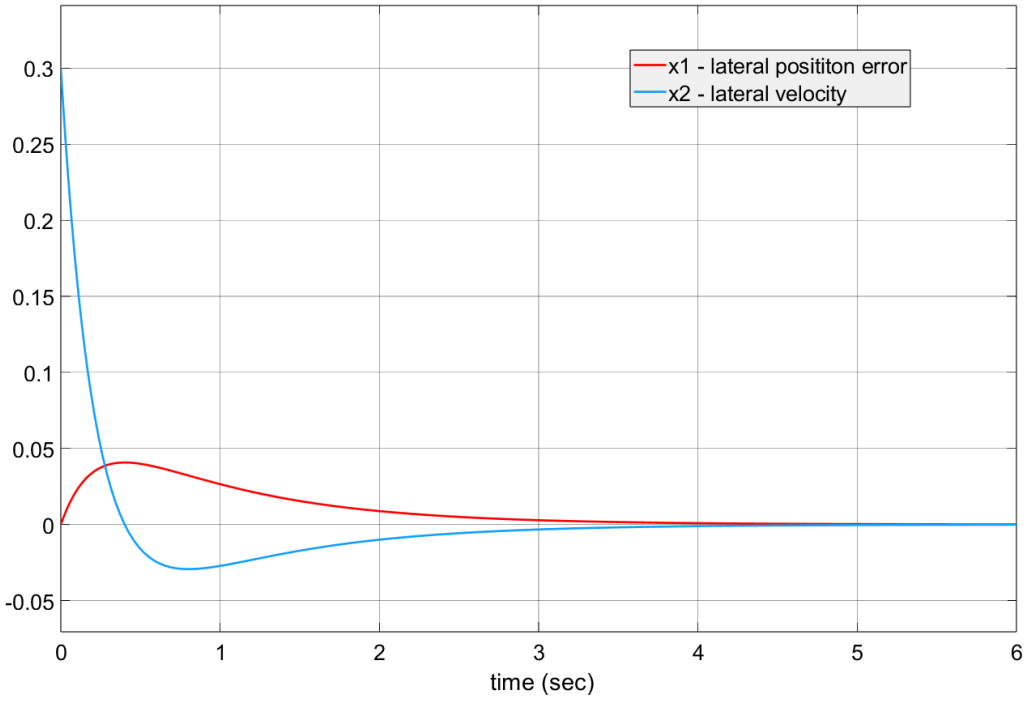

Figure 4.4 Integrator Backstepping with Uncertainties - x_1 and x_2

This is the first result, with the lateral states converging to zero as expected. Fig. 4.5 contains the time histories of the yaw states, in which the yaw angle (x_3) does not converge to zero, as anticipated.

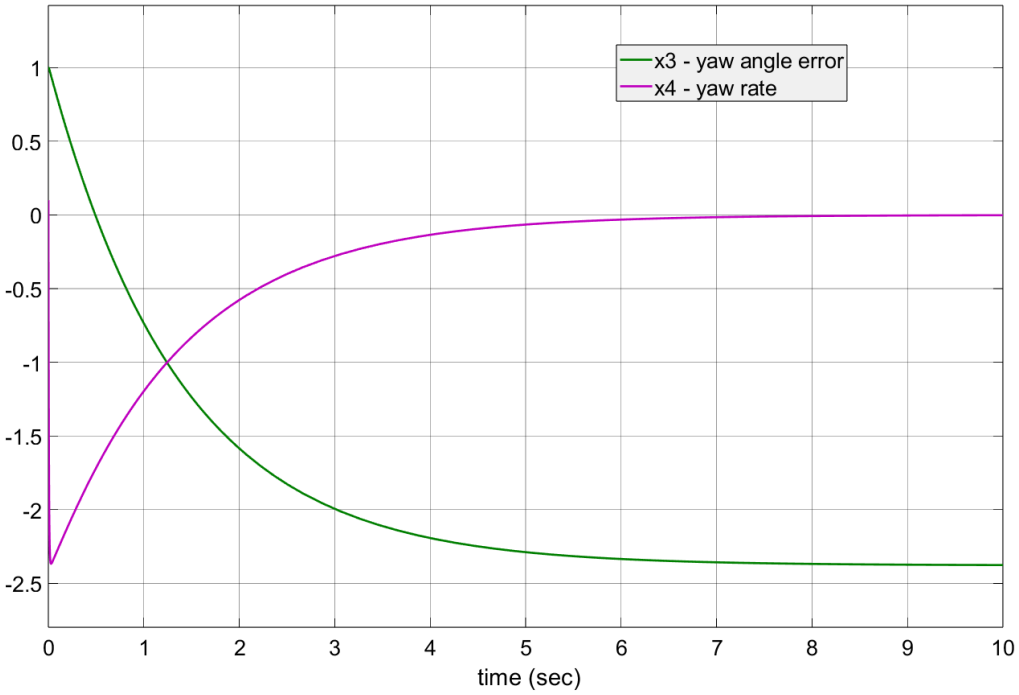

Figure 4.5 Integrator Backstepping with Uncertainties - x_3 and x_4

The performance of the uncertainties controller is a promising result, stabilizing the states, and the lateral distance is bounded within the requirement. The significant difference between the known and unknown controller parameters, is the yaw angle error. However, since the lateral states at zero, and the yaw angle still very small, it still satisfies our safety requirement.

CHAPTER V

CONTROL LYAPUNOV AND CONTROL BARRIER FUNCTIONS

Having presented some basic feedback controllers as well as the proposed method of the integrator backstepping controller, it is relevant to consider an existing method too. The unified controller utilizing control Lyapunov functions (CLF) and control barrier functions (CBF) through a quadratic problem such as designed in [4] and [5].

Throughout this thesis, and through the definitions in this chapter, consider the following nonlinear affine control system:

$$\dot{x} = f(x) + g(x)u, \tag{5.1}$$

with f and g as locally Lipschitz, meaning it is differentiable everywhere, x is positive definite, containing all real numbers, and u contains the set of admissible inputs.

5.1 Control Lyapunov Functions

First, stabilizing the system stability objective will be enforced through a CLF. Let's begin by aiming to stabilize the control system in Eq. (5.1) to a point $x^* = 0$, i.e., that drives $x(t) \to 0$. This is possible and in a nonlinear context, it is done by choosing a feedback control law that drives V, a positive definite function to zero. It is shown as:

$$\exists u = k(x) \quad s.t. \quad \dot{V}\big(x, k(x)\big) \leq -\gamma\big(V(x)\big), \tag{5.2}$$

where

$$\dot{V}\big(x, k(x)\big) \;=\; L_f V(x) \;+\; L_g V(x) k(x),$$

and with this, the system becomes stable to $V(x*) \;=\; 0$. γ is a class K function, and in control theory, they are used for system stability tests. They are continuous, strictly increasing, and $\gamma(0) \;=\; 0$. $L_f V(x)$ and $L_g V(x)$ are the Lie-derivatives of $V(x)$ along $f(x)$ and $g(x)$, respectively.

The aforementioned facts presented lead to the idea of the CLF, in which it is demonstrated that a function V can stabilize a system without structuring a feedback controller such as $u \;=\; k(x)$. Initially noted in [22], [23], [24], all that is required for a controller is to satisfy the inequality in Eq. (5.2). It is now established that if V is positive definite and meets the following criteria:

$$\inf_{u \in U} \big[L_f V(x) + L_g V(x) u\big] \leq -\gamma\big(V(x)\big), \tag{5.3}$$

then V is a valid CLF. Through the definition in Eq. (5.3), it is seen that u is constrained to satisfy the inequality as well, this allows the formulation of optimization-based controllers. This is key, as once the CBF definition and constraint is defined later in this chapter, we can structure an optimization controller that enforces multiple constraints on the system. As a general remark, we the following for CLFs from [25].

Theorem 1. *For the nonlinear control system (5.1), if there exists a control Lyapunov function* $V : D \;\rightarrow \mathbb{R}_{\geq 0}$*, i.e., a positive definite function satisfying Eq. (5.3), then any Lipschitz continuous feedback controller* $u(x)$ *that also satisfies Eq. (5.3) asymptotically stabilizes the system to x* = 0.* [4]

5.2 Control Barrier Functions

Stability was shown to drive a system to a particular point or set, this being the liveness property that has been mentioned previously. Safety, again, enforces invariance of a set, staying within the safe set. We can define a set C as a superlevel set of the continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$, that gives

$$\begin{aligned} C &= \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\}, \\ \partial C &= \{x \in D \subset \mathbb{R}^n : h(x) = 0\}, \\ Int(C) &= \{x \in D \subset \mathbb{R}^n : h(x) > 0\}. \end{aligned} \tag{5.4}$$

C is known to be the safe set.

Now, let $u = k(x)$ be a feedback controller, such as previously defined in Eq. (3.2), be applied to the system in (5.1)

$$\dot{x} = f_{cl(x)} \coloneqq f(x) + g(x)k(x). \tag{5.5}$$

If Eq. (5.5) is locally Lipschitz, this means that the system does not drastically change over a short amount of time. The derivative for any point must be bounded. Under this assumption, if the initial *x* state, x_0 is real, and results in a unique solution x_t eventually, then the set *C* is forward invariant. For proof, refer to [26].

Similar to the structure of the CLF, a CBF function *h* needs to be chosen. First, *h* is a valid candidate if $\frac{\partial h}{\partial x} \neq 0$ for all $x \in \partial C$ and class K_∞ function α exists. Then, *h* must satisfy:

$$\exists\, u \;\; s.t. \;\; \dot{h}(x,u) \geq -\alpha\big(h(x)\big), \qquad \alpha \in K_\infty. \tag{5.6}$$

With the above defined, the proper defined of a CBF *h* can be expressed as:

$$\sup_{u \in U} \left[L_f h(x) + L_g h(x) u \right] >= -\alpha\big(h(x)\big). \tag{5.7}$$

Again, like the CLF criteria defined in Eq. (5.3), u is constrained to only the set of control inputs that render the system safe. As a result, found through [4], is that if a CBF exists for the system in question, this means that the control system is deemed safe.

Theorem 2. *Let C be a set defined as superlevel set of a continuously differentiable function $h: D \subset \mathbb{R}^n \to \mathbb{R}$. If h is a CBF on D and $\frac{\partial h}{\partial x} \neq 0$ for all $x \in \partial C$, then any controller u(x) for the system in Eq. (5.1) renders the set C safe. [4]*

One remark, again a result found from [4], is that disturbances could drive the system out of the safe set *C*, but since the controller *u(x)* satisfies Eq. (5.6) it is also guaranteed that the system will return to the safe set, therefore *C* is asymptotically stable as well.

Finally, it is noted that CBFs provide the best conditions to enforce safety, both necessary and sufficient.

Theorem 3. *Let C be a set defined as the superlevel set of a continuously differentiable function $h: D \subset \mathbb{R}^n \to \mathbb{R}$ with the property that $\frac{\partial h}{\partial x} \neq 0$ for all $x \in \partial C$. If there exists a control law u=k(x) that renders C safe, i.e., C is forward invariant with respect to Eq. (5.5), then $h|_C: C \to \mathbb{R}$ is a CBF on C. [4]*

5.3 Optimization Based Control

After it has been determined that control barrier functions provide (required and sufficient) conditions for safety, the question of how to create controllers arises. Importantly, we want to accomplish so with the least amount of disruption possible, which means we'll only

make minor changes to an existing controller to ensure safety. As a result of this requirement, optimization-based controllers emerge naturally as a contender to handle this problem.

5.3.1 CBF Constraint

Given the nominal feedback controller $u=k(x)$ for the affine system back in Eq. (5.1), the objective is to guarantee safety. However, the chosen $k(x)$ may not be an eligible controller for the CBF condition in Eq. (5.7). To adjust this controller to guarantee safety, we would like to do so while being minimally invasive as possible. From the affine relationship on u in Eq. (5.7), a new controller design can be written, one that is safety critical. This calls for a quadratic program (QP) based controller, and we can define one on u as:

$$u(x) = \underset{u\in\mathbb{R}^m}{argmin}\frac{1}{2}\|u - k(x)\|^2 \tag{5.8}$$

$$s.t.\ L_f h(x) + L_g h(x)u \ \geq\ -\alpha\big(h(x)\big)$$

If the system has no input constraints, then only the CBF inequality constraint exists, and Eq. (5.8) becomes a valid closed-loop solution that guarantees safety.

5.3.2 CLF Constraint

Having structured the QP formula for a safety-critical controller, a medium of unifying both safety and stability requirements has emerged. Combining multiple objectives into an optimization controller has been done with CLFs, in both [27] and [28]. Considering the CLF criteria in Eq. (5.3) into the QP controller, we have:

$$u(x) = \underset{u,\delta\ \in\mathbb{R}^{m+1}}{argmin}\frac{1}{2}u^T H(x)u + p\delta^2 \tag{5.9}$$

$$s.t. \qquad L_f V(x) + L_g V(x) u \;\leq\; -\gamma\big(V(x)\big) + \delta$$

$$L_f h(x) + L_g h(x) u \;\geq\; -\alpha\big(h(x)\big)$$

in which $H(x)$ is any positive definite matrix. Additionally, a relaxation term, δ is considered to ensure the QP is feasible and is penalized by $p > 0$. To ensure a solution is computed, the stability constraint can be relaxed. This is done because for the safety critical system, safety should always be guaranteed and meeting the constraint should be a priority over stability.

The controller in Eq. (5.9) will serve as the base of this chapter, and simulations will be conducted following a similar problem formulation. Although both stability and safety objectives can be achieved through the controller, input constraints are not considered. This can lead to an infeasible optimization problem, and if safety cannot be guaranteed, then the safety-critical design is flawed. This is addressed in [5], through pointwise feasibility. For this work, we will not consider input constraints on the lateral control problem.

5.4 Lane Keeping via QPs

Finally, simulations can be conducted on the lane keeping problem we conducted in the previous chapter. In literature, most simulations are done for ACC applications, with a desired velocity acting as the CLF constraint, and the minimum distance behind the lead vehicle acting as the CBF constraint. For a lane keeping problem, we would like to ensure that the error state vector *x* is stable for all time, and this will act as the CLF constraint. The CBF constraint will require that the vehicle stays "centered" in the lane, assuming a constant longitudinal velocity like before. Applying the LK problem using the QP defined in the previous subsection, will ask to provide a steering input angle to keep the vehicle in the center of the lane. There will be no input constraint for this simulation. First, the original dynamics are rewritten in the affine form.

Then the safety and stability constraints will be defined, finally results of the errors will be plotted.

The lateral vehicle dynamics from Eq. (3.1) is rewritten in the control affine form as:

$$\dot{x} = \underbrace{Ax + B2\dot{\psi}_{des}}_{f(x)} + \underbrace{B1u}_{g(x)}, \tag{5.10}$$

where x is the state error vector containing $[e_1 \quad \dot{e_1} \quad e_2 \quad \dot{e_2}]^T$. The control input δ is the steering angle, and is rewritten as u for simplicity. Next, the control objectives can be introduced.

5.4.1 Safety Objective for LK

For this safety-critical design, we want to reiterate that the safety requirement is enforced as a hard constraint, which means that it must not be violated at any point over time. To choose such constraint for a LK problem, we know that e_1 is the lateral distance between the c.g. of the vehicle to the center of the lane. Thus, we would want to ensure the distance stays under a certain maximum. In the U.S., the width of a lane is 12ft, and the typical car width is around 6ft. This leaves 3ft, on each side of the vehicle to maneuver within. 3ft is about 0.9 meters, giving us the term $e_{1_{max}}$. This gives us the hard constraint to be enforced on e_1 as:

$$|e_1| \leq e_{1_{max}} \tag{5.11}$$

Now, the CBF written to enforce safety on Eq. (5.10) is:

$$h = e_{1_{max}} - sign(e_1)e_1 \tag{5.12}$$

5.4.2 Stability Objective for LK

For stability, we would like to ensure that the entire state vector x stays finite, this can be done easily through the general Lyapunov function:

$$V = x^T P x \tag{5.13}$$

Recall that the system (5.10) is unstable by nature, so by choosing the CLF to stabilize all the states, a feasible solution is guaranteed for all time. And if a feasible solution is guaranteed, coexisting with the hard CBF safety constraint, we have a valid safety-critical controller design. As far as input constraints, it is known that the range of a steering angle is:

$$-30° \leq u \leq 30°$$

However, this is not considered for the simulations done in this work.

5.5 Simulation Results

To compare the results of this controller versus the previous chapter, first the initial condition will be tested. Further, e_1, e_2 and their respective velocities will be adjusted to see the differences in performance from varying initial states. The initial state vector from the previous chapter was chosen as $x_0 = [\, 0\ m;\ 0.3\ m/s\ ;\ 1\ deg\ ;\ 0.1\ deg/sec]$. It is worth mentioning that an initial condition that does not satisfy the CBF constraint will not compute a result, and this is what we want from a safety-critical system. We know that it can still, in theory, converge back to the safe set, but safety conditions are strictly enforced.

First, the time histories of the lateral position error and orientation error are found in Fig. 5.1.

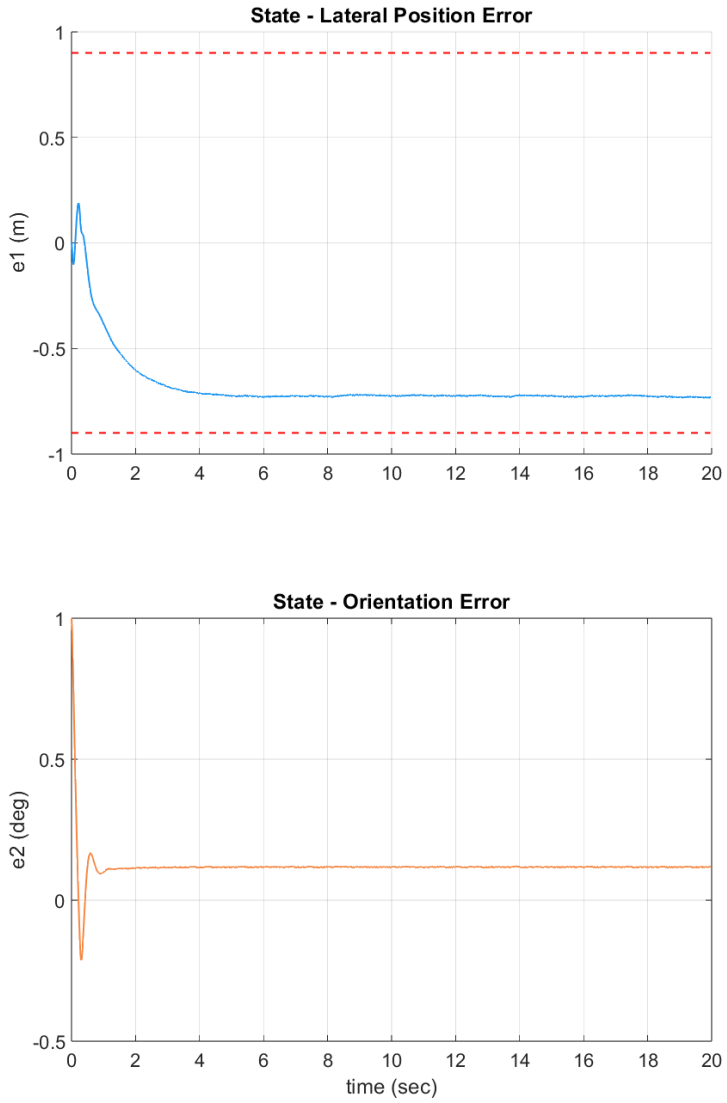

Figure 5.1 CBF-CLF-QP Controller

We can see that both displacements converge to a finite value, and notably e_1 stays within the safe set, defined by the region in red borders. The orientation error e_2 also converges to a very small value, which is ideal for a vehicle along a road curvature.

When compared to the pole placement method in the previous chapter, e_1 does not follow the safety constraint and was never capable of doing so. Although the value settles to a finite value, it would not meet our requirement of guaranteeing safety for all time. However, the Riccati method manages to just stay in the safe bounds. While a nice performance, keep in mind an adjustment to the initial condition can lead to a safety violation, that is if the initial states are also within the bounds. This is proven in a "stress test", next we will choose $e_1(0) = 0.89m$. Additionally, it makes sense to set $e_2(0) = 0\ deg$, therefore it will stay at zero for other state adjustments. Fig. 5.2 shows lateral displacement, using the Riccati control method from the previous chapter. Fig. 5.3 uses the QP method, both tests with the new initial state for $e_1(0)$.

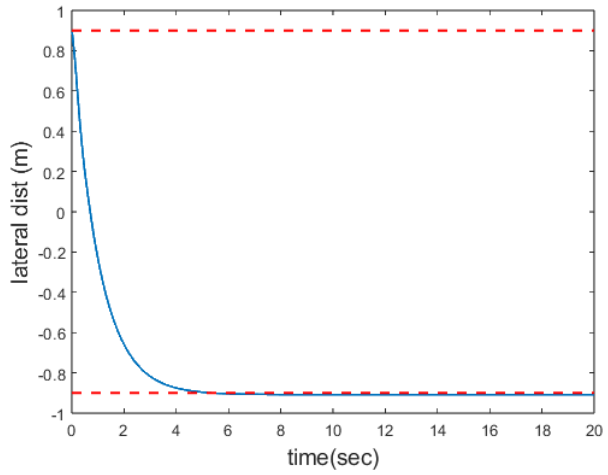

Figure 5.2 Riccati Stress Test

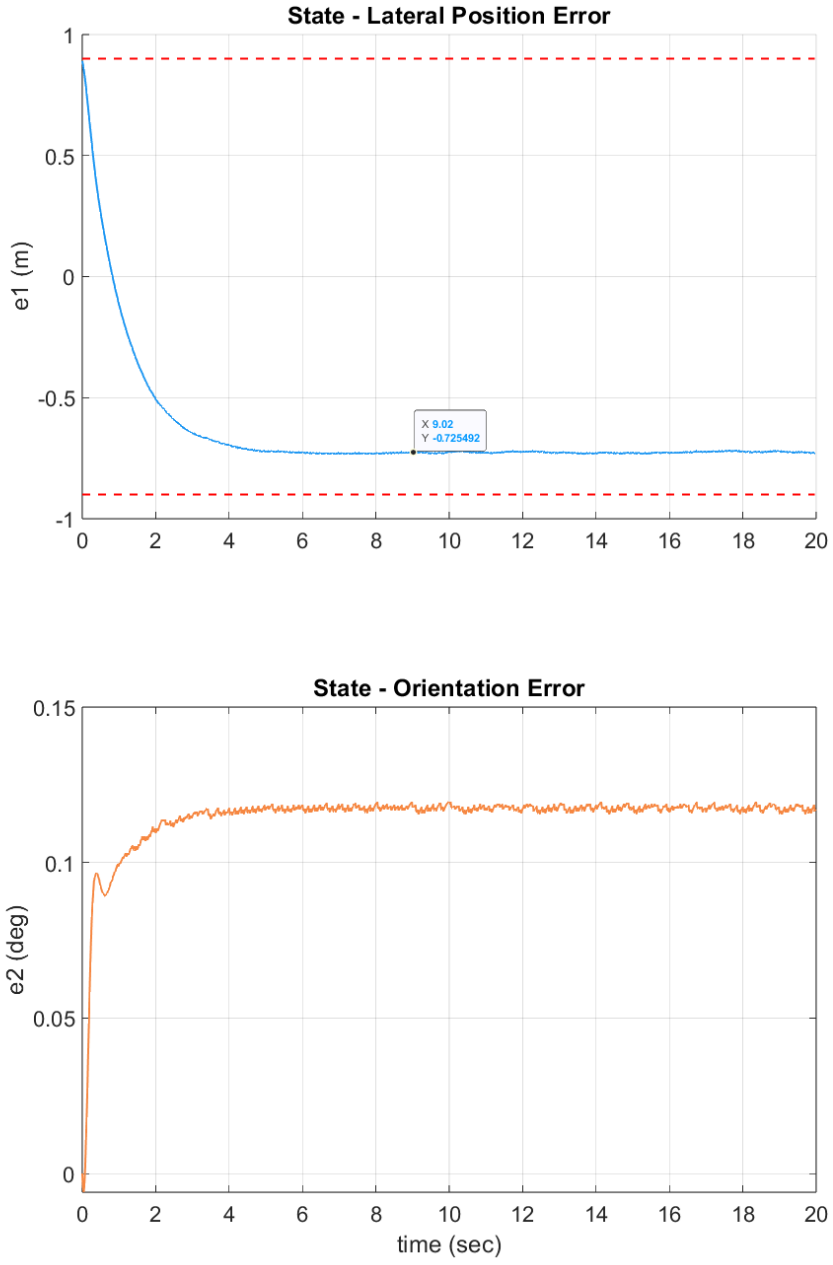

Figure 5.3 CBF-CLF-QP Stress Test

It is confirmed that the Riccati method does not satisfy the hard constraint by nature. Even if it is by less than 1% different than the 0.9 mark, any violation renders the system unsafe. In a

practical example, this would result in a potential collision. As for the QP controller, the lateral position error quickly moves away from the border of the unsafe set and settles at a comfortable displacement value.

5.6 Conclusion

It is proven that the CBF-CLF-QP controller can guarantee safety as a design requirement. Although a liveness property can be relaxed to achieve safety, the system still proves to be an impactful design for various control problems. This controller would outperform the integrator backstepping controller, but what the CBF-CLF controller doesn't consider is uncertainties in the dynamics. Additionally, the desired error states do not converge to zero, which will be a requirement going forward to guarantee safety. These challenges are to be addressed in the next chapter.

CHAPTER VI

COMBINED LONGITUDINAL AND LATERAL CONTROL

Thus far, this work has only considered lateral control of a vehicle, a study of path following. For a more realistic, and useful study, we now want to consider combined longitudinal and lateral control of a vehicle. This scenario will consider a follower vehicle, and the leader vehicle traveling along a curved road. The follower vehicle must maintain a desired distance from the leader, as well as stay centered in the lane. To achieve this, we will once again design an adaptive feedback controller through the integrator backstepping technique, like shown in chapter IV.

6.1 System Dynamics

First, let's consider the dynamics of the follower and leader vehicle. In Fig. 6.1, the model of the leader and follower vehicle are shown. *F* is the follower, *L* is the leader, *R* is the constant radius of the path, and we define $\kappa = 1/R$ as the curvature of said path. The position of the leader vehicle is noted by the coordinates x_L, y_L. ϕ_L is the yaw angle of the leader, V_L and ω_L are the velocity and angular velocity of the leader. Finally, a_L and Ω_L are the acceleration and angular acceleration of the leader.

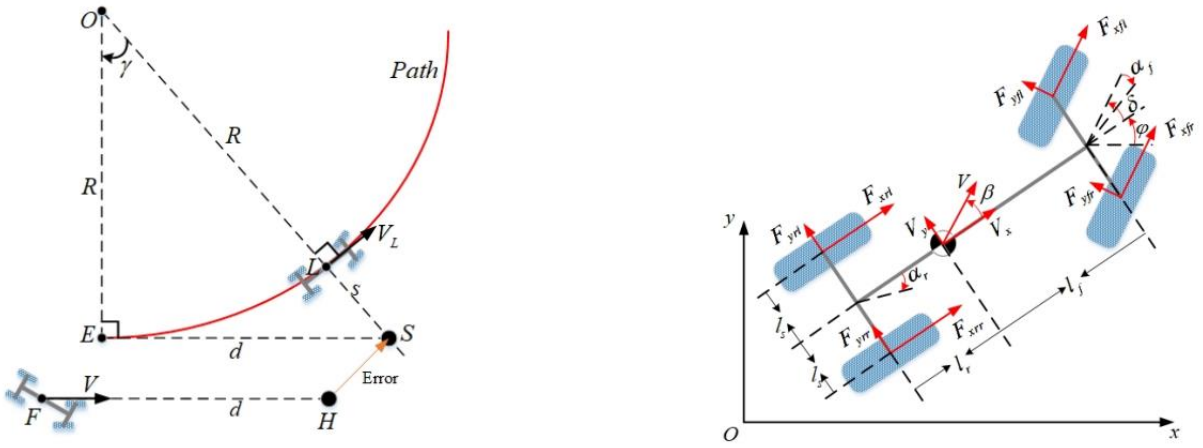

Figure 6.1 Leader and Follower; Follower Dynamics

Now, the states of the leader can be defined as $X_L = [x_L, y_L, \phi_L, V_L, \omega_L\ ]^T$. Further, the kinematics of the ideal unicycle model are used here, as

$$\dot{x}_L = V_L cos(\phi_L), \qquad \dot{y}_L = V_L sin(\phi_L) \tag{6.1}$$

$$\dot{\phi}_L = \omega_L, \qquad \dot{V}_L = a_L, \qquad \dot{\omega}_L = \Omega_L$$

Next, we can define the dynamics of the follower vehicle. Let (x, y) be the position of the center mass of the follower vehicle, and ϕ be the yaw angle. The kinematics are defined as

$$\dot{x} = V_x cos(\phi) - V_y sin(\phi) \tag{6.2a}$$

$$\dot{y} = V_x sin(\phi) + V_y cos(\phi) \tag{6.2b}$$

$$\dot{\phi} = \omega \tag{6.2c}$$

where V_x, V_y and ω are the longitudinal, lateral and yaw angular velocity of the follower vehicle. Now, the dynamics are defined as

$$\dot{V}_x = V_y \omega\ - \frac{C_a}{M} V_x^2 + \frac{2k}{M} u_1 + \frac{2k}{M} u_3$$

$$\dot{V}_y = -V_x\omega - \frac{C_f + C_r}{M}\frac{V_y}{V_x} + \frac{C_r l_r - C_f l_f}{M}\frac{\omega}{V_x} + \frac{C_f}{M}u_2 \tag{6.3}$$

$$\dot{\omega} = \frac{C_r l_r - C_f l_f}{I_z}\frac{V_y}{V_x} - \frac{C_f {l_f}^2 + C_r {l_r}^2}{I_z}\frac{\omega}{V_x} - \frac{2l_s k}{I_z}u_1 + \frac{C_f l_f}{I_z}u_2 + \frac{2l_s k}{I_z}u_3$$

where M is the vehicle mass, I_z is inertia moment, C_a is a ratio related aerodynamic drag, C_r and C_f are the tire cornering stiffness of the rear and front wheels. u_i are the driving signal inputs of various motors, where k is the gain of the motor. $u_1 = u_{fl} = u_{rl}$, $u_2 = \delta$ and $u_3 = u_{fr} = u_{rr}$. For convenience, we can write the entire dynamics as

$$\begin{bmatrix} \dot{X}_x \\ \dot{V}_y \\ \dot{\omega} \end{bmatrix} = F_1 + F_2^T a + F_3 u \tag{6.4}$$

where

$$a = \left[\frac{C_a}{M}, \frac{C_f + C_r}{M}, \frac{C_r l_r - C_f l_f}{M}, \frac{C_r l_r - C_f l_f}{I_z}, \frac{C_f {l_f}^2 + C_r {l_r}^2}{I_z}\right]^T$$

$$b = \left[\frac{k}{M}, \frac{C_f}{M}, \frac{l_s k}{I_z}, \frac{C_f l_f}{I_z}\right]^T$$

$$F_1 = \begin{bmatrix} V_y\omega \\ -V_x\omega \\ 0 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} -V_x^2 & 0 & 0 \\ 0 & -\frac{V_y}{V_x} & 0 \\ 0 & \frac{\omega}{V_x} & 0 \\ 0 & 0 & \frac{V_y}{V_x} \\ 0 & 0 & -\frac{\omega}{V_x} \end{bmatrix}$$

$$F_3 = \begin{bmatrix} \frac{2k}{M} & 0 & \frac{2k}{M} \\ 0 & \frac{C_f}{M} & 0 \\ -\frac{2l_s k}{I_z} & \frac{C_f l_f}{I_z} & \frac{2l_s k}{I_z} \end{bmatrix}$$

We consider a tracking control problem of the follower vehicle to the leader vehicle. To avoid the cutting-corner problem, the follower vehicle tracks a virtual point *S* which is the intersection between the line of forward velocity of the follower vehicle, and the line which is perpendicular to the forward velocity of the leader vehicle. If d is a constant look-ahead distance, we can define *s* as

$$s = R\frac{\sqrt{1+\kappa^2 d^2} - 1}{\kappa} \tag{6.5}$$

The angle γ is calculated as

$$\gamma = arctan(\kappa d) \tag{6.6}$$

Finally, the coordinates of the virtual point of *S* is

$$S = \begin{bmatrix} S_x \\ S_y \end{bmatrix} = \begin{bmatrix} x_L + s * sin(\phi_L) \\ y_L - s * cos(\phi_L) \end{bmatrix} \tag{6.7}$$

To properly define the follower vehicle to track the leader vehicle with a desired constant distance, we define the error variables

$$z_1 = cos(\phi + \gamma)(x_L + ssin(\phi_L) - x - dcos(\phi)$$

$$+ sin(\phi + \gamma)(y_L - scos(\phi_L) - y - dsin(\phi)$$

$$z_2 = -sin(\phi + \gamma)(x_L + ssin(\phi_L) - x - dcos(\phi)$$

$$+cos(\phi+\gamma))(y_L - scos(\phi_L) - y\ -\ dsin(\phi)$$

$$z_3 = \phi_L - \phi - \gamma \tag{6.8}$$

$$z_4 = V_L - V_x$$

$$z_5 = -V_y$$

$$z_6 = \omega_L - \omega$$

Then, we define the derivative terms as

$$\dot{z}_1 = z_2\omega + (V_L + s\omega_L)cos(z_3)\ -\ (V_L - z_4)cos(\gamma) + (z_5 - d\omega_L + dz_6)sin(\gamma)$$

$$\dot{z}_2 = -z_1\omega + (V_L + s\omega_L)sin(z_3) + (V_L - z_4)sin(\gamma) + (z_5 - d\omega_L + dz_6)cos(\gamma)$$

$$\dot{z}_3 = z_6 \tag{6.9}$$

$$\dot{z}_4 = a_L + z_5(\omega_L - z_6) + \frac{C_a}{M}\left({V_L}^2 - 2V_L z_4 + {z_4}^2\right) - \frac{2k}{M}u_1 - \frac{2k}{M}u_3$$

$$\dot{z}_5 = V_x\omega + \frac{C_f + C_r}{M}\frac{V_y}{V_x} - \frac{C_r l_r\ -\ C_f l_f}{M}\frac{\omega}{V_x} - \frac{C_f}{M}u_2$$

$$\dot{z}_6 = \Omega_L - \frac{C_r l_r\ -\ C_f l_f}{I_z}\frac{V_y}{V_x} + \frac{C_f {l_f}^2 + C_r {l_r}^2}{I_z}\frac{\omega}{V_x} + \frac{2l_s k}{I_z}u_1 - \frac{C_f l_f}{I_z}u_2 - \frac{2l_s k}{I_z}u_3$$

In compact, cascade control form, the dynamics can be rewritten as

$$\dot{z}_{\overline{13}} = f_1 + g_1 z_{\overline{46}} \tag{6.10a}$$

$$\dot{z}_{\overline{46}} = f_2 + g_2 u \tag{6.10b}$$

where we let $z_{\overline{13}} = [z_1, z_2, z_3]^T$ and $z_{\overline{46}} = [z_4, z_5, z_6]^T$.

Also, we define

$$f_1 = \begin{bmatrix} z_2\omega + (V_L + s\omega_L)cos(z_3) - V_L cos(\gamma) - d\omega_L sin(\gamma) \\ -z_1\omega + (V_L + s\omega_L)sin(z_3) + V_L sin(\gamma) - d\omega_L cos(\gamma) \\ 0 \end{bmatrix}$$

$$g_1 = \begin{bmatrix} cos(\gamma) & sin(\gamma) & dsin(\gamma) \\ -sin(\gamma) & cos(\gamma) & dcos(\gamma) \\ 0 & 0 & 1 \end{bmatrix}$$

$$f_2 = \begin{bmatrix} a_L - V_y\omega + \frac{C_a}{M}V_x^2 \\ V_x\omega + \frac{C_f + C_r}{M}\frac{V_y}{V_x} - \frac{C_r l_r - C_f l_f}{M}\frac{\omega}{V_x} \\ \Omega_L - \frac{C_r l_r - C_f l_f}{I_z}\frac{V_y}{V_x} + \frac{C_f {l_f}^2 + C_r {l_r}^2}{I_z}\frac{\omega}{V_x} \end{bmatrix}$$

$$= \begin{bmatrix} a_L \\ 0 \\ \Omega_L \end{bmatrix} - F_1 - F_2^T a$$

$$g_2 = -F_3$$

Finally, we can define the problem for this chapter, to design a controller u such that $z_{\overline{13}}$ converges to zero. They also converge to zero under assumptions regarding unknown parameters such as inertia.

6.2 Tracking Controller Design – Known Dynamics

This section will consider all known parameters for the controller design. Given the cascade structure of the error system, we once again consider a backstepping controller design procedure.

First, to ensure that z_1, z_2 and z_3 converge to zero, we choose a Lyapunov function as

$$V_1 = \frac{1}{2} z_{\overline{13}}^T z_{\overline{13}} \tag{6.11}$$

The derivative of the Lyapunov function Eq. (6.11) is

$$\dot{V}_1 = z_{\overline{13}}^T \dot{z}_{13} = z_{\overline{13}}^T (f_1 + g_1 z_{\overline{46}}) \tag{6.12}$$

Then, to make Eq. (6.12) a negative definite function, we choose a virtual control input

$$\alpha_1 = g_1^{-1}(-f_1 - K_1 z_{\overline{13}}) \tag{6.13}$$

where K_1 is a positive definite matrix. If $z_{\overline{46}} = \alpha_1$, then

$$\dot{V}_1 = -z_{\overline{13}}^T K_1 z_{\overline{13}} \leq 0.$$

However, since $z_{\overline{46}}$ is not a real control input, $z_{\overline{46}}$ does not equal to α_1. Next, we can define

$e_1 = g_1 z_{\overline{46}} - g_1 \alpha_1$, then

$$\dot{z}_{\overline{13}} = -K_1 z_{\overline{13}} + e_1 \tag{6.14}$$

$$\dot{e}_1 = \dot{g}_1 z_{\overline{46}} + g_1 f_2 + g_1 g_2 u - \frac{d(g_1 \alpha_1)}{dt} \tag{6.15}$$

Next, another Lyapunov function is chosen as

$$V_2 = V_1 + \frac{1}{2} e_1^T e_1$$

then, the derivative

$$\dot{V}_2 = -z_{\overline{13}}^T K_1 z_{\overline{13}} + z_{\overline{13}}^T e_1 + e_1^T \left(\dot{g}_1 z_{\overline{46}} + g_1 f_2 + g_1 g_2 u - \frac{d(g_1 \alpha_1)}{dt} \right).$$

Finally, the control input u is chosen as

$$u = (g_1 g_2)^{-1}\left(-\dot{g}_1 z_{\overline{46}} - g_1 f_2 - z_{\overline{13}} - K_2 e_1 + \frac{d(g_1 \alpha_1)}{dt}\right) \tag{6.16}$$

where K_2 is a positive definite constant. Then, $\dot{e}_1$ and $\dot{V}_2$ can be rewritten as

$$\dot{e}_1 = -z_{\overline{13}} - K_2 e_1$$

$$\dot{V}_2 = -z_{\overline{13}}^T K_1 z_{\overline{13}} - e_1^T K_2 e_1$$

which means that $z_{\overline{13}}$, as well as e_1 converges to zero.

Theorem 6.1: The control law Eq. (6.16) ensures that $z_{\overline{13}}$ converges to zero.

The result of Theorem 6.1 is proven through simulation. With the control law from Eq. (6.16), Fig. 6.2 shows the plot of $z_{\overline{13}}$ with respect to time. $z_{\overline{13}}(0) = [5, -5, 0]^T$. Additionally, it is also proven that e_1 converges to zero too, this is shown in Fig. 6.3.

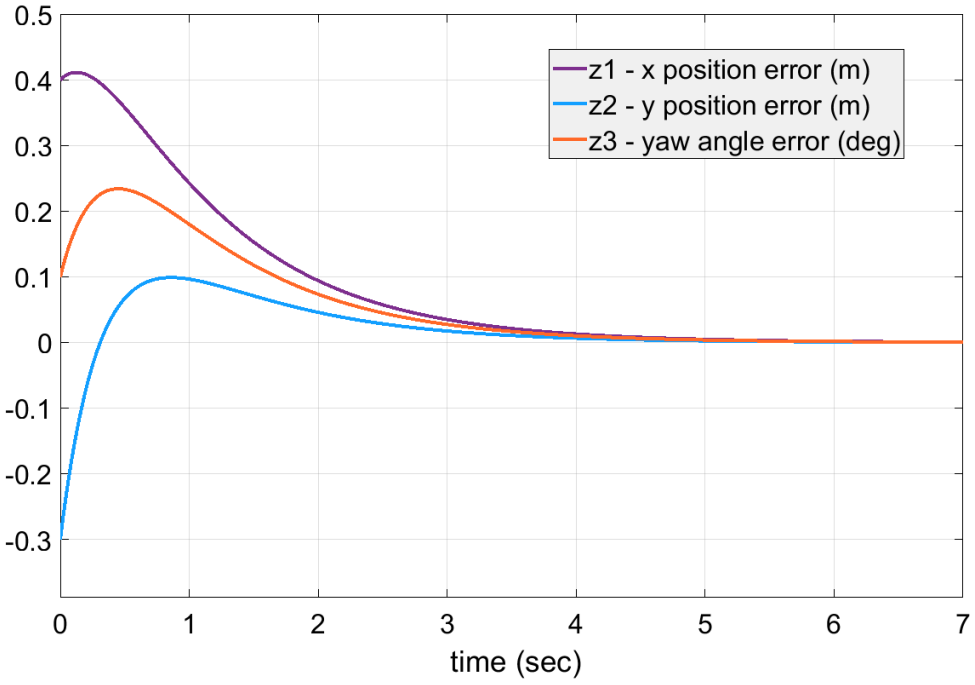

Figure 6.2 Combined Longitudinal and Lateral Controller – Known Dynamics - $z_{\overline{13}}$

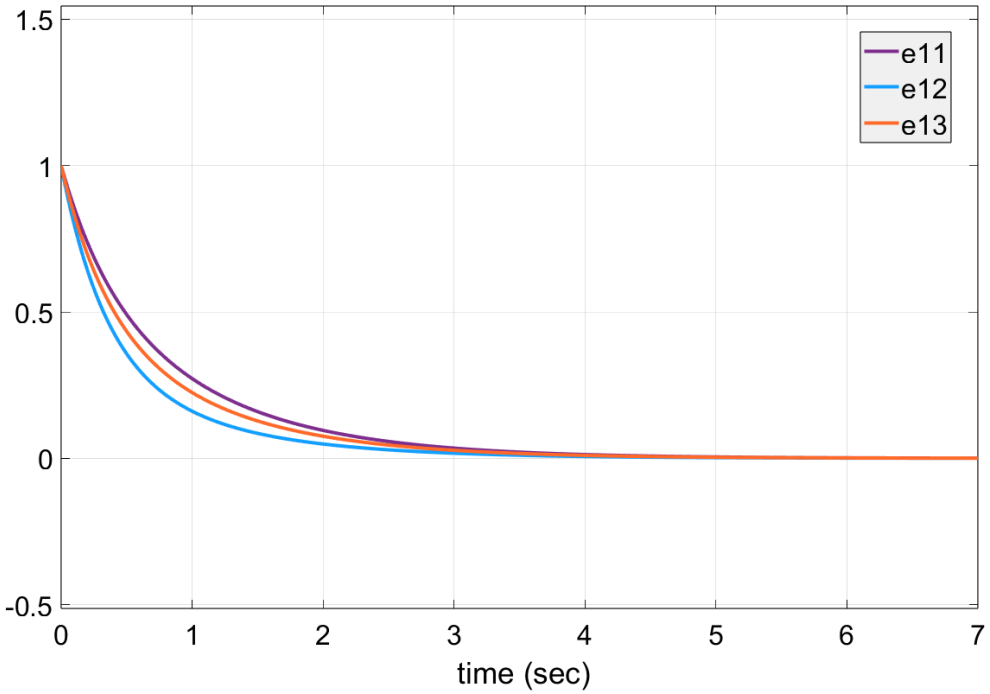

Figure 6.3 Combined Longitudinal and Lateral Controller – Known Dynamics – e1

With the above simulation results, the controller in Eq. (6.16) proves Theorem 6.1 to be satisfied. The next step is to consider uncertainties in the dynamics, specifically regarding the moment of Inertia values.

6.3 Tracking Controller Design – Unknown Parameters

If there are uncertainties regarding the inertia and/or tire stiffness parameters, the error system previously stated in Eq. (6.10) can be rewritten as

$$\dot{z}_{\overline{13}} = f_1 + g_1 z_{\overline{46}} \tag{6.16a}$$

$$\dot{z}_{\overline{46}} = f_3 + Y_1^T a + Y_2^T(u) b \tag{6.16b}$$

where

$$f_3 = \begin{bmatrix} a_L - V_y\omega \\ V_x\omega \\ \Omega_L \end{bmatrix}$$

$$Y_1 = \begin{bmatrix} V_x^2 & 0 & 0 \\ 0 & \frac{V_y}{V_x} & 0 \\ 0 & -\frac{\omega}{V_x} & 0 \\ 0 & 0 & -\frac{V_y}{V_x} \\ 0 & 0 & \frac{\omega}{V_x} \end{bmatrix} = -F_2$$

$$Y_2(u) = \begin{bmatrix} -2(u_1 + u_3) & 0 & 0 \\ 0 & -u_2 & 0 \\ 0 & 0 & 2(u_1 - u_3) \\ 0 & 0 & -u_2 \end{bmatrix}$$

First, if $z_{\overline{46}}$ is assumed to be a virtual control input. To make $z_{\overline{13}}$ converge to zero, a Lyapunov function is chosen as

$$V_1 = \frac{1}{2} z_{\overline{13}}^T z_{\overline{13}}$$

And the derivative being

$$\dot{V}_1 = z_{\overline{13}}^T \dot{z}_{13} = z_{\overline{13}}^T (f_1 + g_1 z_{\overline{46}})$$

which are both the same as the previous step. Similarly, to ensure that $\dot{V}_1$ is a negative definite function, a virtual control input is chosen as

$$\alpha_1 = g_1^{-1}(-f_1 - K_1 z_{\overline{13}})$$

where K_1 is a positive definite matrix. If $z_{\overline{46}} = \alpha_1$, then

$$\dot{V}_1 = -z_{\overline{13}}^T K_1 z_{\overline{13}} \leq 0.$$

However, since $z_{\overline{46}}$ is not a real control input, $z_{\overline{46}}$ does not equal to α_1. Next, we can define

$e_1 = g_1 z_{\overline{46}} - g_1 \alpha_1$, then

$$\dot{z}_{\overline{13}} = -K_1 z_{\overline{13}} + e_1 \tag{6.17}$$

$$\dot{e}_1 = g_1(f_3 + Y_1^T a + Y_2^T b) + \dot{g}_1 z_{\overline{46}} - \frac{d(g_1 \alpha_1)}{dt} \tag{6.18}$$

$$= g_1 f_3 + g_1 Y_1^T \hat{a} + g_1 g_2(\hat{b}) u + \dot{g}_1 z_{\overline{46}} - \frac{d(g_1 \alpha_1)}{dt} + g_1 Y_1^T (a - \hat{a}) + g_1 Y_2^T (b - \hat{b})$$

Another Lyapunov function is chosen as

$$V_2 = V_1 + \frac{1}{2}e_1^T e_1 + \frac{1}{2}\tilde{a}^T \Gamma_1^{-1}\tilde{a} + \frac{1}{2}\tilde{b}^T \Gamma_2^{-1}\tilde{b}$$

Then, the derivative is

$$\dot{V}_2 = -z_{\overline{13}}^T K_1 z_{\overline{13}} + z_{\overline{13}}^T e_1 + e_1^T \left(g_1 f_3 + g_1 Y_1^T \hat{a} + g_1 g_2(\hat{b}) u + g_1 Y_1^T (a - \hat{a}) + g_1 Y_2^T (b - \hat{b}) + \dot{g}_1 z_{\overline{46}} - \frac{d(g_1 \alpha_1)}{dt} \right) + \tilde{a}^T \Gamma_1^{-1} \dot{\tilde{a}} + \tilde{b}^T \Gamma_2^{-1} \dot{\tilde{b}}$$

Finally, the control input u, as well as the adaptive laws of $\hat{a}$ and $\hat{b}$ as chosen as

$$u = \left(g_1 g_2(\hat{b}) \right)^{-1} \left(-g_1 f_3 + \frac{d(g_1 \alpha_1)}{dt} - \dot{g}_1 z_{\overline{46}} - g_1 Y_1^T \hat{a} - z_{\overline{13}} - K_2 e_1 \right) \tag{6.19}$$

$$\dot{\hat{a}} = \Gamma_1 Y_1 g_1^{-1} e_1 \tag{6.20a}$$

$$\dot{\hat{b}} = \Gamma_2 Y_2 g_1^{-1} e_1 \tag{6.20b}$$

where K_2 is a positive definite constant. Then,

$$\dot{V}_2 = -z_{\overline{13}}^T K_1 z_{\overline{13}} - e_1^T K_2 e_1$$

which means that $z_{\overline{13}}$ and e_1 converge to zero, as well as bounding both $\hat{a}$ and $\hat{b}$.

Again, the results from this section are summarized in the following theorem below.

Theorem 6.2: The control law Eq. (6.19) and the adaptive laws Eq. (6.20) ensure that $z_{\overline{13}}$ and e_1 converge to zero, as well as bounding both $\hat{a}$ and $\hat{b}$.

Additionally, in the control law design, $\hat{b}$ should be chosen such that $g_2(\hat{b})$ is nonsingular. It can be shown that $g_2(\hat{b})$ is nonsingular if each element of $\hat{b}$ is positive. To this end, the project operator may be introduced in the adaptive law of $\hat{b}$. To make all elements of $\hat{b}$ positive, the adaptive law of $\hat{b}$ can be changed as follows

$$\dot{\hat{b}} = Proj_{\Omega}(\Gamma_2 Y_2 g_1^{-1} e_1)$$

$$\dot{\hat{b}}_j = \begin{cases} \Gamma_2 Y_2 g_1^{-1} e_1, & if\ \hat{b}_j > 0\ or\ \hat{b}_j = 0\ and\ Y_{j2} g_1^{-1} e_1 > 0 \\ 0, & otherwise \end{cases}$$

where $\Omega = R^+$and Y_{j2} denotes the j-th row of Y_2.

Finally, the plot of $z_{\overline{13}}$ is shown in Fig. 6.4, which considers the unknown parameters of vector *a* and vector *b*. As mentioned previously, the states do in fact converge to zero.

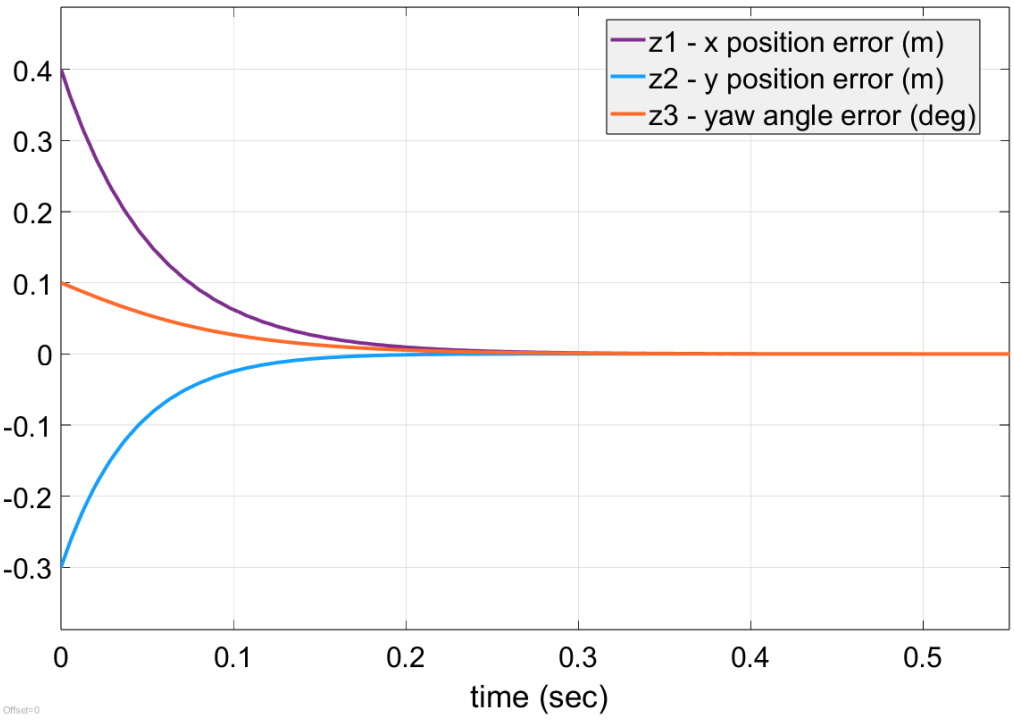

Figure 6.4 Combined Longitudinal and Lateral Controller – Unknown Dynamics - $z_{\overline{13}}$

Also, in Fig. 6.5, e_1 is also shown to converge to zero, although not as smoothly as with the known dynamic model, which is to be expected.

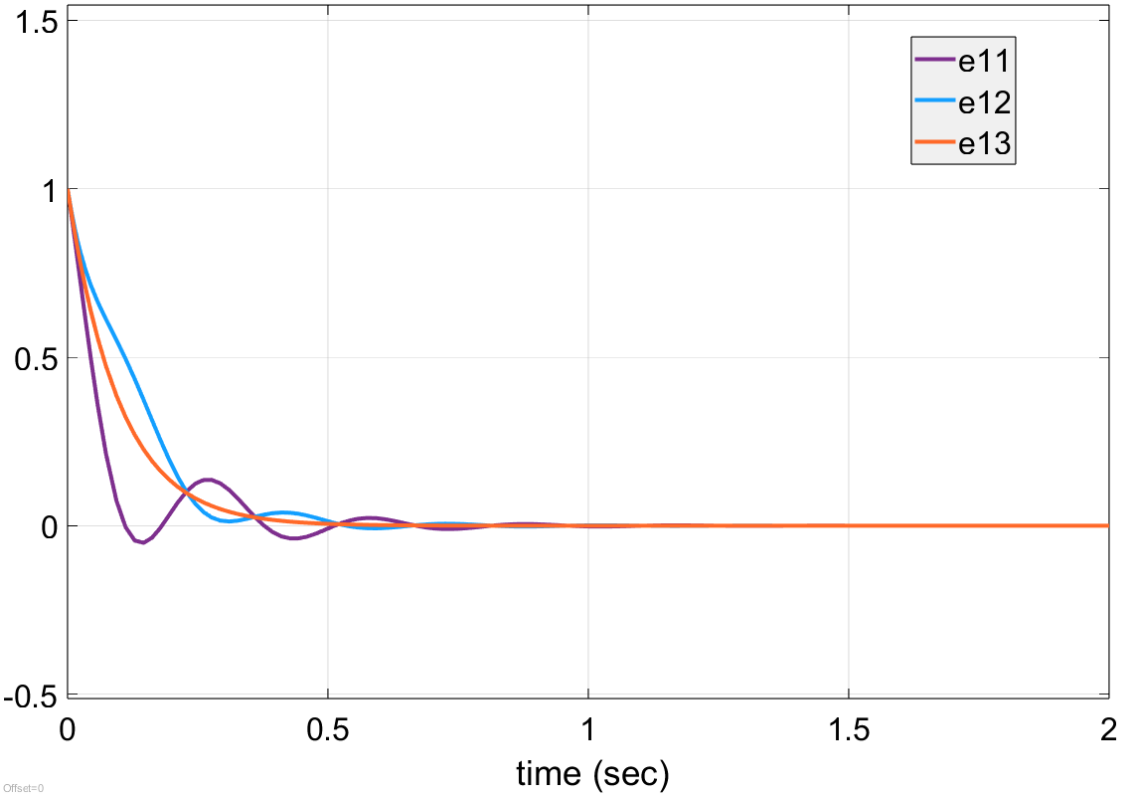

Figure 6.5 Combined Longitudinal and Lateral Controller – Unknown Dynamics - e_1

With the above simulation results, Theorem 6.2 is proven to hold true for this study. The final consideration for this work is to consider a safety boundary within the controller design.

6.4 Tracking Controller Design with Unknown Parameters and Safety Consideration

Finally, this work will look to design a controller with unknown parameters, considering a safety boundary for z_1 and z_2. The same cascade structure from Eq. (6.16) in the previous subsection is considered, along with the previously defined vectors and matrices.

First, it is assumed that $z_{\overline{46}}$ is a virtual control input. To make z_1 and z_2 converge to zero, as well as enforcing $|z_1| < c_1$ and $|z_2| < c_2$, we choose a Lyapunov function

$$V_1 = \frac{1}{2} log \frac{c_1^2}{c_1^2 - z_1^2} + \frac{1}{2} log \frac{c_2^2}{c_2^2 - z_2^2} + \frac{1}{2} z_3^2.$$

Then, the derivative of V_1 is

$$\dot{V}_1 = \frac{z_1 \dot{z}_1}{c_1^2 - z_1^2} + \frac{z_2 \dot{z}_2}{c_2^2 - z_2^2} + z_3 \dot{z}_3$$

$$= \left[\frac{z_1}{c_1^2 - z_1^2}, \frac{z_2}{c_2^2 - z_2^2}, z_3 \right] \dot{z}_{13}$$

$$= \left[\frac{z_1}{c_1^2 - z_1^2}, \frac{z_2}{c_2^2 - z_2^2}, z_3 \right] (f_1 + g_1 z_{\overline{46}}).$$

To make $\dot{V}_1$ a negative definite function, we choose a virtual control input as

$$\alpha_1 = g_1^{-1}(-f_1 - K_1 diag([c_1^2 - z_1^2, c_2^2 - z_2^2, 1]) z_{\overline{13}}) \tag{6.21}$$

where K_1 is a positive definite matrix. If $z_{\overline{46}} = \alpha_1$, then

$$\dot{V}_1 = -z_{\overline{13}}^T K_1 z_{\overline{13}} \leq 0.$$

The next step is to consider $z_{\overline{46}}$ as a not real control input, therefore $z_{\overline{46}}$ does not equal α_1. Let $e_1 = g_1 z_{\overline{46}} - g_1 \alpha_1$, then

$$\dot{z}_{\overline{13}} = -K_1 diag([c_1^2 - z_1^2, c_2^2 - z_2^2, 1]) z_{\overline{13}} + e_1 \tag{6.22}$$

$$\dot{e}_1 = g_1(f_3 + Y_1^T a + Y_2^T b) + \dot{g}_1 z_{\overline{46}} - \frac{d(g_1\alpha_1)}{dt} \tag{6.23}$$

$$= g_1 f_3 + g_1 Y_1^T \hat{a} + g_1 g_2(\hat{b}) u + \dot{g}_1 z_{\overline{46}} - \frac{d(g_1\alpha_1)}{dt} + g_1 Y_1^T (a - \hat{a}) + g_1 Y_2^T (b - \hat{b}).$$

Another Lyapunov function is chosen as

$$V_2 = V_1 + \frac{1}{2} e_1^T e_1 + \frac{1}{2} \tilde{a}^T \Gamma_1^{-1} \tilde{a} + \frac{1}{2} \tilde{b}^T \Gamma_2^{-1} \tilde{b}$$

then its derivative is

$$\dot{V}_2 = -z_{\overline{13}}^T K_1 z_{\overline{13}} + \left[\frac{z_1}{c_1^2 - z_1^2}, \frac{z_2}{c_2^2 - z_2^2}, z_3 \right] e_1$$

$$+ e_1^T \Big(g_1 f_3 + g_1 Y_1^T \hat{a} + g_1 g_2(\hat{b}) u + g_1 Y_1^T (a - \hat{a}) + g_1 Y_2^T (b - \hat{b}) + \dot{g}_1 z_{\overline{46}}$$

$$- \frac{d(g_1\alpha_1)}{dt} \Big) + \tilde{a}^T \Gamma_1^{-1} \dot{\tilde{a}} + \tilde{b}^T \Gamma_2^{-1} \dot{\tilde{b}}.$$

Finally, the control input u and the adaptive laws of $\hat{a}$ and $\hat{b}$ as chosen as

$$u = \left(g_1 g_2(\hat{b}) \right)^{-1} \left(-g_1 f_3 + \frac{d(g_1\alpha_1)}{dt} - \dot{g}_1 z_{\overline{46}} - g_1 Y_1^T \hat{a} - \left[\frac{z_1}{c_1^2 - z_1^2}, \frac{z_2}{c_2^2 - z_2^2}, z_3 \right]^T - K_2 e_1 \right) \tag{6.24}$$

$$\dot{\hat{a}} = \Gamma_1 Y_1 g_1^{-1} e_1 \tag{6.25a}$$

$$\dot{\hat{b}} = \Gamma_2 Y_2 g_1^{-1} e_1 \tag{6.25b}$$

where K_2 is a positive definite constant. Then,

$$\dot{V}_2 = -z_{\overline{13}}^T K_1 z_{\overline{13}} - e_1^T K_2 e_1$$

which means that $z_{\overline{13}}$ and e_1 converge to zero, as well as bounding both $\hat{a}$ and $\hat{b}$.

Like before, the theory of this subsection is summarized in Theorem 6.3.

Theorem 6.3: The control law Eq. (6.24) and the adaptive laws Eq. (6.25) ensure that $z_{\overline{13}}$ and e_1 converge to zero, as well as bounding both $\hat{a}$ and $\hat{b}$.

Additionally, in the control law design, $\hat{b}$ should be chosen such that $g_2(\hat{b})$ is nonsingular. It can be shown that $g_2(\hat{b})$ is nonsingular if each element of $\hat{b}$ is positive. To this end, the project operator may be introduced in the adaptive law of $\hat{b}$. To make all elements of $\hat{b}$ positive, the adaptive law of $\hat{b}$ can be changed as follows

$$\dot{\hat{b}} = Proj_{\Omega}(\Gamma_2 Y_2 g_1^{-1} e_1)$$

$$\dot{\hat{b}}_j = \begin{cases} \Gamma_2 Y_2 g_1^{-1} e_1, & if\ \hat{b}_j > 0\ or\ \hat{b}_j = 0\ and\ Y_{j2} g_1^{-1} e_1 > 0 \\ 0, & otherwise \end{cases}$$

where $\Omega = R^+$and Y_{j2} denotes the j-th row of Y_2.

At this time, tuning to the gamma and gain parameters are needed to prove Theorem 6.3 to hold true. Future work will include verifying the theorem for this application.

CHAPTER VII

CONCLUSION

This thesis considered the integrator backstepping technique to guarantee safety for both lateral control problems, as well as combined longitudinal and lateral control problems. Additionally, this control technique can be designed to consider uncertainties in the dynamics and its parameters. The controller design ensures that our desired states converge to either a chosen value, or to zero. Simulations were conducted in MATLAB/Simulink to prove our expected results.

Two different autonomous vehicle applications were considered for this work, a lane keeping problem, in which we track the lateral distance, lateral velocity, yaw angle and yaw rate. Ideally, the lateral distance and lateral velocity converge to zero, ensuring that the vehicle is centered within a lane. Then, a combined longitudinal and lateral control problem was considered, not only a lane keeping problem, but a trajectory tracking problem of a leader and follower vehicle. Both problems assuming a road with curvature, not a straight-line path.

Each control problem's controller design considers the integrator backstepping technique, both for fully known dynamics and unknown dynamics. From this point, future work can be to consider additional update laws for the dynamics, or to utilize a different technique for the controller design such as reinforcement learning.

REFERENCES

[1] Rajamani. (2012). Vehicle Dynamics and Control (2nd ed. 2012.). Springer US. https://doi.org/10.1007/978-1-4614-1433-9

[2] Lamport. (1977). Proving the Correctness of Multiprocess Programs. IEEE Transactions on Software Engineering, SE-3(2), 125–143. https://doi.org/10.1109/TSE.1977.229904

[3] Ames, Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control Barrier Functions: Theory and Applications. 2019 18th European Control Conference (ECC), 3420–3431. https://doi.org/10.23919/ECC.2019.8796030

[4] Ames, Grizzle, J. W., & Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. 2014 IEEE 53RD ANNUAL CONFERENCE ON DECISION AND CONTROL (CDC), 6271–6278. https://doi.org/10.1109/CDC.2014.7040372

[5] Zeng, Zhang, B., Li, Z., & Sreenath, K. (2021). Safety-Critical Control using Optimal-decay Control Barrier Function with Guaranteed Point-wise Feasibility.

[6] He, Zeng, J., Zhang, B., & Sreenath, K. (2021). Rule-Based Safety-Critical Control Design using Control Barrier Functions with Application to Autonomous Lane Change.

[7] Coen, Anthonis, J., & De Baerdemaeker, J. (2008). Cruise control using model predictive control with constraints. Computers and Electronics in Agriculture, 63(2), 227–236. https://doi.org/10.1016/j.compag.2008.03.003

[8] Naus, Ploeg, J., Van de Molengraft, M. J. G., Heemels, W. P. M. H., & Steinbuch, M. (2010). Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach. Control Engineering Practice, 18(8), 882–892. https://doi.org/10.1016/j.conengprac.2010.03.012

[9] Zeng, Zhang, B., & Sreenath, K. (2020). Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function.

[10] Ma, Chu, L., Guo, J., Wang, J., & Guo, C. (2020). Cooperative Adaptive Cruise Control Strategy Optimization for Electric Vehicles Based on SA-PSO With Model Predictive Control. *IEEE Access*, *8*, 225745–225756. https://doi.org/10.1109/ACCESS.2020.3043370

[11] Abbas, Milman, R., & Eklund, J. M. (2017). Obstacle Avoidance in Real Time With Nonlinear Model Predictive Control of Autonomous Vehicles. Canadian Journal of Electrical and Computer Engineering, 40(1), 12–22. https://doi.org/10.1109/CJECE.2016.2609803

[12] Grandia, Taylor, A. J., Singletary, A., Hutter, M., & Ames, A. D. (2020). Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions. arXiv.org. https://doi.org/10.15607/RSS.2020.XVI.098

[13] Kamalapurkar, R., Walters, P., Rosenfeld, J., & Dixon, W. (2018, May 28). Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach (Communications and Control Engineering) (1st ed. 2018). Springer.

[14] *Reinforcement Learning Toolbox*. (n.d.). MATLAB. Retrieved October 4, 2022, from https://www.mathworks.com/products/reinforcement-learning.html

[15] Sutton, & Barto, A. (2018). Reinforcement Learning: An Introduction, 2nd Edition (pp. 1–526). Mit Press.

[16] García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res., 16, 1437-1480.

[17] Coraluppi, & Marcus, S. . (2000). Mixed risk-neutral/minimax control of discrete-time, finite-state Markov decision processes. IEEE Transactions on Automatic Control, 45(3), 528–532. https://doi.org/10.1109/9.847737

[18] Sato, M., Kimura, H., & Kobayashi, S. (2001). TD algorithm for the variance of return and mean-variance reinforcement learning. Transactions of the Japanese Society for Artificial Intelligence, 16(3), 353-362. https://doi.org/10.1527/tjsai.16.353

[19] Abbeel, Coates, A., & Ng, A. Y. (2010). Autonomous Helicopter Aerobatics through Apprenticeship Learning. The International Journal of Robotics Research, 29(13), 1608–1639. https://doi.org/10.1177/0278364910371999

[20] Koppejan, & Whiteson, S. (2011). Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. Evolutionary Intelligence, 4(4), 219–241.

[21] Ogata. (2010). Modern control engineering (5th ed.). Prentice Hall.

[22] Sontag, E. (1983). A Lyapunov-Like Characterization of Asymptotic Controllability. SIAM Journal on Control and Optimization, 21, 462-471.

[23] Sontag. (1989). A “universal” construction of Artstein’s theorem on nonlinear stabilization. Systems & Control Letters, 13(2), 117–123. https://doi.org/10.1016/0167-6911(89)90028-5

[24] Artstein, Z. (1983). Stabilization with relaxed controls. Nonlinear Analysis-theory Methods & Applications, 7, 1163-1173.

[25] Ames, Galloway, K., Sreenath, K., & Grizzle, J. W. (2014). Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics. IEEE Transactions on Automatic Control, 59(4), 876–891. https://doi.org/10.1109/TAC.2014.2299335

[26] Khalil, H. (2014, February 6). Nonlinear Control (1st ed.). Pearson.

[27] Ames, & Powell, M. (n.d.). Towards the Unification of Locomotion and Manipulation through Control Lyapunov Functions and Quadratic Programs. In Control of Cyber-Physical Systems (pp. 219–240). Springer International Publishing. https://doi.org/10.1007/978-3-319-01159-2_12

[28] Galloway, Sreenath, K., Ames, A. D., & Grizzle, J. W. (2015). Torque Saturation in Bipedal Robotic Walking Through Control Lyapunov Function-Based Quadratic Programs. IEEE Access, 3, 323–332. https://doi.org/10.1109/ACCESS.2015.2419630

BIOGRAPHICAL SKETCH

Randy Guadalupe Chapa began his M.S.E. in Electrical Engineering at the University of Texas Rio Grande Valley in January 2021. Before joining UTRGV, he obtained his B.S. in Computer Engineering Technology from the University of Houston, located in Houston, TX, back in May 2020. During his tenure at UTRGV, Randy worked as a student project associate for the Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (CREST) under Dr. Wenjie Dong, where he researched safety guarantees of autonomous systems. In the Summer months, Randy worked as a software engineering intern for prestigious companies like Jacobs Engineering Group and DISH Network. He was awarded the Master of Science in Engineering degree from UTRGV in December 2022. After graduation, Randy joined CapitalOne as a technology development associate, to continue his development as a professional, while exploring his interests in finances and data analytics.

To contact him for any information, please reach out to randygchapa@gmail.com