UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Lane-keeping and Adaptive Speed Control for Robotic Systems

Permalink

https://escholarship.org/uc/item/1691708p

Author

Qian, Cheng

Publication Date

2020

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA SAN DIEGO

Lane-keeping and Adaptive Speed Control for Robotic Systems

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Engineering Science (Mechanical Engineering)

by

Cheng Qian

Committee in charge:

Professor Mauricio de Oliveira, Chair
Professor Jorge Cortes, Co-Chair
Professor Thomas Bewley

2020

Copyright
Cheng Qian, 2020
All rights reserved.

The thesis of Cheng Qian is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____
Co-Chair

_____
Chair

University of California San Diego

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my advisor, Professor Mauricio de Oliveira for his continuous guidance through my graduate research. He always showed great patience and gave insightful instructions whenever I ran into trouble or had questions about my research. He is not only an inspirational professor for my research work but also a great mentor for my life.

I would also like to thank professor Jorge Cortes and professor Thomas Bewley for sparing time from their busy schedule and attending my thesis presentation. I have gained a lot of experience and knowledge from their insightful lectures.

At last but not the least, I would like to thank my family for supporting me in higher education. Without their countless love and continuous encouragement, the thesis could not have been successfully conducted.

ABSTRACT OF THE THESIS

Lane-keeping and Adaptive Speed Control for Robotic Systems

by

Cheng Qian

Master of Science in Engineering Science (Mechanical Engineering)

University of California San Diego, 2020

Professor Mauricio de Oliveira, Chair
Professor Jorge Cortes, Co-Chair

Control Lyapunov functions and barrier functions have been successfully applied to control the motion of robotic systems as a means to ensure safety and performance. This study aims at constructing a new approach to the problem of simultaneous lane-keeping and adaptive speed control problem of a robotic system, and a modified Lyapunov function and barrier setup are also proposed that fits this problem more naturally than the ones available in the main literature.

Under the new setup, it is possible to offer guarantees of convergence as well as to explicitly calculate the solution of the associated quadratic program used to determine the control

input. At last, a few examples are provided to illustrate the results, showing how different control strategies affect the performance of the robot system.

Chapter 1

Introduction

1.1 Motivation

The techniques of autonomous vehicles have been increasingly developed over the past few years. Autonomous vehicles, especially indoor/outdoor robots and self-driving cars have brought significant changes to our life. Although negative news about autonomous vehicles is there all the time, people still believe that autonomous vehicles have more potential benefits than disadvantages. Due to the development of higher-accuracy sensors, machine learning techniques and motion planning algorithms, autonomous vehicles are able to accomplish various tasks. However, due to the uncertainty of the environment, autonomous vehicles still cannot eliminate all dangers which might cause serious consequences. To make autonomous vehicles more reliable and present more benefits for our lives, people are looking for ways to overcome the safety challenges.

For self-driving cars, most of the accidents are due to human error. Therefore, Advanced Driver-Assistance Systems (ADAS) are developed and implemented to help minimize human error and improve the safety of autonomous vehicle systems. In ADAS, there are two main tasks to achieve: lane-keeping and adaptive speed control, which are the key to the safety and

performance of autonomous vehicle systems.

- **Lane-keeping:** Controlling the throttle and steering of the vehicle according to a prescribed path and stay within the boundaries of a prescribed lane around the desired path.
- **Adaptive Speed Control:** Allowing the vehicle to drive at a prescribed velocity whenever possible and reducing the speed if necessary to maintain a safe distance from other vehicles.

Even though ADAS has been widely equipped on modern passenger vehicles, people should never slow down the pace of developing higher performance algorithms in terms of safety.

1.2 Previous Works

Myriad methods have been reported in the literature that can perform the lane-keeping and adaptive speed control tasks, e.g. [3, 11, 10]. In the following sections, the Lyapunov and barrier certificates based approach will be introduced. In this approach, control inputs are calculated by solving convex Quadratic Programs (QP) [4].

Some relatively new methods that blend Control Lyapunov functions (CLF), for convergence to one or multiple control objectives, and Control Barrier Functions (CBF), for meeting safety requirements, e.g. [4, 3, 1] are first revisited. Such methodology provides a framework within which one can deal with safety-critical tasks that might involve potentially conflicting control objectives and safety constraints [3].

The idea behind the control Lyapunov functions is to first find an appropriate Lyapunov function to stabilize the nonlinear robotic system in terms of some specific control objectives, and then by ensuring $\dot{V}(x,u) \leq 0$, optimal controls can be generated through constrained optimization problem.

For barrier functions, [4] introduces two types of barrier functions, which are the Reciprocal Barrier Function (RBF) and Zeroing Barrier Function (ZBF). The main difference between

these two barrier functions is that, for ZBF, the barrier function is defined as:

$$B = h(x)$$

which the value of B will vanish as the state x approaches the boundary of a given set $\mathcal{C}$.

For RBF, the barrier function is defined as:

$$B = \frac{1}{h(x)}$$

which B will go to infinity as the state x approaches the boundary. Similar to CLFs, the idea behind the use of CBFs is that the safety requirements can also be encoded into inequality constraints via $\dot{B}(x,u) \geq 0$ for generating optimal controls.

In this CLF-CBF framework, the control Lyapunov function acts as a tool to stabilize the system around the desired control objectives, whereas the control barrier function is used to enforce the trajectory will remain within prescribed safety boundaries.

The combination of control Lyapunov and control barrier function is often done in an ad hoc way, guarantee the system to achieve the control objective which, at the same time, satisfy the safety requirements.

The CLF-CBF approach, enabled by the availability of efficient numerical QP solvers, has been applied to several real-world robotic problems such as bipedal robot walking [7, 2], automatic cruise control [10], and lane-keeping [12], collision-free multi-agent system and obstacle avoidance [6].

1.3 Contributions

Inspired by previous works, this research will mainly focus on the improvement of the CLF-CBF-QP approach towards the simultaneous lane-keeping and adaptive speed control

problem.

In previous works, a CLF-CBF-QP framework is used to deal with conflicting control objectives. To ensure that QP is feasible, relaxation parameters are considered in the conflicting CLF constraint. However, this approach can never offer guarantees on the asymptotical stability of the robotic system which will be discussed later.

The starting point in this study is the simultaneous adaptive speed regulation and lane-keeping CLF-CBF-QP approach proposed in [12]. In that paper, the authors propose an ad hoc combination of three CLFs and two CBFs using a combination of slack variables and penalties, treating the inequalities arising from the need to reduce the CLF as *soft constraints*, to keep the adherence of CBF based safety constraints for lane-keeping and adaptive speed control.

The drawback of this approach will weaken the performance of lane-keeping and adaptive speed control, in other words, the control objectives will never converge to zero under this implementation. In practice, one will always want the control objectives to be achieved as much as possible while the safety specifications are not violated.

In this study, the simultaneous adaptive speed regulation and lane-keeping problem is reconstructed by proposing the use of a single CLF that is capable of simultaneously ensuring convergence to a target trajectory and adaptive speed control, and a single CBF for lane-keeping. With this modification, the relaxation parameters are no longer necessary in this problem since the control objectives have become compatible, and it is possible to asymptotically stabilize the robotic system. The resulting QP requires no artificial penalties as well.

A Karush-Kuhn-Tucker (KKT) condition based constrained optimization problem is solved to get the optimal controller. The advantage of the simpler problem is utilized to derive analytic formulas for the calculation of control inputs that do not require the use of an online solver. Simulation examples are also provided to illustrate the corresponding results.

1.4 Thesis Organization

The structure of this thesis is as follows. Chapter 2 briefly reviews the CLF-CBF-QP approach of [12] including the robot kinematic model, the CLF and CBF constraints, and the prescribed trajectory. Chapter 3 introduces the formal definitions of the Exponentially Stabilizing Control Lyapunov Function (ES-CLF) and Zeroing Control Barrier Function (ZCBF) that are used in this study. The discussion on the drawbacks of this approach is also provided in this chapter. In Chapter 4, the new CLF and CBF is proposed, and the discussion on the non-smooth analysis regarding to the new CLF is also presented. Chapter 5 calculates explicit solution to the resulting QP. Finally, Chapter 6 presents two simulation examples that illustrate the proposed methods, followed by brief Conclusions in Chapter 7.

Chapter 2

Modeling

Before moving forward to the simultaneous lane-keeping and adaptive speed control problem, an approximate model must be selected first to represent the physical robotic system. One commonly used robotic model is the standard unicycle model.

2.1 State Space Representation

In this study, a differential-drive ground robotic system which contains one leading robot and one following robot is considered. Each of the robots has two independent driving wheels. Our goal is to steer the robot by controlling the left and right wheel's angular velocity to achieve the control objectives. The original kinematic model of this robotic system is defined as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{R}{2}(\omega_r + \omega_l)\cos(\theta) \\ \frac{R}{2}(\omega_r + \omega_l)\sin(\theta) \\ \frac{R}{L}(\omega_r - \omega_l) \end{pmatrix} \tag{2.1}$$

where

R	radius of wheels
ω_r	angular velocity of right wheel
ω_l	angular velocity of right wheel
L	length of wheel axis
(x,y)	robot position
θ	robot orientation

According to previous works, a standard unicycle model is often used to simplify this two-wheeled robot model without any loss. Namely, by taking the mapping:

$$v = \frac{R}{2}(\omega_r + \omega_l)$$
$$\omega = \frac{R}{L}(\omega_r - \omega_l)$$

the control inputs become the linear velocity v and the angular velocity ω of the robot, and the kinematics of the standard unicycle model is given by:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{pmatrix} \tag{2.2}$$

Additionally, in practice, it is more natural in taking the longitudinal force and torque as inputs, as in [12], [9]. To coincide with these control inputs, a constant $a > 0$ is used to locate (x,y) ahead of the wheel axis, as shown in Figure 2.1.

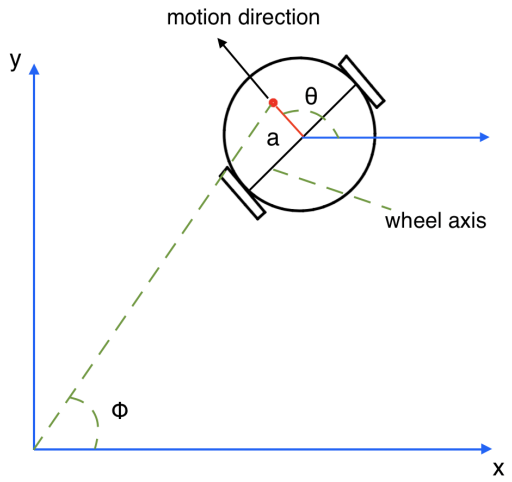


Figure 2.1: Robot state and a distance ahead of the wheel axis

The corresponding dynamics is given by:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{v} \\ \dot{\theta} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} v\cos(\theta) - a\,\omega\sin(\theta) \\ v\sin(\theta) + a\,\omega\cos(\theta) \\ u_l/m - a\omega^2 \\ \omega \\ u_a/I_z \end{pmatrix} = f(\mathbf{x}) + g(\mathbf{x})\,\mathbf{u} \tag{2.3}$$

where

u_l	tangential force input
u_a	angular torque input
I_z	moment of inertia about the z-axis
m	mass of the robot
v	robot tangential velocity
ω	robot angular velocity
a	a distance ahead the wheel axis

and the functions f and g can be calculated from the model above, which are:

$$f(\mathbf{x}) = \begin{pmatrix} v\cos(\theta) - a\omega\sin(\theta) \\ v\sin(\theta) + a\omega\cos(\theta) \\ -a\omega^2 \\ \omega \\ 0 \end{pmatrix}$$

$$g(\mathbf{x}) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 0 \\ 0 & 1/I_z \end{pmatrix}$$

The state $\mathbf{x}$ of the robot and the input $\mathbf{u}$ are defined as:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ v \\ \theta \\ \omega \end{pmatrix} \qquad \mathbf{u} = \begin{pmatrix} u_l \\ u_a \end{pmatrix}$$

2.2 Lane-keeping and Adaptive Speed Control Constraints

The goals of Lane-Keeping (LK) problem are to keep the robot driving on the desired path which is considered as a ”soft” constraint, and to avoid driving across the inner and outer boundaries of the lane, which is considered as a ”hard” constraint. Similarly, the ”soft” constraint for Adaptive Speed Control (ASC) is to make the robot drive at the desired speed while the ”hard”

constraint is to keep a safe distance with the leading robot.

2.2.1 ASC Control Constraints

The hard constraint for the ASC problem ensures avoiding collision between the leading robot and the following robot. Hard constraints always have the highest priority, which means they should not be violated at all times. There are numerous ways to represent this constraint, as in [4], the rule for ASC can be stated as:

$$D \geq \tau v \tag{2.4}$$

where

D	safe distance between the leading and following robot
v	forward velocity of the following robot
τ	time-scaled parameter

and more specifically, τ is a pre-defined parameter that defines the time for safe distance reducing to zero under current velocity.

The soft constraint can be stated as when the distance between the two robots is ”safe”, the controlled robot should drive at the desired speed. Namely:

$$v \rightarrow v_d \tag{2.5}$$

where

v	forward velocity of the following robot
v_d	desired forward velocity

2.2.2 LK Control Constraints

The hard constraint for LK problem can be considered as the lateral displacement y of the robot from the center of the path should not exceed a pre-defined constant d_{max} which represents the inner and outer edge of the path, namely:

$$| y | \leq d_{max} \tag{2.6}$$

To encode this rule to a more general pattern, the techniques in [4] is used here. Consider at time zero, the lateral displacement is $y(0)$ and the lateral velocity is $\dot{y}(0)$. Assume that under the maximum allowable acceleration a_{max}, it takes time T for the robot to reduce the lateral velocity to zero, this can be formed by an equation:

$$y(T) = y(0) + T\dot{y}(0) - \frac{sign(\dot{y}(0))}{2} T^2 a_{max}$$

Take $T = \frac{|\dot{y}(0)|}{a_{max}}$ the above equation becomes:

$$y(T) = y(0) + \frac{|\dot{y}(0)|}{2a_{max}} \dot{y}(0)$$

together with (2.6), the following inequality is generated to represent the hard constraint in the same manner:

$$sign(\dot{y})y + \frac{\dot{y}^2}{2a_{max}} \leq d_{max} \tag{2.7}$$

There are two soft constraints in the LK problem, one is the position of the controlled robot converges to the center of the desired path:

$$(x, y) \rightarrow (x_d, y_d) \tag{2.8}$$

and the angular velocity converges to the desired angular velocity to smooth the trajectory:

$$\omega \to \omega_d \tag{2.9}$$

Note that in [12], the author set $\omega_d = 0$ in order to make the trajectory much more smoother. With a relaxation parameter δ take into consideration, this choice is easy to implement. However, it cannot guarantee asymptotically stable of the system. See the upcoming chapters for further discussion.

2.3 Target Trajectory

Our goal is to steer the robot as close as possible to a prescribed path, R_{path}, which is assumed here to be given in the polar form, that is, $R_{\text{path}}(\phi)$ is a radial distance and ϕ is an angle as shown in Figure 2.1. The particular trajectory can be defined as:

$$R_{\text{path}}(\phi) = R + b\sin(n\phi) \tag{2.10}$$

with parameters, n and R, given in Table 6.1 will be used in the illustrative examples later. One should also note that ϕ is a function of (x,y).

With the above target trajectory, x_d and y_d in (2.8) can be represented as:

$$x_d = R_{path}(\phi)\cos\phi$$
$$y_d = R_{path}(\phi)\sin\phi$$

and the control objective (2.8) can be rewrite as $\eta \to 0$ where the output η is defined given by:

$$\eta = \begin{pmatrix} x - R_{path}(\phi)\cos\phi \\ y - R_{path}(\phi)\sin\phi \end{pmatrix}$$

Chapter 3

Combined CLF-CBF Control

This chapter mainly focuses on the CLF-CBF-QP framework based control method. The process that the controller is constructed under this method is shown in detail. The pros and cons of this approach have also been discussed, and a modified version to overcome these drawbacks will be presented in the next chapter.

3.1 Quadratic Program based Safe Control Design

In this section, the definitions and lemmas related to the combined control Lyapunov and barrier function will be presented. The Lyapunov function is encoded into the soft constraints (2.5), (2.8) and (2.9) to generate controls for achieving control objectives, whereas the barrier function is encoded into the hard constraints (2.4) and (2.7) to ensure safety. Particularly, define a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, the barrier function can be defined as:

$$h_s(\mathbf{x}) = D - \tau v \tag{3.1}$$

$$h_l(\mathbf{x}) = d_{\max} - y_{\text{lat}}\,\text{sign}(v_{\text{lat}}) - \frac{1}{2}\frac{v_{\text{lat}}^2}{a_{\max}} \tag{3.2}$$

where in this study, D can be calculated by the Euclidean distance between two robots, and under the polar trajectory assumption the lateral displacement and velocity can be calculated by $y_{lat} = \sqrt{x^2+y^2} - R_{path}$ and $v_{lat} = v\cos(\theta - \phi)$, respectively.

As in [4], consider an admissible set $\mathcal{C}$ of safe states of the controlled robot, which is defined by

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\} \tag{3.3}$$

$$\partial\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) = 0\} \tag{3.4}$$

$$Int(\mathcal{C}) = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) > 0\} \tag{3.5}$$

It is clear that when the state of the robot is in the interior of $\mathcal{C}$, $Int(\mathcal{C})$, the robot is said to be "safe" and any actions of the robot will be accepted. When the robot state is close to the set boundary $\partial\mathcal{C}$, we need to constrain the actions of the robot using the barrier function in order to ensure the robot stay in the safe set $\mathcal{C}$.

In general, while the hard constraints cannot be violated at all times, the constraints generated by Lyapunov functions that representing control objectives can be violated which means they should be achieved as much as possible.

3.1.1 Zeroing Control Barrier Function

The author in [4] proposed two kinds of barrier functions: reciprocal barrier function and zeroing barrier function. As stated in chapter 1, the main idea behind reciprocal barrier function B, is that as $\mathbf{x} \to \partial\mathcal{C}$, $B(\mathbf{x}) \to \infty$. With $\dot{B}(\mathbf{x}) \geq 0$ will guarantee that the set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^m : h(\mathbf{x}) \geq 0\}$ is invariant [4]. Instead of allowing barrier function to grow to infinity at the boundary of set $\mathcal{C}$, the zeroing barrier function will vanish as the robot state $\mathbf{x}$ approaches the boundary of $\mathcal{C}$. This study will only focus on zeroing barrier function for two reasons: First of all, either of these two

types of barrier functions will guarantee set invariant; Secondly, the unbounded function B might cause bad consequences numerically when implemented in practice.

To construct the control barrier functions for lane-keeping and adaptive speed control problem, some basic definitions in [4] is presented:

Definition 3.1.1. A continuous function α: $(-b, a) \to (-\infty, \infty)$ is said to belong to extended class $\mathcal{K}$ for some $a, b > 0$ if it is strictly increasing and $\alpha(0) = 0$.

Definition 3.1.2. For the dynamical system (2.3), a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is a **zeroing barrier function (ZBF)** for the set $\mathcal{C}$ defined by (3.3) - (3.5), if there exist an extended class $\mathcal{K}$ function α and a set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$ such that, for all $x \in \mathcal{D}$,

$$L_f h(\mathbf{x}) \geq -\alpha(h(\mathbf{x})) \tag{3.6}$$

Note that the previous condition $\dot{h}(\mathbf{x}) \geq 0$ is sometimes undesirable since it might be too restrictive to constrain the robot's motion, therefore the author in [4] relaxes the condition to (3.6). Followed by the above definition and a Lyapunov-like condition, the definition for zeroing control barrier function is as follows [4]:

Theorem 3.1.1. Given a set $\mathcal{C} \subset \mathbb{R}^n$ defined by (3.3) - (3.5) for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, the function h is called a **zeroing control barrier function (ZCBF)** defined on set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$, if there exists an extended class $\mathcal{K}$ function α such that:

$$\inf_{u \in U} [L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x}))] \geq 0, \forall \mathbf{x} \in \mathcal{D} \tag{3.7}$$

Given the above definitions, for a ZCBF $h(x)$, a set for control $\mathbf{u}$ can be defined as:

$$\mathcal{K}_h(\mathbf{x}) = \{\mathbf{u} \in U : L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x})) \geq 0\}$$

and selecting an input $\mathbf{u}$ in the above set will guarantee that the set $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^m : h(\mathbf{x}) \geq 0\}$ is

invariant which is concluded into the following corollary:

Corollary 3.1.1.1. Given a set $\mathcal{C} \subset \mathbb{R}^n$ defined by (3.3) - (3.5) for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, if h is a ZCBF on $\mathcal{D}$, then any Lipschitz continuous controller $u : \mathcal{D} \to U$ such that $u(\mathbf{x}) \in K_h(\mathbf{x})$ will render the set $\mathcal{C}$ forward invariant.

3.1.2 Control Lyapunov Function

This section will provide the basic definitions of control Lyapunov function, and the Lypunov functions related to the simultaneous lane-keeping and adaptive speed control problem will be given later. First, the definition of control Lyapunov function in [2] is presented below

Definition 3.1.3. A continuously differentiable function $V : X \times Z \to \mathbb{R}$ is an exponentially stabilizing control Lyapunov function (ES-CLF) if there exist positive constants $c_1, c_2, c_3 > 0$ such that for all $\mathbf{x} \in X \times Z$, the following inequalities hold:

$$c_1||x||^2 \leq V(x) \leq c_2||x||^2 \tag{3.8}$$

$$\inf_{u \in U}[L_f V(\mathbf{x}) + L_g V(\mathbf{x})\mathbf{u} + c_3 V(\mathbf{x})] \leq 0 \tag{3.9}$$

and we follow the same method in generating the control barrier function, a set for control $\mathbf{u}$ is defined as:

$$\mathcal{K}_V(\mathbf{x}) = \{\mathbf{u} \in U : L_f V(\mathbf{x}) + L_g V(\mathbf{x})\mathbf{u} + c_3 V(\mathbf{x}) \leq 0\}$$

in which any proper selection of control $\mathbf{u}$ will exponentially stabilize the system. In the simultaneous lane-keeping and adaptive cruise control problem, the appropriate control $\mathbf{u}$ will drive the robot velocity, position, and angular velocity in (2.5), (2.8) and (2.9) to the desired values.

To simultaneously make the robot drive at the desired velocity and track the desired path, as in [12], the adaptive speed control Lyapunov function can be defined as:

$$V_1(\mathbf{x}) = (v - v_d)^2 \tag{3.10}$$

where v_d is the desired forward velocity.

For the lane keeping control Lyapunov function, the input-output linearization technique can be used to generate a valid Lyapunov function. Consider the output $\eta(\mathbf{x})$:

$$\eta = \begin{pmatrix} x - R_{path}(\phi)\cos\phi \\ y - R_{path}(\phi)\sin\phi \end{pmatrix}$$

which has a relative degree two. According to the input-output linearization in [2]:

$$\ddot{\eta} = \mathcal{L}_f^2\eta + \mathcal{L}_g\mathcal{L}_f\eta\mathbf{u}$$
$$\mathbf{u} = -(\mathcal{L}_g\mathcal{L}_f\eta)^{-1}(\mathcal{L}_f^2\eta + K_p\eta + K_d\dot{\eta})$$

and the linearization of the output is:

$$\ddot{\eta} = -K_p\eta - K_d\dot{\eta}$$

where $\dot{\eta}$ can be calculated as:

$$\dot{\eta} = \begin{pmatrix} \dot{x} - \dot{R}_{path}(\phi)\cos\phi + \dot{\phi}R_{path}(\phi)\sin\phi \\ \dot{y} - \dot{R}_{path}(\phi)\sin\phi - \dot{\phi}R_{path}(\phi)\cos\phi \end{pmatrix}$$

Define a new variable η_n:

$$\eta_n = \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix}$$

the linear output dynamics can be written as:

$$\frac{d}{dt}\eta_n(t) = A_l \eta_n(t)$$

where A_l is Hurwitz:

$$A_l := \begin{bmatrix} 0, & I \\ -K_p, & -K_d \end{bmatrix}$$

Therefore, for any symmetric and positive definite matrix Q, there exists a positive definite matrix P satisfies the Lyapunov equation:

$$A_l^T + PA_l = -Q$$

By choosing K_p, K_d and Q, we can solve the above Lyapunov equation. Using the techniques in [], the following valid Lyapunov function can be defined, namely:

$$V_3(\mathbf{x}) = [\eta^T, \dot{\eta}^T] P [\eta^T, \dot{\eta}^T]^T, \tag{3.11}$$

which satisfies the exponentially stabilizing control Lyapunov function conditions [2].

To try to smooth the path, the author in [12] also defined a Lyapunov function for angular velocity:

$$V_2(\mathbf{x}) = \omega^2, \tag{3.12}$$

which corresponds to make the robot trajectory as straight as possible.

3.1.3 Quadratic Programming Control Solution

The problem of calculating a control input $\mathbf{u}$ belonging to the sets $\mathcal{K}_V$ and $\mathcal{K}_b$ leads to a quadratic programming problem. Because the choices of the above V_i, $i = 1,2,3$, cannot be simultaneously zero (more on this in the next section), slack variables δ_j, $j = 1,2,3$, were introduced in [12] leading to the augmented control vector $\tilde{\mathbf{u}}$ and the following associated Quadratic Program (QP):

$$\begin{aligned}
\tilde{\mathbf{u}}^* = \arg\min_{\tilde{\mathbf{u}}} \tilde{\mathbf{u}}^T \tilde{H} \tilde{\mathbf{u}} \\
\text{s.t. } L_f V_1(\mathbf{x}) + (L_g V_1(\mathbf{x}), -1, 0, 0)\tilde{\mathbf{u}} + \alpha_1 V_1(\mathbf{x}) \leq 0 \\
L_f V_2(\mathbf{x}) + (L_g V_2(\mathbf{x}), 0, -1, 0)\tilde{\mathbf{u}} + \alpha_2 V_2(\mathbf{x}) \leq 0 \\
L_f V_3(\mathbf{x}) + (L_g V_3(\mathbf{x}), 0, 0, -1)\tilde{\mathbf{u}} + \alpha_3 V_3(\mathbf{x}) \leq 0 \\
L_f h_s(\mathbf{x}) + (L_g h_s(\mathbf{x}), 0, 0, 0)\tilde{\mathbf{u}} + \alpha_s h_s(\mathbf{x}) \geq 0 \\
L_f h_l(\mathbf{x}) + (L_g h_l(\mathbf{x}), 0, 0, 0)\tilde{\mathbf{u}} + \alpha_l h_l(\mathbf{x}) \geq 0
\end{aligned} \tag{3.13}$$

where

$$\tilde{\mathbf{u}} = \begin{pmatrix} u_l \\ u_a \\ \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} \qquad \tilde{H} = \begin{pmatrix} p_1 & 0 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 & 0 \\ 0 & 0 & p_3 & 0 & 0 \\ 0 & 0 & 0 & p_4 & 0 \\ 0 & 0 & 0 & 0 & p_5 \end{pmatrix}$$

the p_j elements on the diagonal of $\tilde{H}$ act as penalties and the α's are all positive parameters. The solution to this optimization problem is always in $\mathcal{K}_b$ and is in $\mathcal{K}_V \cap \mathcal{K}_b$ if all δ's are positive.

3.2 Drawbacks

The combined control Lyapunov and barrier function of [4] offers a flexible and efficient framework for the design of control laws for vehicles. However, the particular choice of functions in [12] discussed above has two major drawbacks, namely:

- The components of the Lyapunov functions V_i, $i = 1, 2, 3$ are not compatible. The vehicle cannot converge to the desired trajectory, i.e.

$$\dot{\eta} = \eta \to 0 \implies V_3 \to 0$$

with the desired longitudinal velocity, i.e.

$$v \to v_d \implies V_1 \to 0$$

while the angular velocity converges to zero, i.e.

$$\omega \to 0 \implies V_2 \to 0$$

except when the target trajectory is a straight line.

- When the leading vehicle velocity, v_l, is lower than the desired velocity, v_d, the function V_1 cannot converge to zero. That is the barrier function h_s and the function V_1 are not compatible.

The above incompatibilities make the use of slack variables a necessity rather than a luxury. It also means that even with the control barrier function constraint is inactive, the control law cannot guarantee $\dot{V}(\mathbf{x}) < 0$ at all times.

More specifically, assume there's only one CLF and one CBF constraint and consider a more general QP:

$$
\begin{aligned}
u^* = \arg\min_u \frac{1}{2}(u^2 + p\delta^2) \\
\text{s.t. } L_f V(\mathbf{x}) + L_g V(\mathbf{x}) u + \alpha V(\mathbf{x}) \leq \delta \\
L_f h(\mathbf{x}) + L_g h(\mathbf{x}) u + \alpha_h h(\mathbf{x}) \geq 0
\end{aligned}
$$

and we have:

$$
u = -\frac{L_f V(\mathbf{x}) + \alpha V(\mathbf{x})}{L_g V(\mathbf{x})^2 - \frac{1}{p}} L_g V(\mathbf{x})
$$

which is a similar expression to the point-wise minimum norm (PMN) controller that is asymptotically stabilizing the system [8]. Note that the above expression can only approximate the PMN controller since no matter how large p is, the controller cannot asymptotically stabilize the system.

Chapter 4

Proposed Control Lyapunov and Barrier Function

One drawback of the combined CLF-CBF approach is that sometimes it uses all control inputs to guarantee safety specification ignoring the performance of the system even though they can be handled simultaneously [8]. The CLF-CBF-QP framework could overcome this problem with relaxation parameters taking into account to make QP feasible. However, this framework can not offer guarantees on stability of the robotic system which is not as desired in the simultaneous lane-keeping and adaptive speed control problem. To overcome this problem, a modification based on the compatibility of the CLF constraints is considered.

4.1 Control Lyapunov Functions for Lane-keeping

Inspired by the polar target trajectory, the curvature of the trajectory can be naturally taken into consideration. A vehicle traversing a curve $r = r(\phi)$ with tangential velocity v has as angular velocity:

$$\omega = c(\phi)\, v$$

where $c(\phi)$ is the curvature:

$$c(\phi) = \frac{|r^2 + 2(r')^2 - rr''|}{(r^2 + (r')^2)^{\frac{3}{2}}} \tag{4.1}$$

here expressed in polar coordinates, in which r' and r'' denote first- and second-derivatives taken with respect to ϕ. Give the polar trajectory (2.10), the first and second derivatives are:

$$r' = nb\cos(n\phi)$$
$$r'' = -n^2 b\sin(n\phi)$$

With the above formulations, the $V_2(\mathbf{x})$ in (3.12) can be replaced by:

$$V_2(\mathbf{x}) = (\omega - \omega_d)^2 \tag{4.2}$$

$$\omega_d = c\,v \tag{4.3}$$

where ω_d plays the role of the desired angular velocity. Note that v is used here, which is the vehicle's tangential velocity instead of the desired velocity, v_d, which may differ from v if adaptive speed control is engaged. The calculation of c requires only the additional evaluation of the first- and second-derivatives of the path with respect to ϕ, which is readily available.

4.2 Control Lyapunov Functions for Adaptive Speed Control

By the above modification, as $V_2 \to 0$, the robot desired trajectory is no longer a straight line and the compatibility holds for V_3 and V_2. However, even after the above replacement of V_2, it is still not possible that all the V_i's to converge to zero simultaneously, even when the vehicle lies perfectly on its desired trajectory if the adaptive cruise control is engaged.

Since the barrier function should always remain positive i.e.

$$h_s = D - \tau v_f > 0$$

a new approach can be proposed to combine the barrier function into the first control objective:

$$v \to min(v_d, \frac{D}{\tau}) \tag{4.4}$$

When two robots are relatively far away i.e.

$$\frac{D}{\tau} > v_d$$

the following robot will try to drive at the desired velocity:

$$v \to v_d$$

which is the original control objective, and when the safety distance eventually arrives, the following robot velocity will follow:

$$v \to \frac{D}{\tau}$$

in order to obey the barrier constraint i.e. maintain the safe distance.

With this is mind, the barrier function (3.1), h_s can be eliminate ,and V_1 can be replaced by:

$$V_1(\mathbf{x}) = \left(v - \min\left(v_d, \frac{D}{\tau} \right) \right)^2. \tag{4.5}$$

One should notice that $V_1(\mathbf{x})$ is not smooth, the following section will provide further discussion on non-smooth control Lyapunov functions.

4.3 Non-smooth Analysis on Proposed CLF

From the above modification, a non-smooth Control Lyapunov Function is needed to deal with the change of the desired velocity. One famous non-smooth analysis literature [5] is revisited here to provide some important definitions and propositions on non-smooth control Lyapunov functions.

First of all, the definition of generalized gradient is stated as follows:

Definition 4.3.1. Let $f : \mathbb{R}_d \to \mathbb{R}$ be a locally Lipschitz function, and let $\Omega_f \subset \mathbb{R}_d$ denote the set of points where f fails to be differentiable. The generalized gradient $\partial f : \mathbb{R}_d \to B(\mathbb{R}_b)$ of f is defined by:

$$\partial f(x) \triangleq co\{\lim_{i\to\infty} \nabla f(x_i) : x_i \to x, x_i \notin S \cup \Omega_f\}$$

where

co	convex hull
S	a set of measure zero

The definition is saying that the generalize gradient f at x is the convex combinations of all limits of f at differentiable point.

The following propositions in [5] are also useful for analysing the non-smooth Control Barrier Functions:

Proposition 4.3.1. If $f : \mathbb{R}_d \to \mathbb{R}$ is locally Lipschitz at $x \in \mathbb{R}_d$, then the following statements hold:

(i) $\partial f(x)$ is nonempty, compact, and convex.

(ii) The set-valued map $\partial f : \mathbb{R}_d \to \mathcal{B}(\mathbb{R}_d), x \mapsto \partial f(x)$, is upper semi-continuous and locally bounded at x.

(iii) If f is continuously differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$.

Proposition 4.3.2. For $k \in \{1,\ldots,m\}$, let $f_k : \mathbb{R}_d \to \mathbb{R}$ be locally Lipschitz at $x \in \mathbb{R}_d$, and define the functions f_{max}, $f_{min} : \mathbb{R}_d \to \mathbb{R}$ by:

$$f_{max}(y) \triangleq max\{f_k(y) : k \in \{1,\ldots,m\}\},$$
$$f_{min}(y) \triangleq min\{f_k(y) : k \in \{1,\ldots,m\}\},$$

and the following statements hold:

(i) f_{max} and f_{min} are locally Lipschitz at x.

(ii) Let $I_{max}(x)$ denote the set of indices k for which $f_k(x) = f_{max}(x)$. Then

$$\partial f_{max}(x) \subseteq co \cup \{\partial f_i(x) : i \in I_{max}(x)\}.$$

Furthermore, if f_i is regular at x for all $i \in I_max(x)$, then equality holds and f_{max} is regular at x,

(iii) Let $I_{min}(x)$ denote the set of indices k for which $f_k(x) = f_{min}(x)$. Then

$$\partial f_{min}(x) \subseteq co \cup \{\partial f_i(x) : i \in I_{min}(x)\}.$$

Furthermore, if f_i is regular at x for all $i \in I_{min}(x)$, then equality holds and f_{min} is regular at x,

The *Proposition* 4.3.2 gives the generalized gradient of the minimum and maximum value of a finite set of functions which can be used on the analysis of the non-smooth CLF (4.5).

The CLF (4.5) can be rewritten as:

$$V_1(\mathbf{x}) = \max\{V^1(\mathbf{x}), V^2(\mathbf{x})\}$$

where

$$V^1(\mathbf{x}) = (v - v_d)^2$$
$$V^2(\mathbf{x}) = (v - \frac{D}{\tau})^2$$

Both functions are continuously differentiable which means they are also locally Lipschitz and regular.

According to (i) and (ii) in *Proposition* 4.3.2, $V_1(\mathbf{x})$ is locally Lipschitz and regular, and its generalized gradient is:

$$\partial V_1(\mathbf{x}) = \begin{cases} \dot{V}^1(\mathbf{x}) & , v_d < \frac{D}{\tau} \\ [\dot{V}^1(\mathbf{x}), \dot{V}^2(\mathbf{x})] & , v_d = \frac{D}{\tau} \\ \dot{V}^2(\mathbf{x}) & , v_d > \frac{D}{\tau} \end{cases} \tag{4.6}$$

Note that in practice, the intermediate condition $v_d = D/\tau$ happening is very rare, and either the other two conditions can be take into consideration in the simulation part without any loss.

4.4 Combination of New Control Lyapunov Functions

With the above mortification, the control Lyapunov functions become:

$$V_1(\mathbf{x}) = (v - min(v_d, \frac{D}{\tau}))^2$$
$$V_2(\mathbf{x}) = (\omega - \omega_d)^2$$
$$V_3(\mathbf{x}) = [\eta^T, \dot{\eta}^T] P [\eta^T, \dot{\eta}^T]^T$$

Since the second and third control objectives are not conflicting to each other anymore, δ_2 and δ_3 can be canceled. Additionally, since the speed barrier function has been integrated into the first control objective, δ_1 can be canceled as well.

Rewrite the constraints in (3.13) as:

$$L_f V_i(\mathbf{x}) + L_g V_i(\mathbf{x})\mathbf{u} + c_i V_i(\mathbf{x}) \leq 0 \tag{4.7}$$
$$L_f h_{lk}(\mathbf{x}) + L_g h_{lk}(\mathbf{x})\mathbf{u} + \gamma h_{lk}(\mathbf{x}) \geq 0 \tag{4.8}$$

for i = 1,2,3 and

$$\mathbf{u} = \begin{pmatrix} u_l \\ u_a \end{pmatrix}$$

To form a more natural optimization problem, a combination of these three CLF constraints is designed. First of all, the CLF constraints can be rewritten as:

$$C_i(\mathbf{x}, \mathbf{u}) = L_g V_i(\mathbf{x})\mathbf{u} + L_f V_i(\mathbf{x}) + c_i V_i(\mathbf{x})$$
$$= A^i_{clf}(\mathbf{x})\mathbf{u} - b^i_{clf}(\mathbf{x}) \leq 0$$

Because it is now possible for all V_i's to converge to zero at the desired trajectory, one can combine all of the V_i's into a single control Lyapunov function:

$$V(\mathbf{x}) = \sum_{i=1}^{N=3} k_i V_i(\mathbf{x}) \tag{4.9}$$

with $k_i > 0,\ i = 1,2,3$. This control Lyapunov function ensures convergence to the desired trajectory and to either the desired speed v_d or a safe distance from a leading vehicle. The combined CLF constraints can be rewritten as follows:

$$\begin{aligned} C(\mathbf{x},\mathbf{u}) &= \sum_{i=1}^{N=3} k_i (L_g V_i(\mathbf{x})\mathbf{u} + L_f V_i(\mathbf{x}) + c_i V_i(\mathbf{x})) \\ &= \sum_{i=1}^{N=3} k_i \cdot C_i(\mathbf{x},\mathbf{u}) \\ &= \sum_{i=1}^{N=3} k_i A^i_{clf}(\mathbf{x})\mathbf{u} - \sum_{i=1}^{N=3} k_i b^i_{clf}(\mathbf{x}) \\ &= A_{CLF}\mathbf{u} - B_{CLF} \leq 0 \end{aligned}$$

and in the next section, the formulations of a more general inequality constrained optimization problem will be presented to calculate the explicit solution for the QP problem.

Chapter 5

Explicit Solution of the QP Problem

A quadratic program (QP) is a constrained optimization problem with an objective function being quadratic. Quadratic programming has been widely studied over the past few decades and applied to many non-linear optimization based fields of study.

This section starts with the basic formulation of the quadratic program, followed by the study of Karush-Kuhn-Tucker (KKT) conditions for the optimization problem with multiple inequality constraints and how it is related to the simultaneous lane-keeping and adaptive speed control problem.

5.1 Inequality Constrained Optimization with KKT Conditions

5.1.1 Problem Statement

Recall that the general quadratic program is in the form of:

$$Minimize\, f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T Q\mathbf{u} + c\mathbf{u}$$
$$subject\ to\ A\mathbf{u} \leq b \tag{5.1}$$

where Q is a symmetric matrix whose diagonal elements represent the penalties of the quadratic terms, and the elements of c are the penalties of the linear terms in the objective function.

The optimal variable is denoted as $\mathbf{u}$, and the inequality constraints are defined by a linear inequality $A\mathbf{u} \leq b$. The quadratic program is said to be feasible if there is a solution under the above constraint.

When Q is positive definite, the objective function $f(\mathbf{u})$ is strictly convex which means the problem is able to find the global minimum.

5.1.2 Karush-Kuhn-Tucker conditions

Combining the inequality constraint in (5.1) into the quadratic objective function, a Lagrangian can be defined as:

$$\mathcal{L}(\mathbf{u}, \lambda) = f(\mathbf{u}) + \lambda(A\mathbf{u} - b) \tag{5.2}$$

The Karush-Kuhn-Tucker conditions for a local minimum are stated as follows:

$$\begin{cases} \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}, \lambda) = 0 \\ \lambda \geq 0 \\ \lambda g(\mathbf{u}) = 0 \\ g(\mathbf{u}) \leq 0 \\ \nabla_{\mathbf{uu}}\mathcal{L}(\mathbf{u}, \lambda) \succ 0 \end{cases} \tag{5.3}$$

Note that the Lagrangian (5.2) also satisfies multiple inequalities which can be rewritten as:

$$\begin{aligned}\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}) &= f(\mathbf{u}) + \sum_{i=1}^{N} \lambda_i g_i(\mathbf{u}) \\ &= f(\mathbf{u}) + \boldsymbol{\lambda}\mathbf{g}(\mathbf{u})\end{aligned} \tag{5.4}$$

which in our case, the two inequality constraints are (4.7).

5.2 KKT condition-based Optimization Problem

After combining the adaptive speed control into the control Lyapunov function, the problem is only left with the barrier function, h_l, for lane-keeping. This leads to a much simplified QP, with only two constraints, namely:

$$\begin{aligned}\mathbf{u}^* = \arg \min_{\mathbf{u}=(u_l,u_a)} \; & \frac{1}{2}\mathbf{u}^T H \mathbf{u} \\ \text{s.t. } & L_f V(\mathbf{x}) + L_g V(\mathbf{x})\mathbf{u} + \alpha V(\mathbf{x}) \leq 0 \\ & L_f h_l(\mathbf{x}) + L_g h_l(\mathbf{x})\mathbf{u} + \alpha_s h_l(\mathbf{x}) \geq 0\end{aligned} \tag{5.5}$$

Note that there is no need for slack variables since the barrier and control Lyapunov functions are now compatible. Since $L_g h_l(\mathbf{x}) \neq 0$, $L_g V(\mathbf{x})$ and $L_g h_l(\mathbf{x})$ are linearly independent. Furthermore, since $\mathbf{u}$ has dimension 2, Problem (5.5) is generically solvable, which means that it is possible to simultaneously enforce exponential stability as well as the barrier function. Note that such properties can not be enforced with the setup of [12].

The above problem can be cast in the standard QP form:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \frac{1}{2}\mathbf{u}^T H \mathbf{u} \\ \text{s.t.} \quad & a_V^T \mathbf{u} \leq b_V \\ & a_h^T \mathbf{u} \leq b_h \end{aligned} \tag{5.6}$$

where

$$a_V = L_g V(\mathbf{x})^T$$

$$a_h = -L_g h_l(\mathbf{x})^T$$

$$b_V = -L_f V(\mathbf{x}) - \alpha V(\mathbf{x})$$

$$b_h = L_f h_l(\mathbf{x}) + \alpha_s h_l(\mathbf{x})$$

$$H = \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix}$$

$$\mathbf{u} = \begin{pmatrix} u_l \\ u_a \end{pmatrix}$$

In this form, it is possible to explicitly characterize all optimal solutions. The key is to write the associated Karush-Kuhn-Tucker (KKT) conditions as discussed above in terms of the primal variable $\mathbf{u}$ and the dual variable $\boldsymbol{\lambda} = (\lambda_V, \lambda_h)$. First, the Lagrangian towards this form can be defined as:

$$\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}) = \frac{1}{2}\mathbf{u}^T H\mathbf{u} + \lambda_V(a_V^T\mathbf{u} - b_V) + \lambda_h(a_h^T\mathbf{u} - b_h)$$

According to the KKT condition (5.1.2), the following equations can be generated:

$$\begin{aligned} H\mathbf{u} + a_V\lambda_V + a_h\lambda_h &= 0 \\ \lambda_V(a_V^T\mathbf{u} - b_V) &= 0 \\ \lambda_h(a_h^T\mathbf{u} - b_h) &= 0 \\ \lambda &\geq 0 \end{aligned} \tag{5.7}$$

5.3 Explicit Solution To the Simultaneous Lane-Keeping and Adaptive Speed Control Problem

According to the above analysis, all possible solutions to the condition 5.7 are analyzed as follows:

- **Case 1**: Both constraints are inactive. This is a trivial case in which the robot motion is neither satisfying the soft nor hard constraint. In this case:

$$a_V^T\mathbf{u}^* < b_V$$
$$a_h^T\mathbf{u}^* < b_h$$
$$\lambda_V = 0$$
$$\lambda_h = 0$$

and since H is positive definite, the solution is:

$$\mathbf{u}^* = 0, \tag{5.8}$$

- **Case 2**: The control barrier function is inactive, namely, the robot motion is far from violating the hard constraint. In this case:

$$a_V^T \mathbf{u}^* = b_V$$
$$a_h^T \mathbf{u}^* < b_h$$
$$\lambda_V > 0$$
$$\lambda_h = 0$$

and the corresponding Lagrangian is:

$$\begin{aligned} \mathcal{L}(\mathbf{u}, \lambda_V) &= \frac{1}{2}\mathbf{u}^T H \mathbf{u} + \lambda_1 g_V(\mathbf{u}) \\ &= \frac{1}{2}\mathbf{u}^T H \mathbf{u} + \lambda_V (a_V^T \mathbf{u} - b_V) \end{aligned} \tag{5.9}$$

and the KKT condition gives:

$$H\mathbf{u} + a_V \lambda_V = 0 \tag{5.10}$$
$$a_V^T \mathbf{u} - b_V = 0 \tag{5.11}$$

since a_V is a column vector, the derivative ends up in the form of (5.10). The equation (5.10) yields:

$$\mathbf{u} = -H^{-1} a_V \lambda_V \tag{5.12}$$

substitute back to (5.11) gives:

$$\begin{aligned} a_V^T \mathbf{u} &= -a_V^T H^{-1} a_V \lambda_V \\ &= b_V \end{aligned}$$

therefore:

$$\lambda_V = -\frac{b_V}{a_V^T H^{-1} a_V}$$

plug back to (5.12) and finally yields:

$$\mathbf{u} = \frac{b_V}{a_V^T H^{-1} a_V} H^{-1} a_V$$

therefore the solution (5.8) for **Case 1** is generated, which is:

$$\mathbf{u}_V^* = \frac{b_V}{a_V^T H^{-1} a_V} H^{-1} a_V, \tag{5.13}$$

- **Case 3**:The control Lyapunov function is inactive. This case corresponds to when the robot needs to ignore the soft constraint in order to satisfy the barrier constraint; in this case:

$$a_V^T \mathbf{u}^* < b_V$$

$$a_h^T \mathbf{u}^* = b_h$$

$$\lambda_V = 0$$

$$\lambda_h > 0$$

and the solution can be calculated in the same manner:

$$\mathbf{u}_l^* = \frac{b_l}{a_l^T H^{-1} a_l} H^{-1} a_l. \tag{5.14}$$

The optimal control is:

$$\mathbf{u}^* = \arg\min({\mathbf{u}_V^*}^T H \mathbf{u}_V, {\mathbf{u}_l^*}^T H \mathbf{u}_l), \tag{5.15}$$

- **Case 4**: Both functions are active; in this case:

$$a_V^T \mathbf{u}^* = b_V$$
$$a_h^T \mathbf{u}^* = b_h$$
$$\lambda_V > 0$$
$$\lambda_h > 0$$

and

$$\mathbf{u}^* = A^{-1} b, \tag{5.16}$$

where

$$A = \begin{bmatrix} a_V^T \\ a_h^T \end{bmatrix}, \qquad b = \begin{pmatrix} b_V \\ b_h \end{pmatrix}.$$

The expressions obtained in the above 4 cases come from an analysis of the solutions to the KKT conditions (5.7) under the assumptions that H is positive definite and, in Case 4 only, that A is non-singular. This will be the case under the previously discussed linear independence property. When this property does not hold, an *ad hoc* least-squares solution exists and is well defined and may be used instead.

In order to distinguish each case is to be considered, one should progress in order from Case 1 to Case 4, substituting the optimal control $\mathbf{u}^*$ and verifying whether the associated constraints are indeed active or inactive. Most of the time, either Case 2 or Case 3 is the one that leads to the optimal control.

Chapter 6

Examples

This chapter presents two simulation examples to illustrate how the proposed simultaneous lane-keeping and adaptive cruise control strategies perform. The simulation setups for this problem are also discussed in details.

6.1 Simulation Setups

Even though the calculations here are performed for a continuous-time setup, to make the simulation more realistic, a fixed time step `ode45` is designed. The main idea is that, after calculating the optimal control using the above formulations, a fourth-order Runge-Kutta integration method (`ode45`) is called in MATLAB, and calculating for a fixed time period starting from the initial time t_0.

The control calculation during the simulation is done at a rate of 100Hz and kept constant between the sampling interval. Then the latest time t will be considered as the new t_0, and the latest robot state will be considered as the new initial state x_0.

To coincide with the modified robot model 2.2, the initial position of the robot needs to be

shifted by:

$$x_{shift} = x_0 + a\cos(\phi)$$

$$y_{shift} = y_0 + a\sin(\phi)$$

The flowchart of the simulation is shown in Figure 6.1:

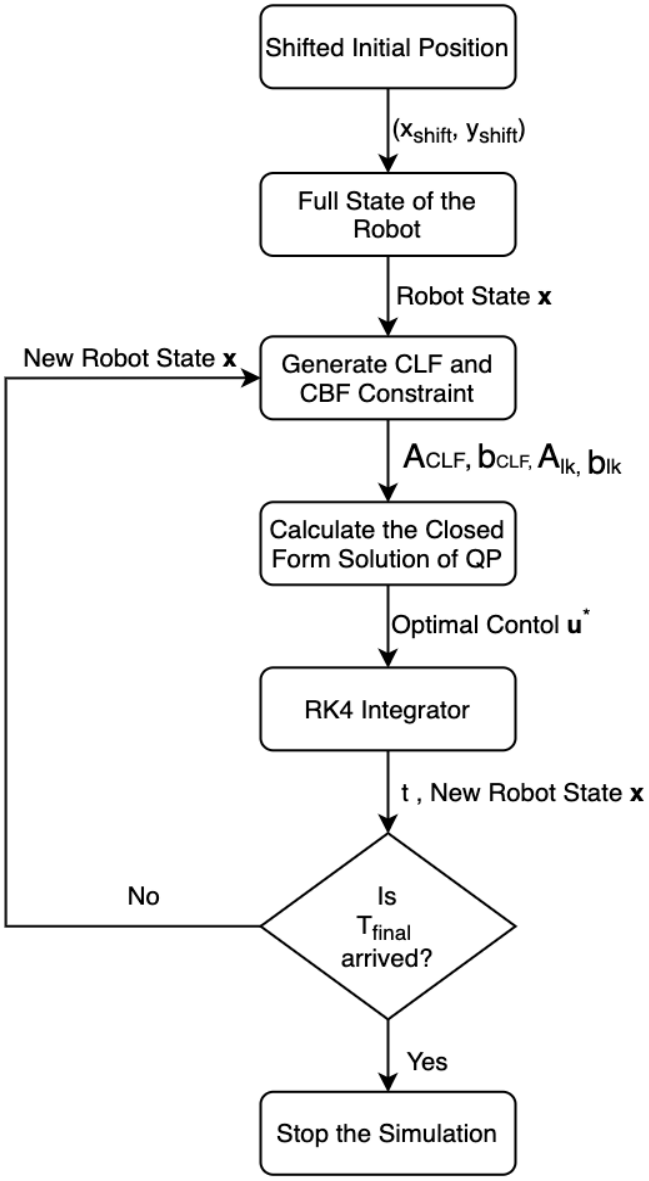


Figure 6.1: The Simulation Flowchart

6.2 Example A: Asymptotic Convergence with lane-keeping and Adaptive Speed Control

In this example, the result of simultaneous lane-keeping and adaptive speed control problem is provided under the above formulations. The the following robot is placed exactly on the path at:

$$(x, y) = (0, -R_{\text{path}}(-\frac{\pi}{2}))$$

with orientation $\theta = 0$. A leading robot is placed ahead of the following robot as shown in Fig. 6.2. Both robots have the same physical parameters, shown in Table. 6.1, with the only difference being that the leading robot's desired speed is half the desired speed of the following robots. This means that somewhere along the path the adaptive speed control should be active for the following robot.

The initial speed of the following robot is set to be twice. With these settings, a non-zero initial value can be expected for the Lyapunov function that should converge to zero as well as observe a change in the speed of the following robot when it approaches the prescribed safety distance.

Table 6.1: Prameters

R	0.9	m
b	0.23	m
n	3	m
d_{max}	0.15	m
τ	1.8	.
v_{df}	0.2	m/s
v_{dl}	0.1	m/s
m	0.69	kg
a	0.02	m
I_z	1.46e-3	$kg \cdot m^2$
a_{max}	0.3g	m/s^2

Fig. 6.2 shows the following robot trajectory. Red (blue) dot and arrow indicate position

and orientation of the following (leading) robot; the following robot remains above the safety distance while in the green marked path; red and blue dots mark the point in which the safe distance is reached; the following robot keeps the safe distance while in the red marked path. In this example, both robots remain very close to the desired trajectory (the center line in the path) and never activate the barrier function. The velocity and orientation of the following robot quickly converge to the desired velocity and orientation (Figs. 6.3 and 6.5), and so does the Lyapunov function Fig 6.6, which quickly converges to zero and remain very small even after the following robots reaches the minimum safe distance. Small fluctuations in the value of the Lyapunov function come from the fact that we insisted on a (more realistic) discrete-time implementation for the controller.

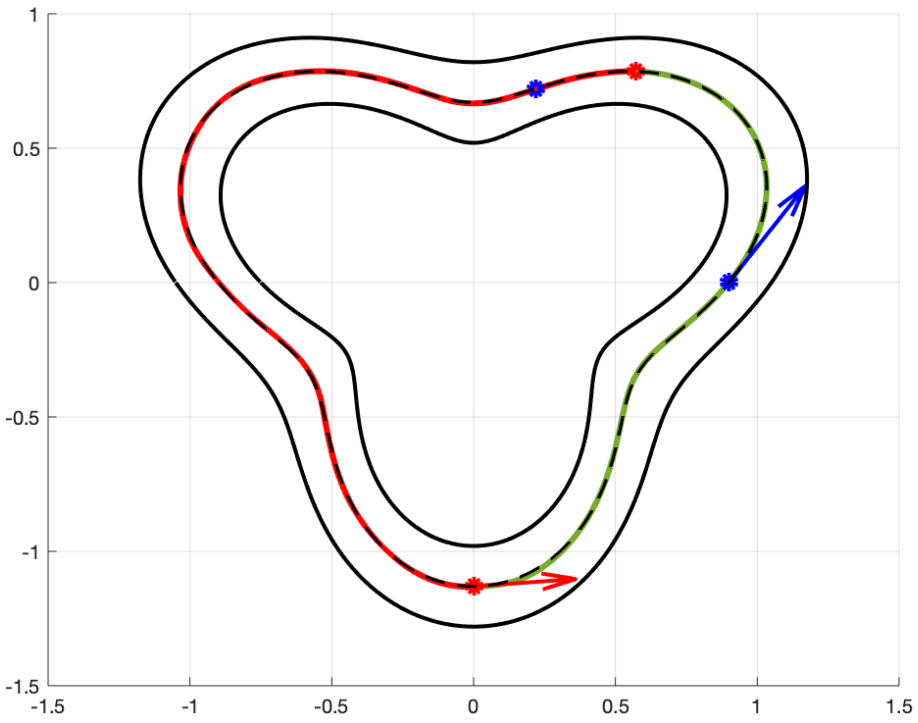


Figure 6.2: Example A: Asymptotic Convergence with lane-keeping and Adaptive Speed Control

Fig. 6.3 shows more details on the following robot's longitudinal velocity. The red dashed

curve represents the desired velocity v_{df} when the distance between two robots are relatively large which is set to be a constant, and the blue dashed curve is the desired safety speed D/τ used in (4.5) when two robots reach the safe distance. The black solid curve is the following robot longitudinal velocity. Note that since the following robot initial velocity is set to be twice as v_{df}, it will quickly converge to the desired velocity if the safe distance is not reached. Then when the safe distance is arrived, the following robot longitudinal velocity will converge to the minimum safe speed at around 14s.

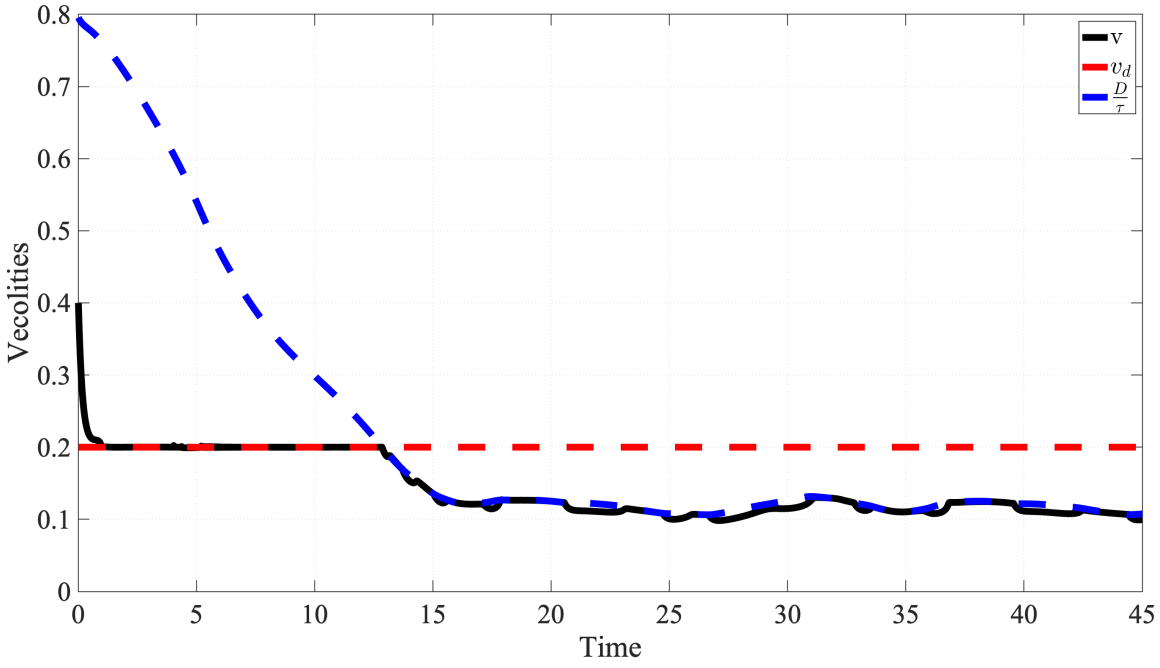
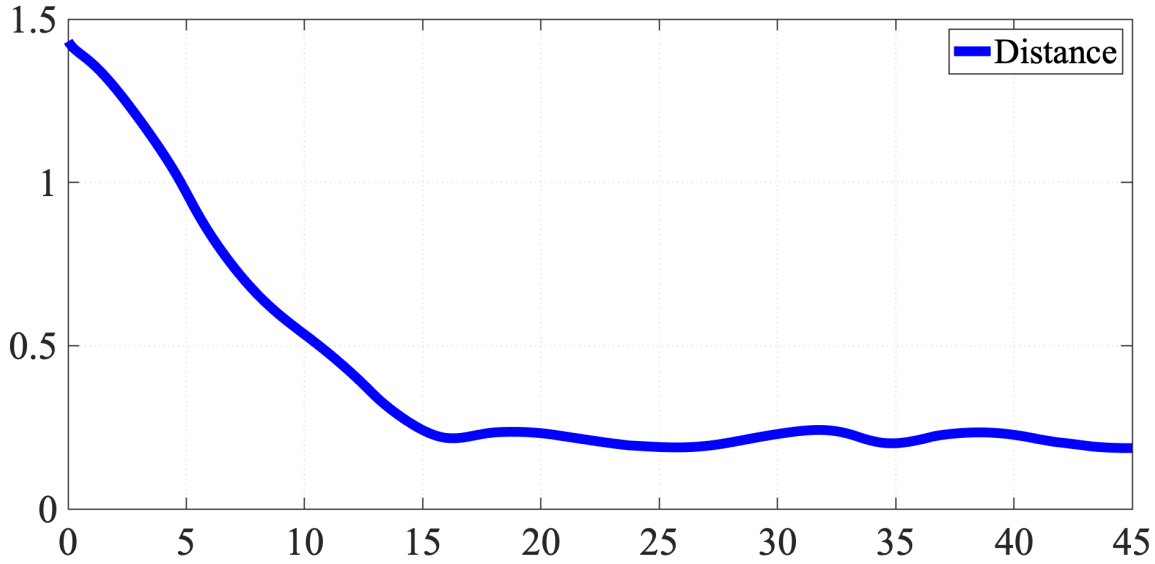


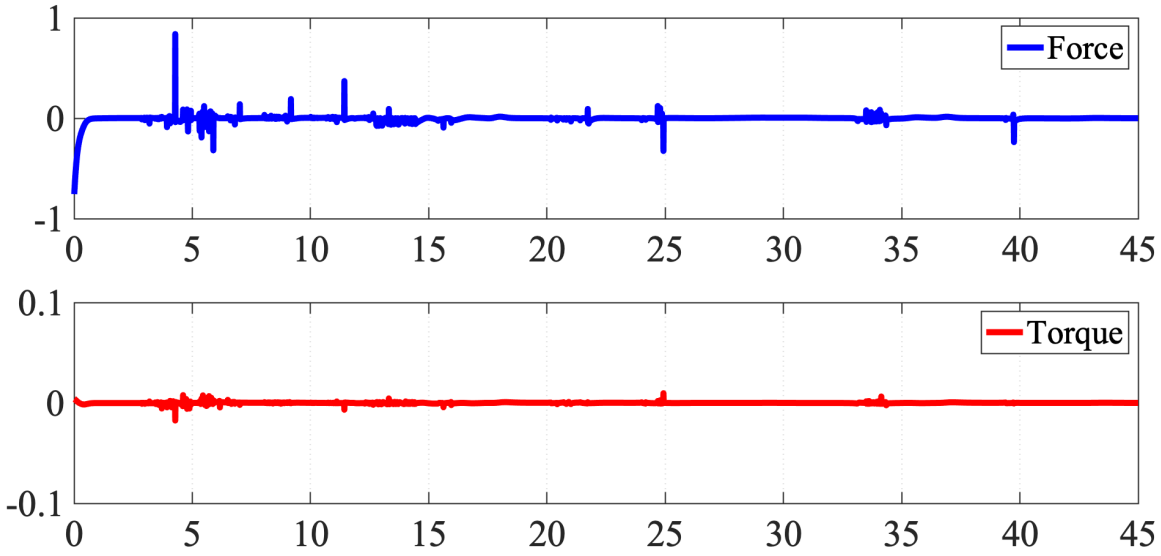
Figure 6.3: Example A: longitudinal velocity of the following robot (black solid), safe distance (blue dashed) and desired longitudinal velocity (red dashed).

Fig. 6.4 (a) is another demonstration of how the proposed Lyapunov function ensure the adaptive speed control safety specification. As we can see, the distance between two robots will remain around a constant once the safe distance is arrived. Fig. 6.4 (b) shows the force and torque control inputs calculated by the explicit solution of QP.

Fig. 6.5 shows the angular velocity of the robot together with the desired angular velocity.



(a) Distance



(b) Force/torque inputs

Figure 6.4: Example A: Distance between two robots and force/torque control inputs

The red curve represents the desired angular velocity calculated by the curvature equation (4.1) and (4.2). One can see that the robot angular velocity (black curve) matches ω_d (red curve) as desired. Fig 6.6 shows the value of combined CLFs which is able to converge to zero when barrier

function is inactive.

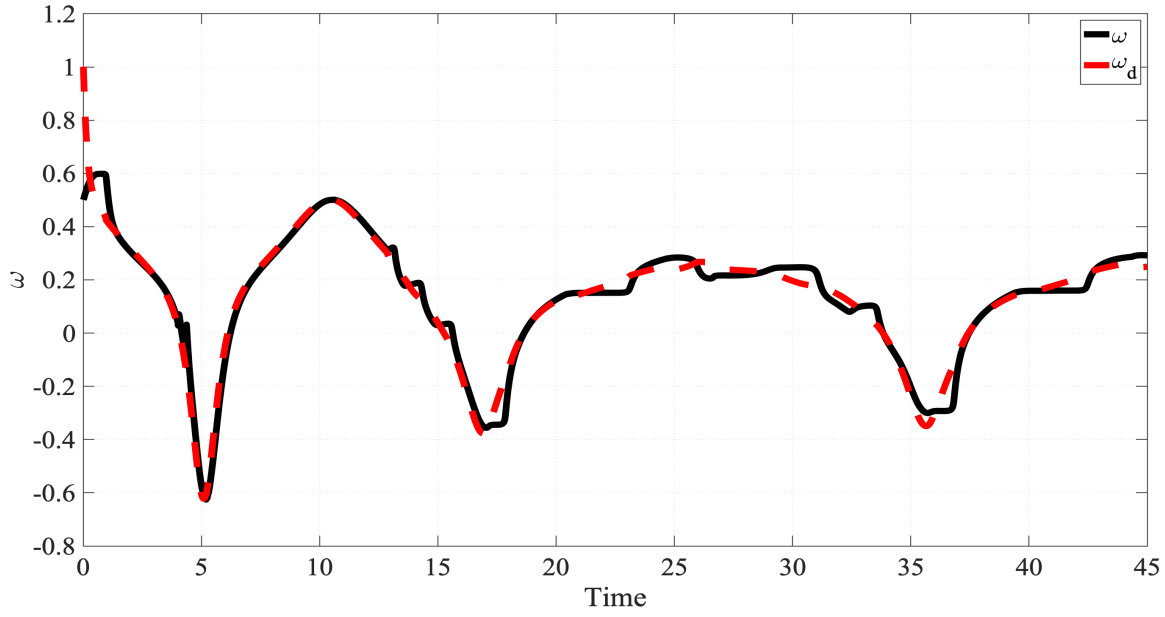


Figure 6.5: Example A: following robot angular velocity (black solid), desired angular velocity (red dashed).

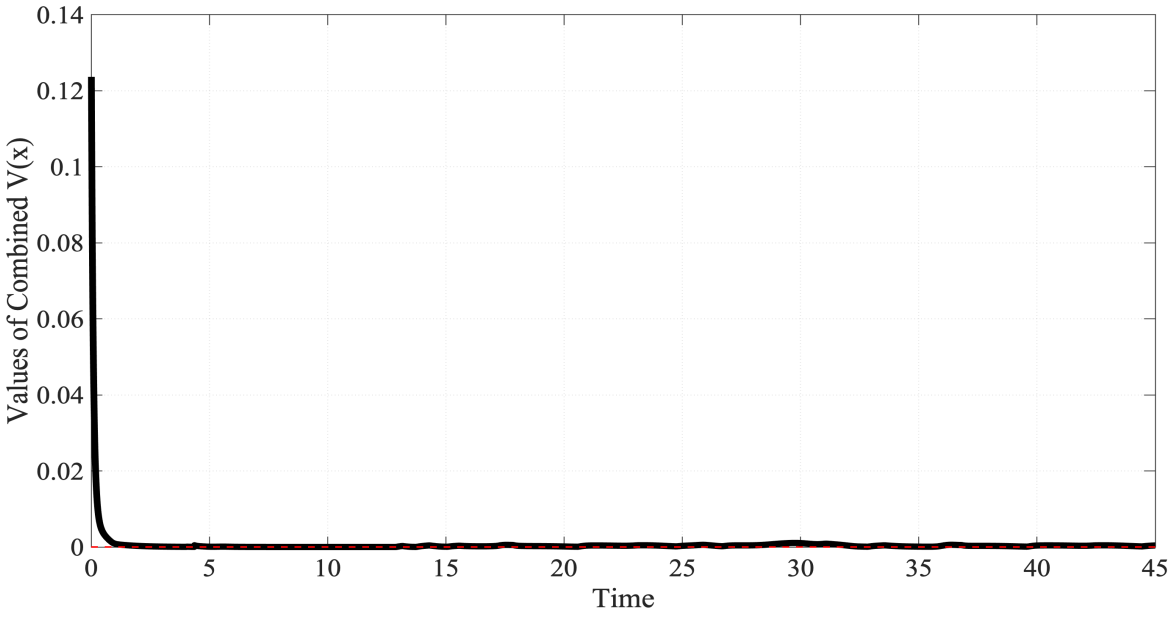


Figure 6.6: Example A: Lyapunov function $V(x)$.

6.3 Example B: Activating the Barrier Function

In this example, the following robot is placed near the center of the desired trajectory (dashed) is initialized, and with an "aggressive" orientation angle $\theta = -\frac{\pi}{3}$ as the red arrow shown in Fig. 6.7. With these settings, the following robot will tend to move outside of the lane. Two simulations will be used to illustrate how the barrier function restrict the behaviour of the robot.

In the first, the barrier function is artificially kept inactive at all times. The control Lyapunov functions are the only constraints that stabilizing the robot system to achieve control objectives. With this aggressive orientation, which, as seen in Fig. 6.7 will lead to a robot trajectory that leaves the prescribed safety lane. The barrier function takes negative values which means the barrier function is violated, as seen in Fig. 6.8.

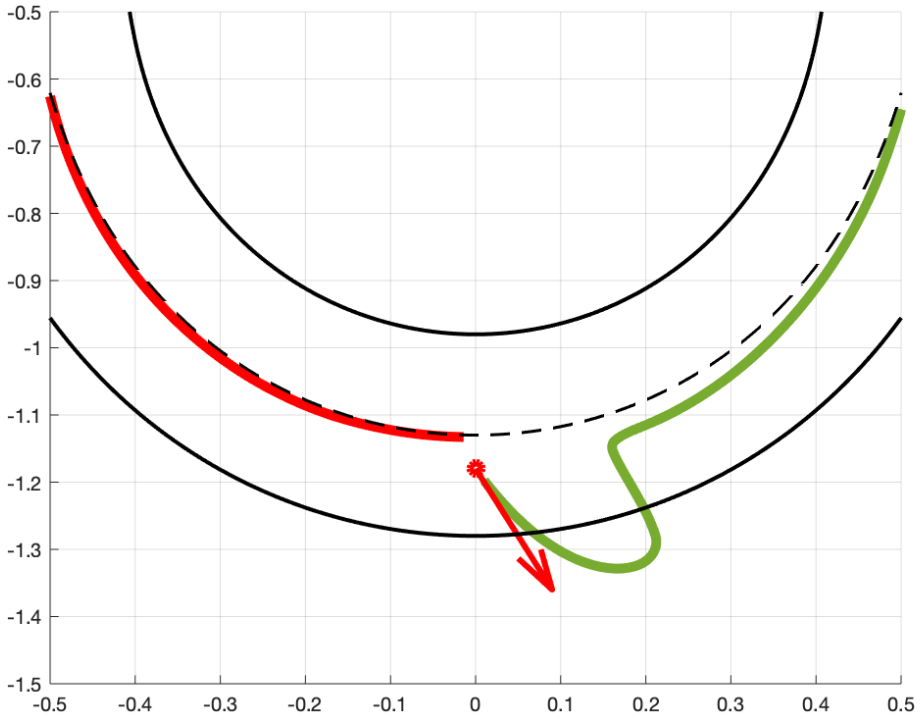


Figure 6.7: When control barrier function is inactive

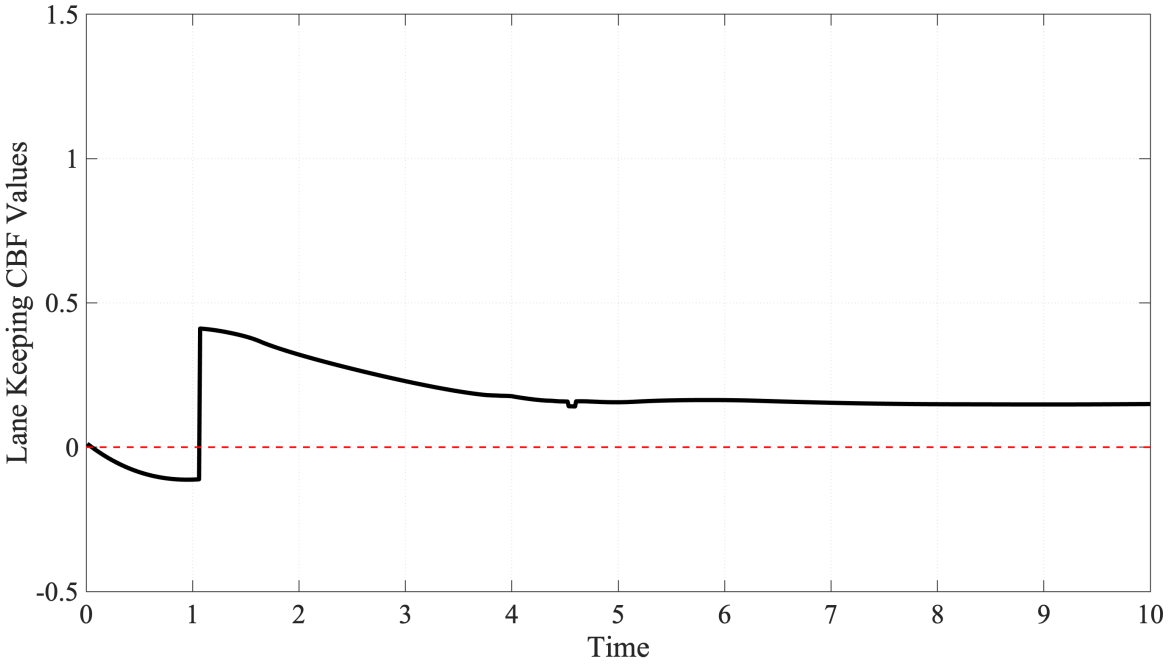


Figure 6.8: Values of control barrier function

In the second, the barrier function is active and the controls are affected by the barrier function. In this case, even though the robot is moving towards the outside of the lane at first, it will then make a turn before arriving the boundary of the lane. This results in the trajectory shown in Fig. 6.9, in which the robot's trajectory remains well within the boundaries of the prescribed lanes and the barrier function shown in Fig. 6.10 remains positive at all times thus satisfying the safety requirement. Fig. 6.11 shows the control inputs of example B calculated by the proposed explicit solution.

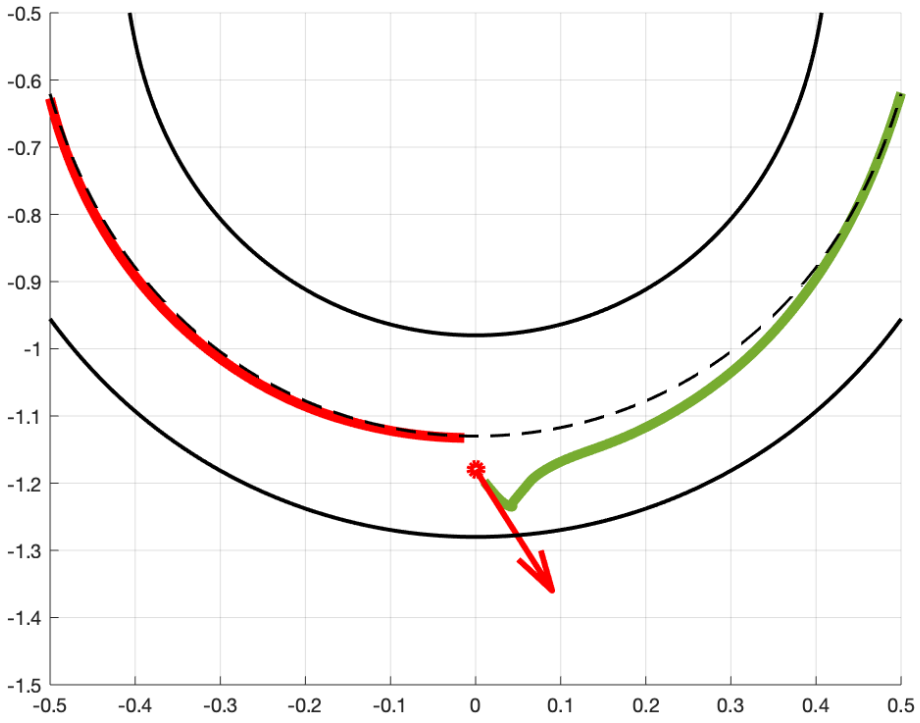


Figure 6.9: When control barrier function is active

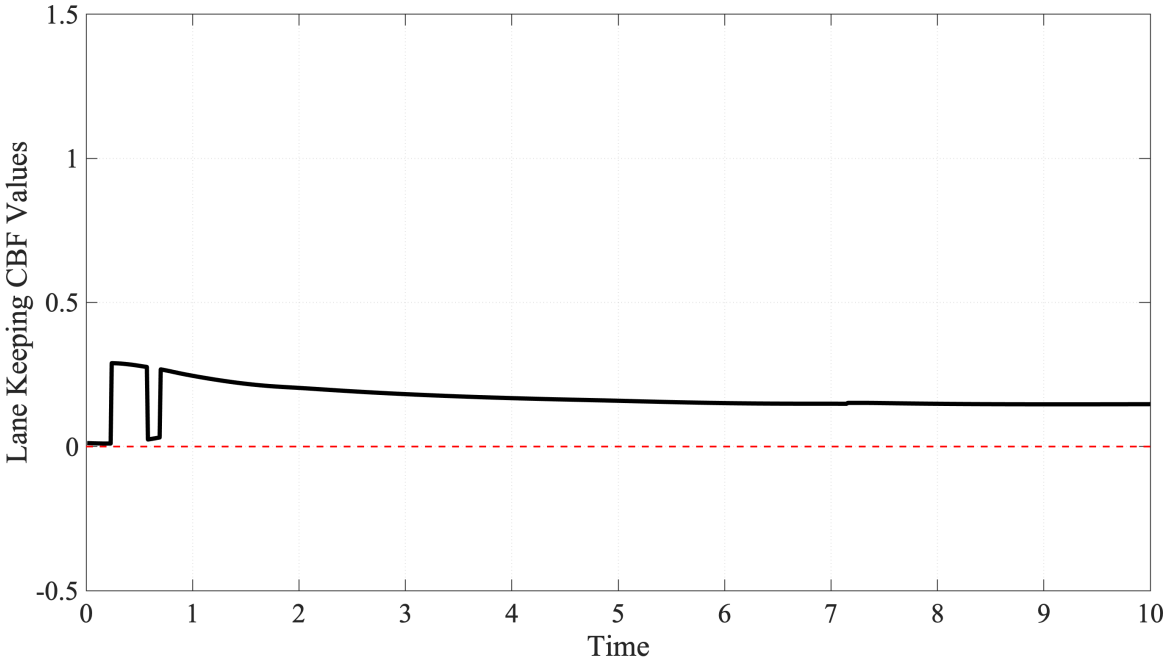


Figure 6.10: Values of control barrier function

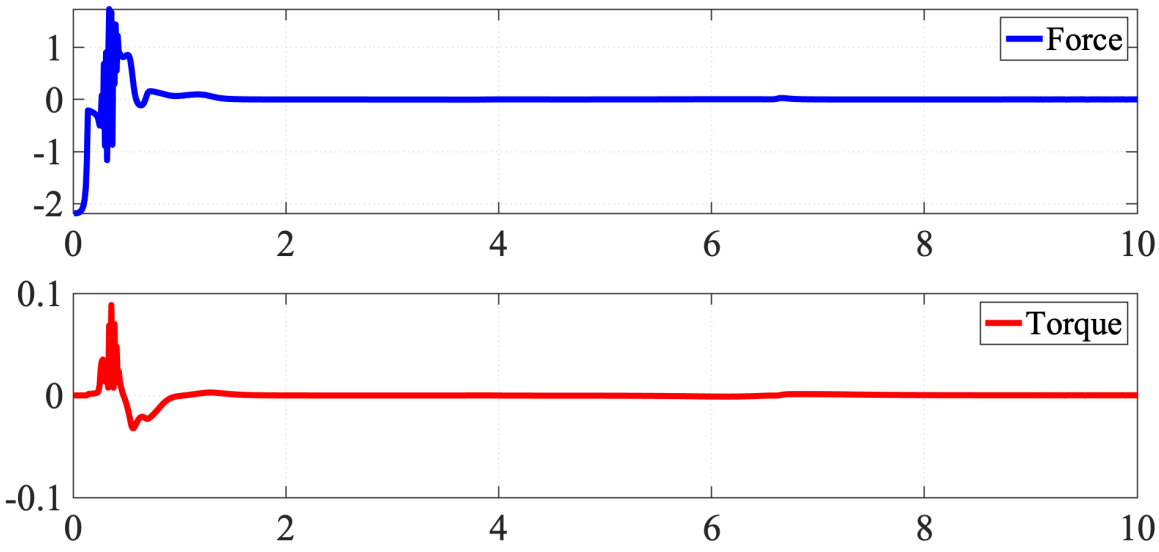


Figure 6.11: Force/torque control inputs

Chapter 7

Conclusion

In this thesis, a new control Lyapunov and barrier function method for simultaneous lane-keeping and adaptive speed control is proposed and implemented. Two simulation examples are constructed to illustrate the robot safety behavior under the proposed method.

In particular, the control Lyapunov function and two types of control barrier functions are introduced first, and a CLF-CBF-QP framework is also revisited. In this framework, relaxation parameters are considered to make QP feasible according to the previous works. This approach has benefits on implementation and can deal with potentially conflicting control objectives. However, the incompatibility will weaken the stability of the robotic system and the control performance. To overcome this challenge, a compact formulation is proposed which allows the robotic system to ensure convergence to the desired control objectives in the simultaneous lane-keeping and adaptive speed control problem. The advantage of this implementation is that the QP problem is much simpler with only one CBF constraint and one CLF constraint. Additionally, this approach can also improve the efficiency of the control input calculation by explicitly solving the associated quadratic program. To demonstrate the advantages of the proposed approach, two simulation examples are implemented in MATLAB, as seen in Chapter 6. According to the result, the control objectives represented by Lyapunov functions have a better convergence under the proposed

approach while satisfying the safety specifications.

Future works include three main parts. First of all, the robot dynamics can be noisy or inaccurate according to different types of real-world robots, therefore take noise or error into account can help generate more reliable controls. Secondly, both the robot dynamic model and barrier functions can be estimated via sensors equipped on the robot. Machine learning algorithms are very useful tools for learning robot dynamics or barrier functions. Again, to ensure satisfying the safety specifications, the control barrier function condition should be modified to encode error terms. Finally, with the above improvement, the proposed algorithm is expected to be implemented on a real-world robotic system for experimental validation.

Bibliography

[1] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431. IEEE, 2019.

[2] Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

[3] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278. IEEE, 2014.

[4] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[5] Jorge Cortes. Discontinuous dynamical systems: a tutorial on solutions, nonsmooth analysis, and stability. *arXiv preprint arXiv:0901.3583*, 2009.

[6] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315, 2017.

[7] Shao-Chen Hsu, Xiangru Xu, and Aaron D Ames. Control barrier function based quadratic programs with application to bipedal robotic walking. In *2015 American Control Conference (ACC)*, pages 4542–4548. IEEE, 2015.

[8] Mrdjan Jankovic. Combining control lyapunov and barrier functions for constrained stabilization of nonlinear systems. In *2017 American Control Conference (ACC)*, pages 1916–1922. IEEE, 2017.

[9] Felipe Nascimento Martins, Mario Sarcinelli-Filho, Teodiano Freire Bastos, and Ricardo Carelli. Dynamic modeling and adaptive dynamic compensation for unicycle-like mobile robots. In *2009 International Conference on Advanced Robotics*, pages 1–6. IEEE, 2009.

[10] Aakar Mehra, Wen-Loong Ma, Forrest Berg, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Adaptive cruise control: Experimental validation of advanced controllers on scale-model cars. In *2015 American Control Conference (ACC)*, pages 1411–1418. IEEE, 2015.

[11] Xiangru Xu, Jessy W Grizzle, Paulo Tabuada, and Aaron D Ames. Correctness guarantees for the composition of lane keeping and adaptive cruise control. *IEEE Transactions on Automation Science and Engineering*, 15(3):1216–1229, 2017.

[12] Xiangru Xu, Thomas Waters, Daniel Pickem, Paul Glotfelter, Magnus Egerstedt, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1769–1775. IEEE, 2017.