FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Advances in Lyapunov-Based Control for Multi-Robot Navigation and Safety-Critical Systems

**Matheus Ferreira dos Reis**


Programa Doutoral em Engenharia Electrotécnica e de Computadores

Supervisor: António Pedro Aguiar

August 1, 2025

# Resumo

Controlo para sistemas críticos é um tópico de crucial importância para garantir a operação segura de uma ampla gama de sistemas em diferentes áreas da engenharia, incluindo a robótica. Robustez é também um requisito crítico para garantir um alto desempenho em tarefas de navegação robótica, já que fontes de perturbações estão sempre presentes em ambientes reais. Esta tese apresenta técnicas inovadoras em teoria de controlo não-linear aplicada a navegação robótica, com foco nos temas de controlo robusto e desvio de obstáculos, introduzindo avanços significativos na teoria de controlo aplicada a sistemas críticos utilizando funções de Lyapunov (CLFs) e funções de barreira (CBFs).

Em particular, a primeira parte da tese apresenta resultados obtidos com novos controladores robustos para seguimento de caminho considerando um único veículo e controlo cooperativo de grupos de veículos a operar em um ambiente marítimo real, sujeito a diversas fontes de perturbação. A segunda parte da tese trata de aplicações em robótica para algoritmos de controlo para sistemas críticos, com foco particular no tópico de desvio de obstáculos em diferentes cenários. Finalmente, são apresentados avanços na teoria de controlo para sistemas críticos com um controlador baseado em programação quadrática, utilizando funções de Lyapunov e de barreira (CLF-CBFs), e são desenvolvidas estratégias para evitar a convergência das trajetórias do sistema em direção a pontos de equilíbrio indesejados.

**Palavras-chave:** Robótica, Seguimento de Caminho, Controlo Robusto, Controlo Cooperativo, Sistemas Críticos, Funções de Lyapunov, Funções de Barreira.

# Abstract

Safety-critical control is a topic of crucial importance for guaranteeing safe operation for a wide range of systems across different engineering areas, including robotics. Robustness is also a critical requirement for guaranteeing high performance in robotic navigation tasks, since sources of disturbances are always present in real-world environments. This thesis presents novel techniques in nonlinear control theory for multi-robot navigation, focusing particularly on the themes of robust control and collision avoidance, while introducing key theoretical advances on the general theory of nonlinear control for safety-critical systems through the framework of Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs).

In particular, the first part of the thesis presents the results obtained with novel robust path following controllers for a single vehicle and cooperative control of a team of robotic vehicles operating in a real maritime environment, subject to many sources of disturbance. The second part of the thesis addresses applications in robotics for safety-critical control schemes, focusing particularly on the topic of collision avoidance in different scenarios. Finally, advances on the theory of safety-critical control with the CLF-CBF QP-based controller are presented, where strategies for avoiding the convergence of the system trajectories towards undesirable equilibrium points are developed.

**Keywords:** Robotics, Path Following, Robust Control, Cooperative Control, Safety-Critical Systems, Lyapunov Functions, Barrier Functions.

# Acknowledgments

First, I would like to thank my advisor António Pedro Aguiar for the invaluable guidance, suggestions, technical discussions and encouragement. I am also thankful to João Borges de Souza and to the LSTS team (Underwater Systems and Technology Laboratory), with special thanks to José Pinto, Pedro Gonçalves and Maria Costa, who made possible to perform experiments with real robotic vehicles at Porto de Leixões.
I am grateful to my laboratory colleagues, R. Praveen Jain, Pedro Sequeira, Gustavo Andrade and Pallov Anand for the endless discussions, knowledge exchange, and overall fun. I also thank Rômulo Rodrigues, who convinced me to move away from MATLAB and try Python. For that, I am deeply grateful.
I am also grateful to my dear mother Angela Reis, who taught me the most important things in life.

This work was financially supported by:

i European Union's Horizon 2020 research and innovation programme MarineUAS, under the Marie Sklodowska-Curie grant agreement No 642153.

ii STRIDE - NORTE-01-0145-FEDER-000033, funded by N2020, ERDF.

iii Ph.D. Grant 2020.06795.BD from Fundação para a Ciência e Tecnologia (FCT), Portugal.

iv The Research Center for Systems and Technologies - SYSTEC, Base Funding (UIDB/00147 /2020) and Programmatic Funding (UIDP/00147/2020), all supported by national funds through FCT/MCTES (PIDDAC).

Matheus F. dos Reis

*"The greatest discoveries often lie not in finding new things,*
*but in seeing familiar things in new ways."*

Alexander Fleming

# Contents

# List of Figures

# Abbreviations

| | |
|---|---|
| AUV | Autonomous Underwater Vehicle |
| LAUV | Light Autonomous Underwater Vehicle |
| USV | Unmanned Surface Vehicle |
| UAV | Unmanned Aerial Vehicle |
| RL | Reinforcement Learning |
| DRL | Deep Reinforcement Learning |
| PF | Path Following |
| MPF | Moving Path Following |
| CMPF | Cooperative Moving Path Following |
| FOSM | First-Order Sliding Mode |
| DO | Disturbance Observer |
| RCMPF | Robust Cooperative Moving Path Following |
| RCMPF-FOSM | Robust Cooperative Moving Path Following with First-Order Sliding Mode |
| RCMPF-FOSM-DO | Robust Cooperative Moving Path Following with First-Order Sliding Mode and Disturbance Observer |
| CLF | Control Lyapunov Function |
| CBF | Control Barrier Function |
| QP | Quadratic Program |
| CLF-CBF-QP | Control Lyapunov Function and Control Barrier Function Quadratic Program |
| CLBF | Control Lyapunov Barrier Function |
| LTI | Linear Time Invariant |
| IFAC | International Federation of Automatic Control |
| SoS | Sums-of-Squares |

# Chapter 1

# Introduction

Many of the greatest technological breakthroughs of the second half of the 20th century happened due to groundbreaking advances in robotics. Highly multidisciplinary, the field of robotics ranges from electromechanical engineering to computer science, control and information theory. This vastness of different and yet profoundly connected areas unveils many great opportunities for original research, given the amount of interesting research problems still open to this day. Following the impact that the development of computers and telecommunications had on the world in the 20th century, the potential applications of autonomous robotic technology in this century will likely have a similar or even greater impact.

In this chapter, we present an introductory overview of the field of autonomous navigation in robotics, focusing on particularly challenging topics and applications, such as operational safety, reliability, and open research problems.

## 1.1 Challenges in Robotic Navigation

Today, one of the greatest successes of robotics is in industrial manufacturing: the field of robot *manipulators* comprise a multibillion-dollar industry, being responsible for the automated production processes that made possible to produce high quality everyday products for an affordable price. In the electronics industry, for example, robot arms are used to place and weld very small surface-mounted components with superhuman precision, making it possible to mass produce smartphones, notebooks, among many other types of electronic devices we use on a daily basis (see Fig. 1.1). Despite these successes, the robotic arms used in industry have a fundamental limitation, which is their *lack of mobility*. A robot manipulator has a fundamentally limited range of motion, which restricts its applicability to a limited range within its surrounding environment. That might be enough to accomplish simple repetitive tasks required of a robotic arm in an assembly line (see Fig. 1.2), but can render the robot completely useless in more general applications in which the robot must have a certain level of *mobility*, such as environmental exploration.

A *mobile robot* is able to move through the environment, effectively expanding its applicability range [74]. The physical capacity of a machine to actively explore its surrounding environment

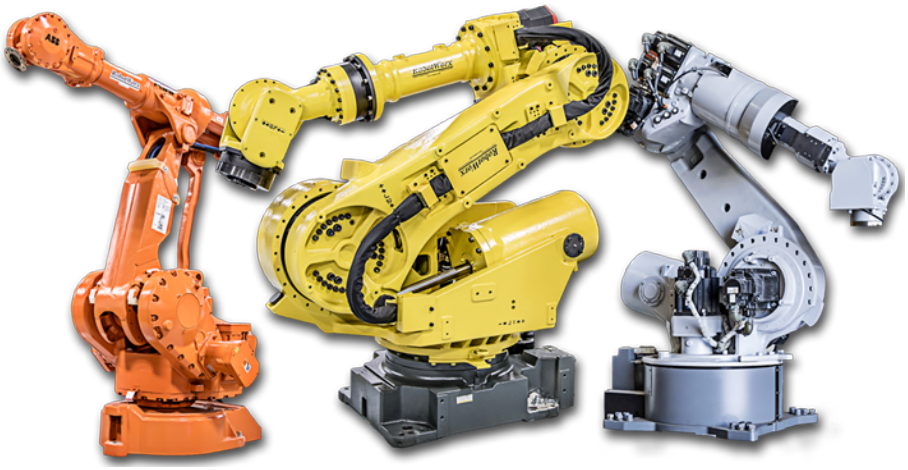


Figure 1.1: Modern industrial robot manipulators. Source: https://www.robots.com/images/general/3_Industrial_Robots.png.

Figure 1.2: Assembly line automation with robotic manipulators. Source: https://www.tm-robot.com/en/why-you-should-automate-your-assembly-lines/.

calls for some sort of automated behavior, aimed at achieving some goal or fulfilling some potentially complex task. Mobile robots are able to perform dangerous tasks that could be detrimental to human life, but have little to no chance of damaging the robot hardware, such as (i) exploring unsafe and hostile environments, inhospitable landscapes, or extraterrestrial bodies; (ii) acquiring data samples in environments with extreme levels of temperature, pressure, radiation, or harmful chemicals. Therefore, autonomy is an essential requirement: it is desirable that their operation require minimal or virtually no human intervention. Thus, an autonomous agent or *controller* must be able to generate automated behaviour for the robot that successfully accomplishes the

task without the need for human support and in an inherently safe way, that is, designed so as not to be capable of:

i harming or injuring sentient beings such as humans or other animals in any way;

ii causing damage to property, equipment (including the robot itself), resources or to the environment.

Systems whose operational failure could result in some of these undesired outcomes are known as *safety-critical systems* [42] (see Fig. 1.3), and include not only robots but also chemical, thermal and nuclear power plants, software systems and even societal, economic or political systems. Therefore, safety-critical systems are of crucial importance for many industrial sectors and production lines, where the stability of feedback-controlled systems is just as important as their capacity to provide safe behaviour under a wide variety of operational circumstances.

Figure 1.3: An example of a robot as safety-critical system. In human-robot interactions, operation must ensure that no harm is caused to humans or to equipment. Source: https://www.roboteq.com/applications/all-blogs/.

Applications for mobile robotics are growing and becoming increasingly integrated into society, with many important and pioneering applications, such as home and delivery services, transportation, logistics, search and rescue missions, environmental monitoring, surveillance and even extraterrestrial exploration (see Fig. 1.4). Recent advances in battery autonomy, electromechanical engineering and computational efficiency are combining the *dexterity* of robot manipulators with the *mobility* of mobile robots in the form of *hybrid* mobile robots. These systems have enough flexibility to move around and can also perform grasping and manipulation tasks using one or more portable manipulators, as illustrated in Fig. 1.5.

Although diverse and seemingly unrelated, many of these robotic applications share common ground in specific problems that must be addressed to achieve higher levels of operational performance. Some of these problems already have solid foundations in literature; however, many are still open problems and are not yet considered to be mature fields on their own. Moreover, as the

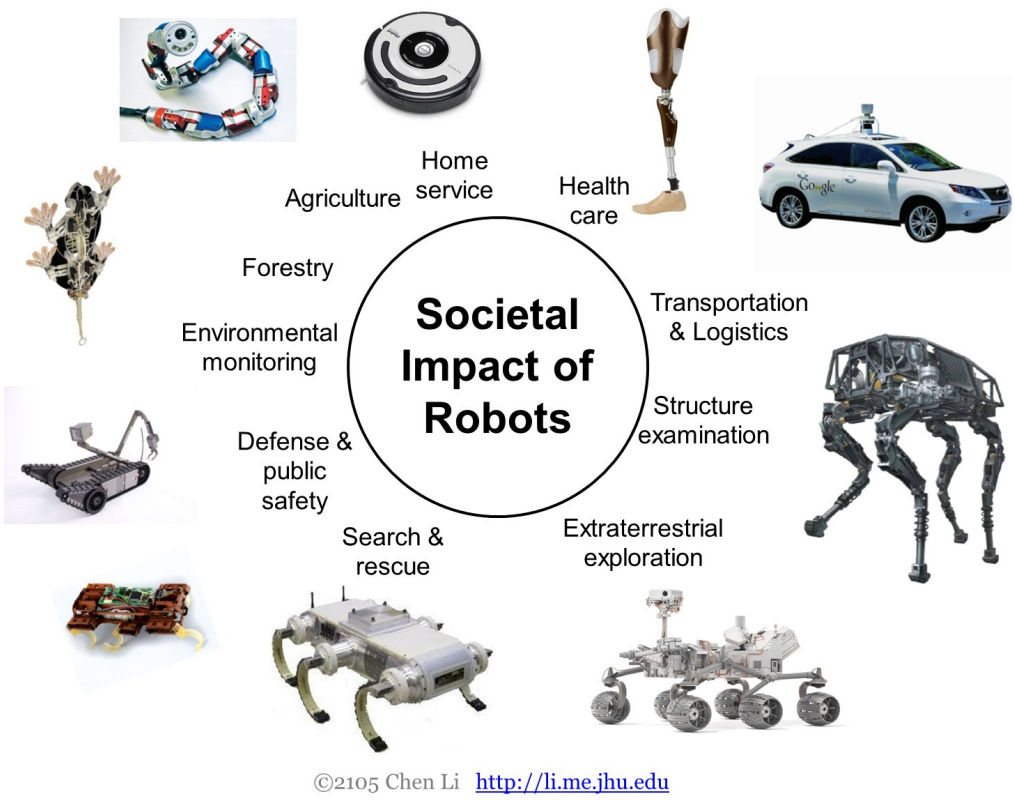



Figure 1.4: Impact of the autonomous mobile robotic technology on society and the many types of existing robot configurations. Source: https://li.me.jhu.edu/.


Figure 1.5: Some modern solutions on hybrid mobile robots. Source: https://youtu.be/7nNZriSZvVc?list=PLMRaVOnUvzLJBqTozmMNAbtLPrGozwpL7.

technology progresses, new applications start to arise and new research problems are proposed. As an example, much of the literature in the field of mobile robotics is concerned with the problems of (i) robot and multi-robot navigation, (ii) localization and mapping, (iii) motion planning, and (iv) obstacle avoidance [59]. More recently, the topics of: (v) cooperation and coordination, (vi) artificial cognition and (vii) learning and adapting through experience are starting to be addressed in robotics, especially considering the recent advances in A.I. technology through the development of generative A.I. models and Large Language Models (LLMs), such as GPT-4 [77]. In this thesis, we focus on the problems of (i) multi-robot navigation, (ii) robotic cooperation, and (iii) collision avoidance with *formal safety guarantees*.

In order to guarantee high performance in robotic navigation tasks, robust control is a particularly crucial topic due to the varied sources of uncertainties present in the dynamic environments in which robots must operate. In these scenarios, robots often encounter disturbances such as sensor noise, model inaccuracies and unexpected forces and torques from external sources that can interfere with their movement, posing challenges to their controller performance. Traditional control approaches may not be able to handle such uncertainties effectively, leading to degraded performance or even operational failure. Robust control techniques allow for the design of navigation systems that can adapt to varying conditions, ensuring stability and precision in the robot's navigation even in the presence of unforeseen challenges. This is especially important in applications such as autonomous vehicles, drones, and mobile robots, where safety, reliability, and real-time responsiveness are paramount. By focusing on minimizing the impact of uncertainties, robust control strategies enhance the robot's ability to navigate complex environments reliably, making it an essential tool for achieving satisfactory performance.

Therefore, both the need for robust control and safety-critical control strategies in robotic navigation tasks arise from the inherent unpredictability and complexity of real-world environments. Robots are often deployed in dynamic, unstructured settings where they must interact with humans, navigate through obstacles, and adapt to environmental changes such as weather, terrain, or unforeseen disturbances. In these contexts, ensuring that a robot can reliably perform motion tasks without compromising safety is paramount. Traditional control methods may struggle to maintain performance when faced with model inaccuracies, sensor noise, or external forces, which could lead to navigation failures or accidents. Robust control strategies address these challenges by designing controllers that can tolerate uncertainties and disturbances, ensuring that robots continue to operate safely and effectively. Since autonomous vehicles, drones, or medical robots can all be considered as safety-critical systems, the ability to guarantee stable and predictable behavior under a wide range of conditions is essential. Hence, the integration of robust control and safety-critical strategies is crucial for the reliable and safe operation of robots in real-world navigation tasks.

## 1.2 Main Contributions

In general, this thesis presents novel results on:

i Robust Moving Path Following (MPF) control framework for single and multi-robot navigation;

ii Safety-critical control applications in robot navigation, especially on the topic of collision avoidance with single and multiple robots;

iii Safety-critical control theory, focusing on the framework of Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs) using Quadratic Programs (QPs).

Chapter 3 presents key contributions to robust MPF control theory for robotic navigation with single and multiple vehicles. The results presented in this chapter were published in the following journal papers:

- Matheus F Reis, R Praveen Jain, A Pedro Aguiar, and João Borges de Sousa. Robust moving path following control for robotic vehicles: Theory and experiments. *IEEE Robotics and Automation Letters*, 4(4):3192–3199, 2019 [71]

- Matheus F. Reis, R. Praveen Jain, A. Pedro Aguiar, and João Borges de Sousa. Robust cooperative moving path following control for marine robotic vehicles. *Frontiers in Robotics and AI*, 6, 2019 [70]

These papers present novel Moving Path Following (MPF) [71] and Cooperative Moving Path Following (CMPF) [70] controllers shown in Section 3.2, contributing to the literature in robotic navigation by proposing robust control strategies for compensating disturbances in MPF and CMPF tasks with robots in real-world environments, and also by validating the proposed methods by means of real experiments.

Chapter 4 presents contributions on the subject of safety-critical control for obstacle avoidance using the framework of Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs), focusing on three distinct robotic applications, namely: (i) safe control of marine vessels, (ii) safe path following control for multiple autonomous vehicles, and (iii) deadlock-free cooperative safe path following control. These results were published in the following conference papers, respectively:

- Matheus F Reis, Gustavo Andrade, Francisco Neves, Paulo Silva, Rômulo T Rodrigues, and A Pedro Aguiar. A ROS implementation of the situational awareness and maneuvering systems for an autonomous marine vessel. In *OCEANS 2021: San Diego–Porto*, pages 1–9. IEEE, 2021 [67]

- Matheus F. Reis, Gustavo A. Andrade, and A. Pedro Aguiar. Safe autonomous multi-vehicle navigation using path following control and spline-based barrier functions. In Lino Marques, Cristina Santos, José Luís Lima, Danilo Tardioli, and Manuel Ferre, editors, *Robot 2023: Sixth Iberian Robotics Conference*, pages 297–309, Cham, 2024. Springer Nature Switzerland [68]

- Matheus F Reis, Pallov Anand, and A Pedro Aguiar. Cooperative path following with collision avoidance guarantees using control lyapunov and barrier functions. In *APCA International Conference on Automatic Control and Soft Computing*, pages 181–193. Springer, 2022 [66]

The work [67] presents an application of a *safety-filter* controller, adding an extra layer of safety to a path-following task with multiple obstacles (Section 4.1). In [68], we present a novel type of barrier using B-splines to achieve safe navigation for a group of autonomous vehicles in a simulated traffic lane (Section 4.2). The work [66] introduces a novel technique to solve the problem of deadlocks in cooperative path-following (PF) control with intersecting paths, where vehicles tend to get stuck at the path intersection to avoid collisions (Section 4.3). This topic introduces the concept of undesirable equilibrium points in the CLF-CBF formulation for safety-critical control, which will be further developed in Chapter 5.

Chapter 5 contributes to the theory of CBF-based control by providing a novel and comprehensive stability analysis for different types of equilibrium points arising in the closed-loop system with two CBF-based formulations for safety-critical control, namely (i) the safety-filter QP and (ii) the CLF-CBF QP. The analysis considers multiple CBFs with possibly intersecting unsafe sets and applies to the general class of control-affine nonlinear systems (Section 5.1). For the CLF-CBF QP controller, different strategies for solving the problem of undesirable stable equilibrium points are proposed by leveraging the newly developed theory. These strategies focus on modifying the CLF shape to induce bifurcations on the phase portrait of the closed-loop system, with the goal of either (i) removing undesirable equilibrium points or (ii) altering their stability properties. Three journal papers were produced on this subject, namely:

- M. F. Reis, A. P. Aguiar, and P. Tabuada. Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria. *IEEE Control Systems Letters*, 5(2):731–736, 2021 [64]

- Matheus F. Reis and A. Pedro Aguiar. On the stability of undesirable equilibria in the quadratic program framework for safety-critical control. `https://arxiv.org/abs/2402.08027`, 2024, available at `https://arxiv.org/abs/2402.08027` [65]

- Matheus F. Reis, José P. Carvalho, and A. Pedro Aguiar. A unified stability analysis of safety-critical control using multiple control barrier functions. `https://arxiv.org/abs/2503.15823`, 2025, available at `https://arxiv.org/abs/2503.15823` [69]

The published journal paper [64] sheds light on the problem of undesirable stable equilibria (deadlocks) in a formal way: (i) conditions for the existence of equilibrium points other than the origin are derived for the case of driftless nonlinear systems with a full-rank input matrix; (ii) a sufficient condition for their instability is derived for the special case of the integrator system (see Corollary 5.1.6.1); and (iii) a novel technique that prevents the system state from evolving towards undesirable equilibria is proposed (see Section 5.2). The work [65] (accepted for publication at IFAC Automatica) contributes to the literature on CLF-CBF-based safety-critical control by: (i)

presenting existence and stability conditions for the formation of equilibrium points for general control affine nonlinear systems (see Section 5.1), under the assumption of CBFs with disjoint unsafe sets; (ii) proposing the concept of CLF-compatibility (see Section 5.1.3), regarding the problem of obtaining a CLF that does not generate undesirable stable equilibrium points in the closed-loop system; and (iii) providing an optimization-based method for computing compatible CLFs, for the special class of Linear Time-Invariant (LTI) and driftless full rank nonlinear systems with a quadratic CLF and CBFs with disjoint unsafe sets (see Section 5.3). The work [69] (currently under review for journal publication) extends the stability analysis of [65] to the safety-filter QP controller as well and considers the possibility of CBFs with intersecting boundaries; in particular, it shows that in this case, equilibrium points may occur at boundary intersections of the CBFs, and results concerning their stability properties are properly generalized.

## 1.3 Thesis Organization

This thesis is organized as follows.

**Chapter 2** introduces the necessary theoretical background and provides a comprehensive literature review, covering the current state of the art on the main themes addressed in this thesis. The focus is on recent developments and the current challenges in the fields of multi-robot navigation and safety-critical systems.

**Chapter 3** presents novel results on robust Moving Path Following (MPF) and Cooperative Moving Path Following (CMPF) control strategies for robotic vehicles.

i Section 3.1 formulates the problem of robust Cooperative Moving Path Following (CMPF).

ii Section 3.2 introduces a theory for the design of robust CMPF controllers.

iii Section 3.3 presents and discusses the results obtained with the proposed robust controllers for a navigation task conducted in a maritime environment using real robotic vehicles. These results were published in [71] and [70], respectively.

**Chapter 4** presents three applications of the theory of safety-critical control using Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs) for robotic navigation tasks with guaranteed collision avoidance.

i Section 4.1 presents an application of safe Path-Following navigation for autonomous vessels, as published in [67].

ii Section 4.2 presents an application of safe Path-Following using spline-based paths for autonomous vehicles, as published in [68].

iii Section 4.3 presents an application of safe Cooperative Path Following for robotic vehicles using CLFs and CBFs, with a focus on deadlock avoidance, as published in [66].

**Chapter 5** introduces key theoretical developments in the field of safety-critical control within the CBF-based QP framework. Particularly, Section 5.1 presents existence conditions and stability properties for all types of equilibrium points arising with two CBF-based controllers: (i) the safety-filter QP and (ii) the CLF-CBF QP, considering multiple CBFs with possibly intersecting unsafe sets. For the CLF-CBF controller, through the notion of *CLF compatibility*, we demonstrate how the developed theory can be applied to achieve quasi-global stabilization of the origin for certain classes of systems under mild assumptions.

i Section 5.1 presents the general theory on the existence and stability for all types of equilibrium points arising with the safety-filter and the CLF-CBF controllers, both implemented using a Quadratic Program (QP) (available online at [69], currently under review for publication).

ii Section 5.2 presents a technique for achieving CLF compatibility using non-radial CLFs, as published in [64].

iii Section 5.3 presents an algebraic theory for quadratic CLF compatibility for certain classes of systems - namely, Linear Time-Invariant (LTI) and driftless full-rank nonlinear systems - and introduces a technique for computing compatible CLFs (available online at [65], accepted for publication in IFAC Automatica).

**Chapter 6** presents concluding remarks and relevant discussions on the results achieved throughout the thesis. It also outlines potential directions for future research within the topics explored in this work.

# Chapter 2

# Theoretical Background and Literature Review

In this chapter, we present the mathematical notation and conventions used throughout this document. Furthermore, a thorough review of the literature of the subjects addressed in this thesis is presented, with the aim of clarifying the state-of-the-art concepts and ideas in each field. We also introduce the necessary theoretical background for the topics covered. More precisely:

- Section 2.1 presents the mathematical notation and conventions used throughout this thesis.
- Section 2.2 presents the theory of Lyapunov-based control for nonlinear systems, which is widely used in robotics and countless other engineering areas, being of critical importance throughout this thesis.
- Section 2.3 presents some of the recent works and strategies for robotic navigation, focusing on the Moving Path-Following (MPF) control strategy for single robot vehicles and cooperative control for multiple vehicles.
- Section 2.4 presents the theory of stabilization and safety using Control Lyapunov Functions (CLFs) and Control Barrier Functions (CBFs), respectively, in the context of safety-critical systems, which will be addressed in Chapters 4 and 5.

## 2.1 Notation

In this section, we seek to standardize the mathematical notation and conventions used throughout this thesis.

The fields of real and complex numbers are $\mathbb{R}$ and $\mathbb{C}$, respectively. Given a vector $v \in \mathbb{R}^n$, $[v]_k \in \mathbb{R}$ is its $k$-th component. Given a matrix $A \in \mathbb{R}^{n\times m}$, $[A]_{ij} \in \mathbb{R}$ denotes its $i$-th row, $j$-th column component and $[A]_k \in \mathbb{R}^n$ denotes its $k$-th column. The null space of $A \in \mathbb{R}^{n\times m}$ is given by $\mathcal{N}(A) = \{v \in \mathbb{R}^m \mid Av = 0\}$. The spectrum of a real square matrix $A \in \mathbb{R}^{n\times n}$ is the set of

eigenvalues of $A$:

$$\sigma(A) = \{\lambda \in \mathbb{C} \,|\, \det(\lambda I_n - A) = 0\}$$

A square matrix is said to be Hurwitz stable if all its eigenvalues lie in the left-half plane, that is, if they all have strictly negative real part. The determinant of a square matrix $A \in \mathbb{R}^{n\times n}$ is $\det A$ or $|A|$ and its trace is $\mathrm{Tr}(A)$. Its Frobenius norm is $\|A\|_{\mathscr{F}} = \sqrt{\mathrm{Tr}(A^\mathsf{T}A)}$, and its adjoint matrix is $\mathrm{adj}\, A \in \mathbb{R}^{n\times n}$ with the property $A\,\mathrm{adj}\, A = |A|I_n$, where $I_n \in \mathbb{R}^{n\times n}$ is the $n\times n$ identity matrix. The set $\mathbb{SO}(n) \subset \mathbb{R}^{n\times n}$ is the special orthogonal group of dimension $n$, also known as the *rotation* group, representing rotational transformations in the $n$-th dimensional Euclidean space and consisting of all orthogonal matrices $R$ with $R^\mathsf{T}R = I_n$ and $\det R = 1$. The set $\mathfrak{so}(n)$ consists of its corresponding Lie algebra, which has the structure of a $\frac{1}{2}n(n-1)$-dimensional vector space and is composed of skew-symmetric $n\times n$ matrices, that is, if $S \in \mathfrak{so}(n)$, $S + S^\mathsf{T} = 0$.

The group of real symmetric matrices is denoted by $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$. All real symmetric matrices $A \in \mathbb{S}^n$ have real spectrum (that is, $\sigma(A) \subset \mathbb{R}$) and are diagonalizable. The minimum and maximum eigenvalues of $A \in \mathbb{S}^n$ are given by $\sigma_{\min}(A)$, $\sigma_{\max}(A)$, respectively. The set of symmetric singular matrices of dimension $n$ is denoted by $\mathbb{S}^n_0$. That is, if $A \in \mathbb{S}^n_0$, then $0 \in \sigma(A)$. The positive and negative semi-definite cones of real symmetric matrices of dimension $n$ are denoted by

$$\mathbb{S}^n_{\geq 0} = \{A \in \mathbb{S}^n \,|\, \sigma(A) \subset \mathbb{R}_{\geq 0}\},$$
$$\mathbb{S}^n_{\leq 0} = \{A \in \mathbb{S}^n \,|\, \sigma(A) \subset \mathbb{R}_{\leq 0}\},$$

respectively. The sets $\mathbb{S}^n_{>0} = \mathbb{S}^n_{\geq 0} \setminus \mathbb{S}^n_0 = \{A \in \mathbb{S}^n \,|\, \sigma(A) \subset \mathbb{R}_{>0}\}$ and $\mathbb{S}^n_{<0} \setminus \mathbb{S}^n_0 = \{A \in \mathbb{S}^n \,|\, \sigma(A) \subset \mathbb{R}_{<0}\}$ are the sets of strictly positive and strictly negative definite matrices, respectively. The cone of semidefinite symmetric matrices of dimension $n$ is given by $\mathbb{S}^n_{\geq 0} \cup \mathbb{S}^n_{\leq 0}$. Then, *semidefiniteness* of $A$ is denoted by $A \geq 0$ (that is, $A \in \mathbb{S}^n_{\geq 0}$) or $A \leq 0$ (that is, $A \in \mathbb{S}^n_{\leq 0}$). Similarly, *definiteness* of $A$ is denoted by $A > 0$ (that is, $A \in \mathbb{S}^n_{>0}$) or $A < 0$ (that is, $A \in \mathbb{S}^n_{<0}$).

Given two vectors $v, w \in \mathbb{R}^n$, the positive semidefinite inner product induced by a positive semidefinite matrix $A \in \mathbb{S}^n_{\geq 0}$ is given by $\langle v, w\rangle_A = \langle w, v\rangle_A = v^\mathsf{T}Aw$. The usual inner product in $\mathbb{R}^n$ is then $\langle v, w\rangle = \langle v, w\rangle_{I_n}$, where $I_n > 0$ is the $n$-th dimensional identity matrix. The norm of $v \in \mathbb{R}^n$ induced by this inner product is simply $\|v\|_A = \sqrt{\langle v, v\rangle_A}$. The usual norm induced by the usual inner product is then $\|v\| = \|v\|_{I_n} = \sqrt{\langle v, v\rangle}$. The orthogonal complement of a subspace $\mathscr{W}$ is denoted by $\mathscr{W}^\perp$, with the notion of orthogonality dependent upon the inner product $\langle\cdot,\cdot\rangle_G$. That is, for all $w_1 \in \mathscr{W}$ and $w_2 \in \mathscr{W}^\perp$, $\langle w_1, w_2\rangle_G = 0$. The spam of a set of $r$ vectors $v_1, \cdots, v_r \subset \mathbb{R}^n$ is defined as the set of all of their linear combinations, as $\mathrm{span}\{v_1, \cdots, v_r\} = \{a_1v_1 + \cdots + a_rv_r \,|\, a_1, \cdots, a_r \in \mathbb{R}\}$.

The ball centered on a point $x_0 \in \mathbb{R}^n$ with radius $r \in \mathbb{R}_{\geq 0}$ is simply denoted by the closed set $\mathscr{B}_r(x_0) = \{x \in \mathbb{R}^n \mid \|x - x_0\| \leq r\}$.

A scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be of (differentiability) class $C^k$ if all of its $k$-th order partial derivatives exist and are continuous. Consider the class $C^2$ function $f : \mathbb{R}^n \to \mathbb{R}$:

i Its *gradient* is defined as the vector-valued function $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$[\nabla f(x)]_k = \frac{\partial f(x)}{\partial x_k} = \partial_k f(x)\,,$$

where $\partial_k$ denotes partial differentiation with respect to the $k$-th component of $x \in \mathbb{R}^n$.

ii Its *Hessian* matrix is defined as the matrix-valued function $H_f : \mathbb{R}^n \to \mathbb{S}^n$ such that

$$[H_f(x)]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

The symmetry of $H_f$ is ensured by the equality of mixed partials, which holds since $f$ is a class $C^2$ function.

The superlevel, sublevel and level sets of $f$ are, respectively,

$$\begin{aligned}
\mathscr{L}_c^+(f) &= \{x \in \mathbb{R}^n : f(x) \geq c\}\,,\\
\mathscr{L}_c^-(f) &= \{x \in \mathbb{R}^n : f(x) \leq c\}\,,\\
\mathscr{L}_c(f) &= \mathscr{L}_c^+(f) \cap \mathscr{L}_c^-(f) = \{x \in \mathbb{R}^n : f(x) = c\}\,,
\end{aligned}$$

for a given constant $c$. If $c = 0$, we simply write $\mathscr{L}_0^+(f) = \mathscr{L}^+(f)$, $\mathscr{L}_0^-(f) = \mathscr{L}^-(f)$ and $\mathscr{L}_0(f) = \mathscr{L}(f)$, for simplicity. Define $\mathscr{T}_p f \subset \mathbb{R}^{n-1}$ as the tangent space to the level set $\mathscr{L}_p(f)$ of function $f$ at a point $p \in \mathbb{R}^n$. Given a vector field $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, we use the notation $L_g f$ to denote the Lie derivative of function $f$ along the vector field $g$, that is, $L_g f = \nabla f^{\mathsf{T}} g \in \mathbb{R}^m$, respectively.

Let $\mathscr{Z} \subset \mathbb{R}^n$ be a subspace of $\mathbb{R}^n$ with dimension $\dim \mathscr{Z} = r$ generated by an orthogonal basis $\mathscr{B} = \{z_1, \cdots, z_r\}$ with respect to the inner product $\langle \cdot, \cdot \rangle_G$ induced by a positive semidefinite matrix $G = G^{\mathsf{T}} \geq 0$. That means that $\langle z_i, z_j \rangle_G = z_i^{\mathsf{T}} G z_j = 0$ for all $i \neq j$. Collecting the basis vectors of $\mathscr{B}$ in matrix $Z = [\,z_1 \ \cdots \ z_r\,] \in \mathbb{R}^{n\times r}$ , define the operator:

$$\mathsf{P}_Z = I_n - GZN_1Z^{\mathsf{T}} \in \mathbb{R}^{n\times n} \tag{2.1}$$

where $N_1 > 0$ is the first element from the infinite sequence of positive definite matrices

$$\{N_1, N_2, \cdots, N_k, \cdots\} \subset \mathbb{S}^r_{>0}$$

defined by the recursive relation $N_{k+1} = 2I_r - N_k Z^{\mathsf{T}} G Z N_k > 0$, $k \in \mathbb{N}$. Notice that, since the columns of $Z \in \mathbb{R}^{n\times r}$ form an orthogonal basis for $\mathscr{Z}$ with respect to the inner product $\langle \cdot, \cdot \rangle_G$, $Z^{\mathsf{T}} G Z \in \mathbb{R}^{r\times r}$ is a diagonal matrix, which means that $N_k$ is diagonal for all $k \in \mathbb{N}$. The operator

$\mathsf{P}_Z$ has the following properties:

$$\mathsf{P}_Z^k = I_n - GZN_kZ^\mathsf{T} \tag{2.2}$$
$$\mathsf{P}_Z^kGZ = GZ(I_r - N_kZ^\mathsf{T}GZ) \tag{2.3}$$
$$Z^\mathsf{T}\mathsf{P}_Z^k = (I_r - Z^\mathsf{T}GZN_k)Z^\mathsf{T} \tag{2.4}$$
$$\mathsf{P}_Z^kGw = Gw \quad \forall Gw \in \mathscr{Z}^\perp \tag{2.5}$$
$$w^\mathsf{T}\mathsf{P}_Z^k = w^\mathsf{T} \quad \forall Gw \in \mathscr{Z}^\perp \tag{2.6}$$

In (2.2), the expression for the $k$-th power of $\mathsf{P}_Z$ is similar to the expression of $\mathsf{P}_Z$, differing only by matrix $N_k > 0$. In fact, if the orthogonal basis $\mathscr{B}$ is also *normal*, that is, all its vectors are unitary with respect to the inner product $\langle\cdot,\cdot\rangle_G$, then $Z^\mathsf{T}GZ = I_r$. If additionally $N_1 = I_r$, all elements of the infinite sequence $\{N_1, N_2, \cdots, N_k, \cdots\}$ are equal to $I_r$, and $\mathsf{P}_Z$ reduces to an *oblique projection* matrix $\mathsf{P}_Z = I_n - GZZ^\mathsf{T}$ into $\mathscr{Z}^\perp$, with (2.2) yielding the usual defining property for projection matrices $\mathsf{P}_Z^k = \mathsf{P}_Z$. Moreover, (2.3) implies $\mathsf{P}_ZGZ = 0$ and (2.4) implies $Z^\mathsf{T}\mathsf{P}_Z = 0$. With the usual inner product (that is, $G = I_n$), $\mathsf{P}_Z$ reduces further to an standard *orthogonal* projection matrix $\mathsf{P}_Z^k = I_n - ZZ^\mathsf{T}$ into $\mathscr{Z}^\perp$, with the additional property $\mathsf{P}_Z = \mathsf{P}_Z^\mathsf{T}$.

Below, we define the notion of comparison functions of classes $\mathscr{K}$, $\mathscr{K}_\infty$ and $\mathscr{KL}$ [41], Section 4.4, which are useful in the topic of Lyapunov stability theory and will be important in CLF-CBF theory introduced in Section 2.4 and further explored in Chapter 5.

**Definition 2.1.1** (Class $\mathscr{K}$)**.** A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathscr{K}$ if: (i) it is strictly increasing and (ii) it is such that $\alpha(0) = 0$. It is said to belong to class $\mathscr{K}_\infty$ if $a = \infty$ and $\alpha(r) \to \infty$ as $r \to \infty$, and to belong to *extended* class $\mathscr{K}_\infty$ if it belongs to class $\mathscr{K}_\infty$ and if its domain and codomain are extended to $\alpha : \mathbb{R} \to \mathbb{R}$.

**Definition 2.1.2** (Class $\mathscr{KL}$)**.** A continuous function $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ is said to belong to class $\mathscr{KL}$ if: (i) for each fixed $s$, the function $\beta(r, s)$ is of class $\mathscr{K}$, and (ii) for each fixed $r$, the function $\beta(r, s)$ is decreasing with respect to $s$ and is such that $\beta(r, s) \to 0$ for $r \to 0$.

## 2.2 Lyapunov-based Control

This section is devoted to the theory of Lyapunov-based control for nonlinear systems. All the control strategies, algorithms and techniques developed in this work are directly dependent on it.

Consider the controlled nonlinear system with dynamics given by

$$\dot{x} = f(x, u) \tag{2.7}$$

where $f : \mathbb{D} \times \mathbb{U} \to \mathbb{R}^n$ is a Lipschitz vector-valued function known as the *system flow*, $x \in \mathbb{D} \subset \mathbb{R}^n$ is the system state and $u \in \mathbb{U} \subset \mathbb{R}^m$ is the control input. Given a certain initial condition $x(0) \in \mathbb{D}$ for the system state and a certain control input $u \in \mathbb{U}$, the trajectories of the system state $x(t) \in \mathbb{D}$ are the solutions of the Ordinary Differential Equation (ODE) (2.7).

The *static* feedback stabilization problem for (2.7) consists of designing a control law

$$u^\star = u(x) \tag{2.8}$$

such that the solutions of the *closed-loop* autonomous system

$$\dot{x} = f_{cl}(x) = f(x, u^\star) \tag{2.9}$$

converge to a desired state configuration $x_d(t) \in \mathbb{D}$ and stay close to $x_d(t)$ as $t \to \infty$. Usually, this condition is expressed in terms of a *state error* $e(t) = x(t) - x_d(t)$, which is desired to converge to the origin $0 \in \mathscr{D}$ as time increases. That is, the controller (2.8) is designed in such a way that $e(t) \to 0$ as $t \to \infty$ or remains close to zero as $t \to \infty$. The latter condition can be formally defined as $\exists t_1, \varepsilon \geq 0$ such that $e(t) \in \mathscr{B}_\varepsilon(0)\ \forall t \geq t_1$.

This requirement usually means that the desired configuration $x_d$ must be an equilibrium point of (2.9). In fact, let $x_e \in \mathbb{D}$ be an arbitrary equilibrium point of (2.9): that is, it satisfies $f(x_e) = 0$. Lyapunov stability theory provides formal methods for determining the stability properties of such points, which is fundamental for the analysis of the solutions of (2.9) and therefore, for the design of feedback controllers (2.8).

**Definition 2.2.1** (Stability [41])**.** An equilibrium point $x_e \in \mathbb{D}$ of the dynamic system (2.9) is

1. Stable if, for each $\varepsilon > 0$, there exists an $\delta = \delta(\varepsilon) > 0$ such that

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon\,, \quad \forall t \geq 0$$

2. Unstable if not stable.

3. Asymptotically stable if stable and if $\delta$ can be chosen such that

$$\|x(0)\| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0$$

The stability of the equilibrium points of a dynamic system is a topic of great importance, since it is often desired to design the state-feedback controller (2.8) in such a way that desired configuration $x_d$ (also known as goal state) is stable. In the context of robotic control, the goal state is usually a position or orientation for the robot (also known as pose) in its configuration space, or its velocities expressed in some reference frame [73]. However, it can also represent a more general desired configuration, such as a specific geometric pattern in a formation with multiple robotic vehicles.

**Theorem 2.2.1** (Lyapunov Stability)**.** Let $x_e \in \mathbb{D} \subset \mathbb{R}^n$ be an equilibrium point of the dynamic system $\dot{x} = f(x)$, and let $V : \mathbb{D} \to \mathbb{R}_{\geq 0}$ be a class $\mathscr{C}^1$ function such that

$$V(x_e) = 0 \text{ and } V(x) > 0 \text{ in } \mathbb{D} \setminus \{x_e\} \tag{2.10}$$

$$\dot{V}(x) \leq 0 \text{ in } \mathbb{D} \tag{2.11}$$

Then, $x_e$ is stable. Moreover, if $\dot{V}(x) < 0$ in $\mathbb{D}$, then $x_e$ is asymptotically stable.

*Proof.* The proof can be found in [41], Theorem 4.1. □

Roughly speaking, Theorem 2.2.1 means that, if a function $V$ with properties (2.10)-(2.11) exist, all trajectories $x(t)$ starting sufficiently close to $x_e$ remain close to it, without ever diverging. In particular, if $x_e$ is asymptotically stable, all trajectories $x(t)$ starting sufficiently close to $x_e$ also converge to it as $t$ increases. The function $V(x)$ satisfying the conditions of Theorem 2.2.1 is called a *Lyapunov* function, and it is useful not only for demonstrating stability of the equilibrium points of a dynamic system, but also as a designing tool for nonlinear control, as extensively studied in literature. Its development by Aleksandr Lyapunov in his thesis "The General Problem of Stability of Motion" in 1892 [49] had strong inspiration from the concept of energy present in physics: much alike the energy function for a physical system, a Lyapunov function is minimized as the system approaches an stable state. However, Lyapunov functions are much more general then physical energy functions and present clear advantages: (i) often, finding the true energy function for complicated physical systems can be quite challenging, with Lyapunov functions presenting an easier method for asserting their stability properties; (ii) Lyapunov functions can be used to study the stability of non-physical systems as well, such as economical, biological or essentially any general dynamical system whose dynamics that can be expressed using differential equations.

Another important notion of stability is defined below.

**Definition 2.2.2** (Input-to-State Stability [41])**.** The system (2.7) is said to be Input-to-State Stable (ISS) if there exists a class $\mathscr{KL}$ function $\beta$ and a class $\mathscr{K}$ function $\gamma$ such that for any initial state $x(0)$ and any bounded input $u(t)$, the solution $x(t)$ exists for all $t \geq 0$ and satisfies

$$\|x(t)\| \leq \beta(\|x(0)\|, t) + \gamma\left(\sup_{0 \leq \tau \leq t} \|u(\tau)\|\right) \tag{2.12}$$

The inequality in Definition 2.2.2 guarantees that for any bounded input $u(t)$, the state $x(t)$ will also be bounded. Furthermore, as the time $t$ increases, the state $x(t)$ will be ultimately bounded by the class $\mathscr{K}$ function on the second term of the right-hand side of (2.12). Input-to-State stability is an important concept for demonstrating results in robust control theory, and will be addressed at Chapter 3.

A somewhat dual concept to Lyapunov stability is that of *instability* of equilibrium points. While important from a theoretical standpoint, proving or even enforcing instability of equilibrium points in a dynamic system is not a topic widely studied in nonlinear system control literature. Instead, the wide number of applications of the Lyapunov stability methods developed during the last few decades focus almost exclusively on nonlinear system stabilization with respect to an equilibrium point, set or trajectory [43]. However, some applications such as generation of oscillating or chaotic behavior in mechanical systems [20, 21] require the use of theoretical methods for instability analysis of equilibrium points.

**Theorem 2.2.2** (Chetaev Instability [41])**.** Let $x_e \in \mathbb{D} \subset \mathbb{R}^n$ be an equilibrium point of the dynamic system $\dot{x} = f(x)$. Let $V : \mathbb{D} \to \mathbb{R}_{\geq 0}$ be a class $\mathscr{C}^1$ function with $V(x_e) = 0$ and such that there exists $\varepsilon > 0$ such that the open set $\mathscr{V}_e = \{x \in \mathscr{B}_\varepsilon(x_e) \mid V(x) > 0\}$ is non-empty. Then, if $\dot{V} > 0$ for all $x \in \mathscr{V}_e$, then $x_e$ is an unstable equilibrium point.

*Proof.* The proof can be found in [41], Theorem 4.3. □

Roughly, Theorem (2.2.2) means that if there exists a function $V$ with $V(x_e) = 0$ such that both $V$ and $\dot{V}$ are positive definite in a neighborhood of $x_e$, then $x_e$ is unstable. Intuitively, it means that if the system trajectory $x(t)$ starts close to the equilibrium point, it will diverge from it. Theorem 2.2.2 will be useful in Chapter 5, where the stability properties of certain equilibrium points will be studied.

In robotics literature, instead of considering the general class of ODEs in (2.13) in its full complexity, the main focus is usually the class of *affine* nonlinear control systems, whose dynamics are described by

$$\dot{x} = f(x) + g(x)u \tag{2.13}$$

where $x \in \mathbb{R}^n$ is the system state and $u \in \mathbb{R}^m$ is the control input. The vector fields $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are locally Lipschitz. Let $u^\star(x)$ be a feedback controller such that the closed-loop system

$$\dot{x} = f_{cl}(x) := f(x) + g(x)u^\star(x) \tag{2.14}$$

is locally Lipschitz. Then, for any initial condition $x(0) \in \mathbb{R}^n$, there exists a maximum interval of existence $\mathscr{I}(x(0)) = [0, \tau_{max})$ such that $x(t)$ is the unique solution to (2.14) on $\mathscr{I}(x(0))$. Furthermore, if the system flow $f_{cl}(x)$ is forward complete, $\tau_{max} = \infty$ [41]. The model (2.13) is also useful for describing a wide range of nonlinear systems encompassing many other areas of engineering, including industrial and aerospace systems, chemical and power plants, among others.

**Definition 2.2.3** (CLF)**.** A positive definite function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a *Control Lyapunov Function* (CLF) for system (2.13) if it satisfies:

$$\inf_{u \in \mathbb{U} \subset \mathbb{R}^m} \left( \underbrace{L_f V(x) + L_g V(x)\, u}_{\dot{V}} \right) \leq -\alpha(V(x))$$

for some class $\mathscr{K}$ function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$.

Definition 2.2.3 means that if there exists a CLF for system (2.13), then there exists a set of stabilizing state-feedback controls $\mathbb{K}_{\text{clf}}(x) \subset \mathbb{R}^m$ that makes $V$ strictly decreasing everywhere outside of the CLF global minimum, denoted here as $x_0 \in \mathbb{D} \subset \mathbb{R}^n$. That is, $V(x) \geq V(x_0)$ and

$\dot{V}(x) < 0 \; \forall x \neq x_0$. This set is defined by

$$\mathbb{K}_{\text{clf}}(x) = \left\{ u \in \mathbb{R}^m : \underbrace{L_f V(x) + L_g V(x)\, u}_{\dot{V}} \leq -\alpha(V(x)) \right\}. \tag{2.15}$$

In [10], it was demonstrated that if a CLF exists for (2.13), then there exists a feedback control law $u^\star = u(x) \in \mathbb{K}_{\text{clf}}(x)$ that makes the CLF global minimum $x_0$ globally asymptotically stable. Hence, the existence of a CLF for a system guarantees that $x_0$ is asymptotic stabilizable. However, a constructive approach for designing such controller was only obtained in 1989, when [76] proposed a universal formula for smooth stabilization.

**Definition 2.2.4** (Sontag's Formula)**.** For a positive scalar $c > 0$, define the *continuous* map $\kappa_\gamma : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$:

$$\kappa(a,b) = \begin{cases} -\frac{b}{\|b\|^2}\left(a + \sqrt{a^2 + c\|b\|^4}\right) & , \quad b \neq 0 \\ 0 & , \quad b = 0 \end{cases} \tag{2.16}$$

It can be shown that, if a CLF $V$ for system (2.13) exists, applying the feedback control law

$$u^\star(x) = \kappa(L_f V(x), L_g V(x)) \tag{2.17}$$

into (2.13) yields $\dot{V} = -\sqrt{L_f V(x)^2 + L_g V(x)^4} < 0$, which, from (2.11) in Theorem 2.2.1, renders $x_0$ asymptotically stable [76]. Sontag's formula (2.16) is very useful for the design of a CLF-based controller that guarantees the stabilization of $x_0$, is smooth everywhere and continuous at $x_0$. It is also useful in other contexts, such as safety of nonlinear systems, as will be shown in Section 2.4.1.

## 2.3 Robotic Navigation

Autonomous navigation can be defined as the combination of three fundamental competences: (i) self-localization, (ii) map building and interpretation and (iii) path planning and motion control. In this section, we present the state-of-the-art for the topics covered in this thesis, concerning widely used techniques for reliable and efficient robot navigation. Motion control is the problem of generating the correct signals for the robot actuators in order to accomplish a specific motion task. It is a general area in robotics and control theory that can be applied to both robot manipulators and mobile robots. In the context of mobile robots, the task to be fulfilled by the motion control law usually consists of steering the robot towards a given point, track a trajectory, or follow a given path, or any kind of parametrized reference. Secondary motion control constraints could also be present, such as speed and acceleration constraints.

An important difference amongst motion controllers is the type of model assumed for the robot. Considering the robot as a mechanical system, two types of control strategies exist, differing by the type of model considered for the robot: (i) *kinematic* control and (ii) *dynamic* control.

**Kinematic Control.** In *kinematic* control, the robot is modeled as a rigid body system with state space defined by $x = (p, \phi)$, where $p \in \mathbb{R}^{n_p}$ is the robot position state in 2D or 3D space ($n_p = 2, 3$) with respect to a reference frame $\mathscr{F}$, and $\phi$ is a planar or spatial orientation state with respect to the same frame $\mathscr{F}$. In the 2D plane, common parametrizations for orientation are: (i) angle $\phi \in [-\pi, \pi]$, (ii) 2D rotation matrices ($\phi \in \mathbb{SO}^2$) and (iii) unitary complex numbers ($\phi = a + bi \in \mathbb{C}, |\phi|^2 = a^2 + b^2 = 1$). In the 3D plane, common parametrizations for orientation are: (i) Euler angles ($\phi \in \mathbb{R}^3$), (ii) 3D rotation matrices ($\phi \in \mathbb{SO}^3$) and (iii) unitary quaternions ($\phi = a + bi + cj + dk \in \mathbb{Q}, |\phi|^2 = a^2 + b^2 + c^2 + d^2 = 1$). The robot kinematics is then defined as

$$\dot{x} = J(x)u \tag{2.18}$$

where $u \in \mathbb{R}^m$ is the kinematic control variable, composed of the controllable velocities of the robot degrees of freedom. In wheeled vehicles and quadrotors, for example, the control variable $u$ consists of the velocity of the wheels and propellers, respectively. The matrix-valued function $J : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ is a Jacobian matrix, transforming the velocities of the robot degrees of freedom into the velocity of the robot pose state $x \in \mathbb{R}^n$. Notice how (2.18) is basically control affine system described by (2.13) with $f(x)$ identically zero and $g(x) = J(x)$.

**Dynamic Control.** In *dynamic* control, the robot is modeled as a Newtonian, Lagrangian or Hamiltonian mechanical system, with state space defined by $x = (p, \phi, v, \omega)$ where $(p, \phi)$ are parametrizations for the robot pose state as before, and $(v, \omega)$ are the robot linear and angular velocities in frame $\mathscr{F}$, which are dependent on the rate of the pose states as $v = \dot{p}$ and $\omega = J_\phi(\phi)\dot{\phi}$, where $J_\phi$ is a Jacobian mapping the rate of the orientation parameter $\phi$ to the angular velocity $\omega$. The robot dynamic equations of motion are given by

$$M(p, \phi) \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} + C(p, \phi, v, \omega) \begin{bmatrix} v \\ \omega \end{bmatrix} + G(p, \phi) = \Gamma\tau \tag{2.19}$$

where $M$ is an positive definite and invertible mass (or inertia) matrix, $C$ is the term of inertial forces and viscous friction forces acting on the robot, $G$ is the term of conservative forces acting on the vehicle, such as gravity, $\Gamma$ is a control matrix, and $\tau \in \mathbb{R}^m$ is the vector of controllable forces and torques of the robot actuators. Since $M$ is always invertible and considering $\tau = u \in \mathbb{R}^m$ as the control variable, it is possible to rewrite the differential equation (2.19) in a state space form of (2.13), showing that in dynamic control, the robot model is also a nonlinear affine control system.

Both models are nonlinear, *control-affine systems*, which is why most of the literature on nonlinear control theory for robotic navigation focuses on this particular model. Particularly, a large portion of the literature on robot navigation focuses on the development of *kinematic* controllers. The reasons are twofold [74]:

1. Kinematic models are usually simpler than dynamic models, having a significative lower number of parameters associated with the robot mechanical design and its actuators. For many applications, a large number of control parameters often results in great difficulty in tuning the controller to achieves acceptable performance.

2. Wheeled robots and quadrotors are usually equipped with electrical motors that are independently controlled by low-level velocity controllers. If these regulation loops are efficient, the velocity errors remain bounded to small values, even when the desired velocity and the actuator load vary continuously within a certain range. This type of robustness allows the desired velocity to be treated as a free control variable. Even if these loops are not present, they are usually easy to design and tune.

In this thesis we focus exclusively on the hypothesis *kinematic* control for robot navigation. Navigation solutions for Autonomous Surface Vessel (ASV) have been developed and improved in recent and relevant works like the usage of dynamic graphs in [88], using data-driven planners with historical navigation data-set in [78], or using angles-guidance fast-marching square algorithms (AFMS) explored in [48, 78]. ASV planning and control frameworks have been addressed in research works that range from classical based methods using PID path tracker controllers [63, 17], robust control with a closed-loop shaping-filter (CSF) [30], Model Predictive Control (MPC) with an Integrated Probabilistic Data Association (IPDA) improvement [44], to newly developed machine learning techniques using Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) planning systems [92, 82, 19].

### 2.3.1 Path Following and Moving Path Following Control

While many different control strategies for robot motion exist, trajectory tracking and path following are widely studied frameworks encompassing a wide range of motion control tasks that mobile robotics are usually required to perform. They present similarities, but also important differences. Basically, in *trajectory tracking*, the vehicle must be follow a specific *trajectory* in space and time, that is, explicit time time constraints are imposed. In contrast, in *path following*, the task is to steer the vehicle to a specific path in space only, without the requirement of meeting specific time constraints. For completeness and clarity, we provide a formal definition for both problems.

**Definition 2.3.1** (Trajectory tracking [1])**.** Consider a single vehicle with position $p \in \mathbb{R}^n$ and let $p_d(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ be a given desired reference trajectory. The trajectory tracking problem consists of designing a feedback control law such that all the relevant closed-loop signals are bounded and the tracking error $\|p(t) - p_d(t)\|$ converges to zero as $t \to \infty$.

**Definition 2.3.2** (Moving Path following [1])**.** Consider a single vehicle with position $p \in \mathbb{R}^n$ and a given target frame $\mathscr{T}$ with origin $p_t \in \mathbb{R}^n$ moving with linear velocity $v_t = \dot{p}_t$ and angular velocity $\omega_t \in \mathbb{R}^{\frac{1}{2}n(n-1)}$. Additionally, let $p_d^t(\gamma) : \mathbb{R} \to \mathbb{R}^n$ be a given desired geometric path expressed in $\mathscr{T}$ and parameterized by a scalar variable $\gamma \in \mathbb{R}$. In an inertial frame $\mathscr{T}$, the geometric path is described by $p_d(\gamma, t) = p_t + R(\phi_t)p_d^t(\gamma)$, where $\phi_t \in \mathbb{R}^{\frac{1}{2}n(n-1)}$ is some parametrization for the

orientation of $\mathscr{T}$ with respect to $\mathscr{I}$, and $R(\phi_i) \in \mathbb{SO}^n$ is the corresponding rotation matrix. The moving path following problem consists of designing a feedback control law such that, as $t \to \infty$: (i) all the relevant closed-loop signals are bounded, (ii) the MPF error $\|p(t) - p_d(\gamma,t))\|$ converges to zero and (iii) the parameter $\gamma$ satisfies some desired dynamic law $\dot{\gamma} = v_\gamma(\gamma,t)$.

Basically, in Definition 2.3.2, if the target frame $\mathscr{T}$ is fixed, the problem reduces to usual path following, where the robot is tasked to follow a fixed path $p_d(\gamma)$ in space, parametrized by the path variable $\gamma$. A practical example of application for MPF control is the problem of an UAV or AUV following a path attached to a moving target, an illustrated in Fig. 2.1.

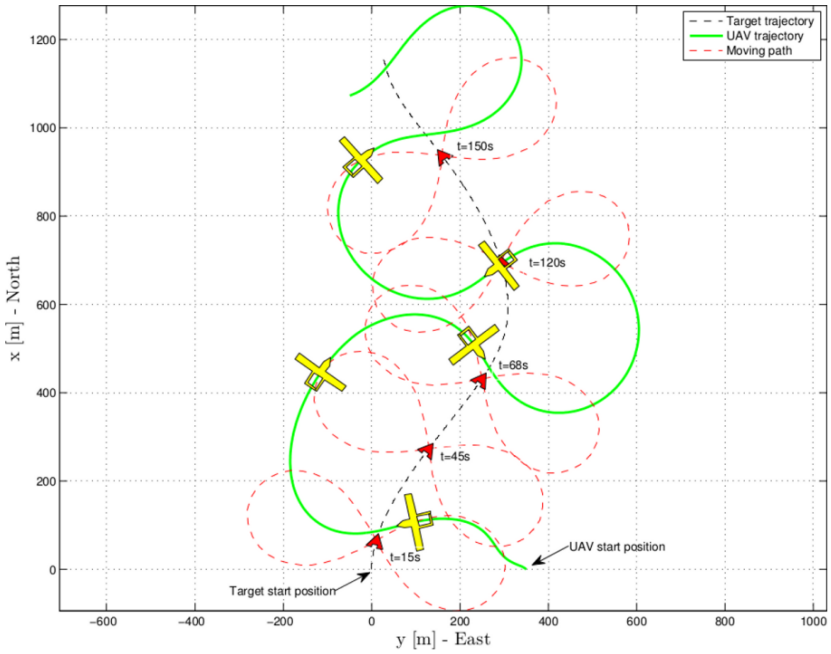



Figure 2.1: Example of a moving path following control application with an aircraft following a target [1].

The trajectory tracking and (moving) path following control literature was inspired by ideas introduced in [16], giving rise to control strategies for marine vehicles in [22, 75] and for Unmanned Aerial Vehicles (UAVs) in [87, 25, 39]. Compared to trajectory tracking, Moving Path Following is a much less restrictive task, simply by the fact that MPF does not impose time constraints in the robot's motion. In contrast, trajectory tracking can be highly demanding for the vehicle's actuators, due to the strict time constraints imposed by the controller. In [3], the performance limits in trajectory tracking and path following for non-minimum phase systems were investigated, and a fundamental difference between the two problems was discovered. While for trajectory tracking there exists a performance limitation in the lower bound of the tracking error norm, this is not the case in the path following problem, even with the speed assignment imposed to the path variable. Besides this advantage of the path following scheme, there are applications where time-dependent constraints are required to be followed by the vehicle.

Additional applications of MPF control include source seeking, convoy protection and target tracking [91], to cite a few [18]. This notion was introduced in [58, 56] for an UAV tracking a ground target, the latter employing the same Serret-Frenet frame approach to the 2D MPF problem. These results were extended to the 3D case in [57], where unit quaternions were used for attitude representation. All of these approaches assume that the virtual point to be followed is located at the point on the path that is closest to the vehicle (taking the projection of the vehicle's position on the path). Therefore, if the vehicle has motion constraints, the proposed control laws constrains the initial position of the vehicle with respect to the moving path. This issue was overcome in the work of [85] by explicitly controlling the progression of the virtual target along the path. Besides, [85] was maybe the first work on the MPF problem for UAVs that considered the presence of *disturbances* acting on the vehicle. Other control methods such as the vector field method [40] and nonlinear model predictive control [37] have been proposed to solve the MPF problem for mobile robots.

Recently, in path following literature, the problem of robustness has been addressed for example in [15], where a cascade sliding mode controller for both kinematics and dynamics of a robotic vehicle was designed. In [4], a disturbance observer for constant unknown ocean currents was designed to solve the problem of dynamic positioning and way-point tracking of an underactuated AUV. Later on, the problem of robustness against parametric uncertainty in trajectory tracking and path following was also addressed in [2]. At [90], a sliding mode technique combined with a predictive control strategy was developed to compensate for the impact of the hydrodynamic damping coupling on a 3D path following task for an Autonomous Underwater Vehicle (AUV). In [84], a $H_\infty$ robust controller for ground vehicles is proposed to achieve path following in the presence of disturbances caused by delays and data packet dropouts. Also recently, in [29], two cooperative path following controllers using an Extended State Observer to estimate and compensate external disturbances in the kinetic level were proposed and validated experimentally using Autonomous Surface Vehicles (ASVs). All of the above schemes consider robustness for the Path Following (PF) problem, but not for MPF problem, where high speeds induced by the motion of the target could potentially be an important source of disturbances that could lead to loss of performance, if not properly compensated. In Chapter 3, we tackle the problems of robust MPF and robust cooperative MPF.

### 2.3.2 Cooperative Control

Regarding multi-robot systems, generalizations of the MPF problem can be considered. Some common motivations for the development of cooperation strategies among multiple robot systems are: (i) task complexity and of distributed nature, (ii) simplified robot design, (iii) parallelism, (iv) increase of robustness through redundancy. In the context of mobile robot systems, the problem of *decentralized* cooperation consists of achieving a formation pattern between the agents (robots) by only exchanging local information about their environment [59].

A possible generalization of the MPF control scheme described in Section 2.3.1 is the following: what if each robot from a fleet of mobile robots must follow a specific path with respect to a

fixed or moving reference frame while at the same time achieving some kind of formation pattern? This task is known as the *Cooperative Path Following* (CPF) problem for a *fixed* target frame $\mathscr{T}$, and the *Cooperative Moving Path Following* (CMPF) problem for a *moving* target frame $\mathscr{T}$. To accomplish this complicated task, each of the vehicles must follow a path following or *moving* path following algorithm, and a *coordination* algorithm must be responsible for controlling the speeds of the vehicles to achieve the desired synchronicity among them. Often, these algorithms must take into account the topology of the inter-vehicle communications network.

The general CMPF problem can be formally described by the following.

**Definition 2.3.3** (Cooperative Moving Path Following [1])**.** Consider a team of $N$ vehicles with positions $p_i \in \mathbb{R}^n$ with $i \in \mathbb{I} = \{1, \cdots, N\}$ and a *moving* target frame $\mathscr{T}$ as in Definition 2.3.2. Let $p^t_{d_i}(\gamma_i) : \mathbb{R} \to \mathbb{R}^n$ be a fixed desired geometric path (when expressed in the target frame $\mathscr{T}$) that the $i$-th robot is tasked to follow, where $\gamma_i \in \mathbb{R}$ is the $i$-th parametric path variable. Additionally, let $v_d \in \mathbb{R}$ be a desired common speed assignment for the team. Suppose that the vehicles are supported by an inter-vehicle communication network, and, therefore, for each vehicle $i \in \mathbb{N}$, the variables $\gamma^i$ and $\gamma^j$, $i \neq j \in \mathbb{N}$ are available. The cooperative moving path following problem consists of designing a feedback control law such that, as $t \to \infty$: (i) the MPF errors converge to zero, (ii) for each pair of vehicles $i \neq j \in \mathbb{N}$, the coordination errors $\gamma^i - \gamma^j$ converge to a constant value and (iii) the path variables $\gamma_i$ satisfy the desired common speed assignment $\dot{\gamma}_i \to v_d, \forall i \in \mathbb{N}$.

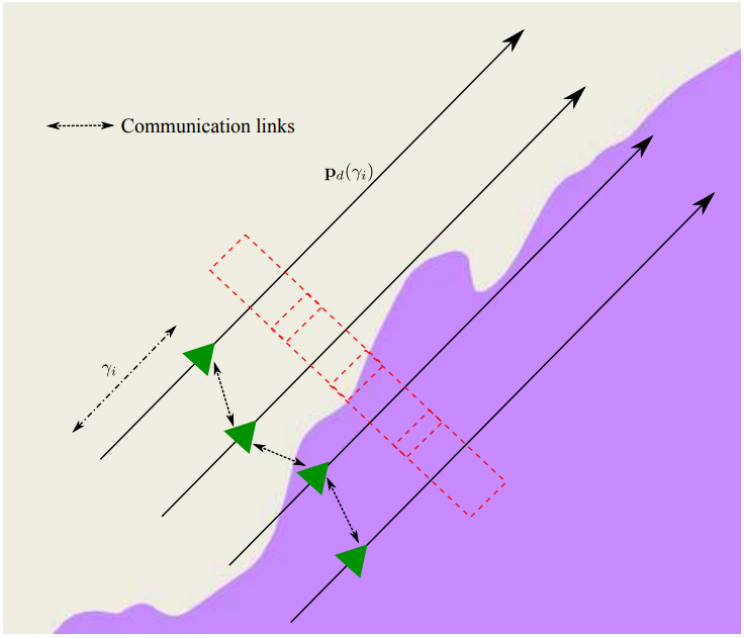



Figure 2.2: Example of a cooperative path following control application for coastal monitoring with UAVs flying in a formation pattern [35]. The vehicles constantly exchange data about their locations to maintain coordination.

The concept of Cooperative Moving Path Following is illustrated in Fig. 2.2. The works [5] and [28] formally introduce the Cooperative Path Following (CPF) framework using dynamical

system concepts, backed with mathematical tools borrowed from graph theory and theory of input-to-state stable (ISS) systems. Since the vehicles must maintain communication among them to achieve consensus in the formation, one practical challenge is how to reduce the frequency of communication while maintaining stability on the formation pattern. There, it is needed to design systems that are able to decide when and what minimum amount of information should be transmitted from each of the vehicles to its neighbors [5]. To accomplish this, a self-triggered control strategy for cooperative path following is proposed in [35]. Similar interesting problems arise in the area of networked control systems [32, 89]. Cooperative target tracking for a fleet of Unmanned Aerial Vehicles (UAVs) is considered in [50, 51] with different communication typologies and time-varying formations. In [14], the problem of collision avoidance with *moving obstacles* in cooperative path following was addressed, were the collision avoidance system was composed by two distinct modules: collision prediction and collision avoidance. The former uses a bank of Kalman filters running in parallel to estimate the most probable trajectory of the obstacle. If a collision is predicted, the latter effectively modifies the vehicle speeds, re-planning the paths through harmonic potential fields. Results on Cooperative Moving Path Following for under-actuated vehicles in the presence of environmental disturbances are presented in Chapter 3.

### 2.3.3 Path Following Control Design for Multiple Vehicles

The results presented in Chapters 3 and 4 concentrate on the planar PF and MPF problem for teams of vehicles under the assumption of kinematic control. Therefore, we introduce a brief discussion on the matter, focusing on the models utilized for the vehicles, how the path following errors are usually defined and a simple closed-loop path following controller. Generalizations of the following derivations will be presented in Chapter 3, when the problem of robust MPF will be considered.

Consider an inertial frame of reference $\mathscr{I}$ and $N$ robotic vehicles, each with its own body frame $\mathscr{B}_i$ attached to its center of mass. Define the set of integer indexes for each of the $N$ robotic vehicles as $\mathbb{I} = \{1, 2, ..., N\}$. The pose of frame $\mathscr{B}_i$ with respect to the inertial frame $\mathscr{I}$ is given by $(p_i, \phi_i) \in \mathbb{R}^n \times \mathbb{R}^{\frac{1}{2}n(n-1)}$, where $p_i = [\, x_i \;\; y_i \,]^\mathsf{T}$ $(n = 2)$ or $p_i = [\, x_i \; y_i \; z_i \,]^\mathsf{T}$ $(n = 3)$ is the position of the center of mass of the $i$-th robot and $\phi_i$ is a certain orientation parametrization (angles, complex number or unit quaternions, etc.). A standard kinematic model for robot navigation is given by

$$\begin{aligned} \dot{p}_i &= R(\phi_i) v_i + d_{v_i} \\ \dot{R}_i &= R(\phi_i) S(\omega_i + d_{\omega_i}) \end{aligned} \tag{2.20}$$

where $v_i = [\, v_{f_i} \;\; 0 \,]^\mathsf{T}$ $(n = 2)$ or $v_i = [\, v_{f_i} \; 0 \; 0 \,]^\mathsf{T}$ $(n = 3)$ and $\omega_i \in \mathbb{R}^{\frac{1}{2}n(n-1)}$ are the robot's linear and angular velocities in its body frame $\mathscr{B}_i$. Considering the $x$-axis of $\mathscr{B}$ as the robot's longitudinal direction, the scalar $v_{f_i} \in \mathbb{R}$ is the longitudinal speed of the robot. Model (2.20) assumes that the robots are underactuated, and thereby cannot generate instantaneous lateral or vertical velocities. Matrix $R_i = R(\phi_i) \in \mathbb{SO}(n)$ denotes the rotation matrix for the rotation of frame $\mathscr{B}_i$ with respect to $\mathscr{I}$, while $S(\omega_i) \in \mathfrak{so}(n)$ is the skew-symmetric matrix associated to the angular velocity $\omega_i$.

Vectors $d_{v_i} \in \mathbb{R}^n$ and $d_{\omega_i} \in \mathbb{R}^{\frac{1}{2}n(n-1)}$ are velocity disturbances acting on each robot. Many different factors can be the source of these disturbances, depending on the type of vehicle and the operational environment. Marine vehicles such as AUVs are affected by unknown sea conditions that can induce unwanted external velocities due to maritime currents, waves and wind. In the case of aerial vehicles, wind and internal dynamics can induce some unwanted disturbances in the kinematic model. The complete model of (2.20) is used in Chapter 3, where robust control strategies are developed to compensate these disturbances. In particular, in the plane and without any disturbances ($d_{v_i} = 0$ and $d_{\omega_i} = 0$), (2.20) reduces to the simple model

$$\dot{x}_i = v_{f_i}\cos(\phi_i)\,, \quad \dot{y}_i = v_{f_i}\sin(\phi_i)\,, \quad \dot{\phi}_i = \omega_i \tag{2.21}$$

also known as the *unicycle* dynamics, where $\phi_i \in [-\pi, \pi]$ is a simple planar angle. The control variable for kinematic control is

$$u_i = \begin{bmatrix} v_{f_i} \\ \omega_i \end{bmatrix} \in \mathbb{R}^{\frac{1}{2}n(n-1)+1} \tag{2.22}$$

composed of controllable linear forward (longitudinal) speed and angular velocities for the $i$-th robot.

Often, in PF or MPF control and their cooperative counterparts, the controlled vehicle's position is given by an offset position from the vehicle's center of mass position $p_i$ where the angular velocity $\omega_i$ is applied: define the inertial position

$$p_{\varepsilon_i} = p_i + R(\phi_i)\,\varepsilon \tag{2.23}$$

where $\varepsilon = [\,\varepsilon_x \;\; \varepsilon_y\,]^\mathsf{T}$ $(n = 2)$ or $\varepsilon = [\,\varepsilon_x \;\; \varepsilon_y \;\; \varepsilon_z\,]^\mathsf{T}$ $(n = 3)$. If the $x$-direction of the robot's body frame $\mathscr{B}_i$ is regarded as its longitudinal direction, setting $\varepsilon_y = \varepsilon_z = 0$ sets the position $p_{\varepsilon_i}$ to a point in front of the robot, displaced from it's center of mass by a distance $\varepsilon_x$. Using this relation, the unperturbed dynamics of the controlled offset position of for the $i$-th robot takes the form

$$\dot{p}_{\varepsilon_i} = R(\phi_i)\Delta_\varepsilon\, u_i \tag{2.24}$$

where $\Delta_\varepsilon$ is a constant matrix that can take the forms

$$\Delta_\varepsilon = \begin{bmatrix} 1 & -\varepsilon_y \\ 0 & \varepsilon_x \end{bmatrix} \quad \text{or} \quad \Delta = \begin{bmatrix} 1 & 0 & \varepsilon_z & -\varepsilon_y \\ 0 & -\varepsilon_z & 0 & \varepsilon_x \\ 0 & \varepsilon_y & -\varepsilon_x & 0 \end{bmatrix} \tag{2.25}$$

for 2D or 3D, respectively. Usually, $\varepsilon$ is chosen in such a way that $\Delta_\varepsilon$ is full rank. Often, the PF error is written in the robot's body frame $\mathscr{B}_i$ as

$$e_i = R(\phi_i)^\mathsf{T}(p_{\varepsilon_i} - p_{d_i}(\gamma_i))\,, \quad i \in \mathbb{I} \tag{2.26}$$

where the $p_{d_i} : \mathbb{R} \to \mathbb{R}^n$ is a map describing the position of a virtual point moving along the path that the $i$-th robot is tasked to follow, as the path variable $\gamma$ changes.

Next, a standard path following controller for the $i$-th vehicle is presented, under the assumption of no disturbances acting on the robots. The controllers developed in Chapter 3 follow a similar design, based in the Lyapunov control theory introduced in Section 2.2.

**Assumption 1.** The geometric path $p_{d_i}$ is a class $C^1$ function.

With $d_{v_i} = 0$ and $d_{\omega_i} = 0$ and using (2.24)-(2.26), the time derivative of (2.26) yields

$$\begin{aligned} \dot{e}_i &= \dot{R}(\phi_i)^\mathsf{T} R(\phi_i) e_i + R(\phi_i)^\mathsf{T} (\dot{p}_{\varepsilon_i} - \dot{p}_{d_i}(\gamma_i)) \\ &= R(\phi_i)^\mathsf{T} S(\omega_i) R(\phi_i) e_i + \Delta_\varepsilon u_i - R(\phi_i)^\mathsf{T} \nabla p_{d_i}(\gamma_i)\, \dot{\gamma}_i \end{aligned} \tag{2.27}$$

where Assumption 1 was needed in order to compute the first derivative of $p_d(\gamma)$.

**Theorem 2.3.1** (Path following control [37])**.** Consider the $i$-th robotic vehicle with state given by $x_i = (p_i, \phi_i)$, kinematic model given by (2.20) without disturbances ($d_{v_i} = 0$, $d_{\omega_i} = 0$) and control signal given by $u_i = [\, v_{f_i} \;\; \omega_i^\mathsf{T} \,]^\mathsf{T}$. Let the path following error be defined by (2.26). Under Assumption 1, the PF control law

$$u_{\mathrm{pf},i}(x) = \Delta_\varepsilon^{-1} \left( -K_p\, e_i + R(\phi_i)^\mathsf{T} \nabla p_{d_i}(\gamma_i)\, \dot{\gamma}_i \right) \tag{2.28}$$

with $K_p > 0$ ensures that the origin $e_i = 0$ of the path following error is globally asymptotically stable.

*Proof.* Consider the Lyapunov candidate $V = \frac{1}{2}\|e_i\|^2$. Using (2.27) and control law (2.28), its time derivative is given by:

$$\begin{aligned} \dot{V} = e_i^\mathsf{T} \dot{e}_i &= e_i^\mathsf{T} \Delta_\varepsilon\, u_{\mathrm{pf},i}(x) - e_i^\mathsf{T} R(\phi_i)^\mathsf{T} \nabla p_{d_i}(\gamma_i)\, \dot{\gamma}_i \\ &= -e_i^\mathsf{T} K_p e_i < 0 \end{aligned} \tag{2.29}$$

Thus, the origin of the PF error is globally asymptotically stable. □

## 2.4 Safety-Critical Systems

In Section 1.1, we have mentioned that a robot must not only fulfill its desired task with the desired levels of efficacy, but must also do so in a *safe* way according to some measurable and objective requirement. In many applications, incorrect, unexpected or faulty robotic operation could result in equipment or enviromental damage (including to the robot itself) or even human injury or loss of lifes. In these cases, the system including the robot and its operational environment is considered a *safety-critical system* [42].

The design of safe controllers for safety-critical systems is a fruitful and rich topic receiving a growing amount of attention nowadays. They are of crucial importance for many industrial

sectors and production lines, where the stability of feedback-controlled systems is just so important as their capacity to provide safe behaviour under a wide variety of operational circumstances. Furthermore, safety is also a mandatory property for systems with high levels of interoperability, cooperation, or coordination with humans.

The notion of *safety* was first introduced in 1977 in the context of program correctness by [45] and later formalized in [6], which also introduced the concept of *liveness*. Intuitively, one can describe these two contrasting system properties as: (i) the requirement of avoiding undesired situations while (ii) guaranteeing the eventual achievement of a desired configuration, respectively. As pointed out by [8], in the context of control systems, liveness can be identified as an *asymptotic stability* requirement with respect to a certain set of desired or objective states (which can be achieved by means of Lyapunov-based controllers, as seem in Section 2.2), while *safety* can be defined as the *invariance* of the system trajectories to some set, defined as the set of safe states.

### 2.4.1 Control Barrier Functions and Safety

The concept of *safety* can be thought as complementary, or dual to that of stabilization: while the purpose of the latter is to guide the system towards an objective, the former aims to avoid the system trajectories from entering regions of the state space that are considered to be unsafe.

The concept of a barrier function was initially used in constrained optimization [23] due to their ability to provably establish invariance properties of sets. In [62, 61], barrier certificates were first introduced as a tool to formally prove safety of nonlinear and hybrid systems in the form of *forward invariance* to the system trajectory with respect to some set on its state space.

**Definition 2.4.1** (Forward Invariance [7])**.** Given the autonomous feedback system (2.14), a set $\mathscr{C} \in \mathbb{R}^n$ is forward invariant if, for every $x(0) \in \mathscr{C}$, $x(t) \in \mathscr{C}$ for all $t \in \mathscr{I}(x(0))$, the maximum interval of existence of $x(t)$.

That is, a set is forward invariant if all the trajectories that start inside of it, remain inside of it. Therefore, the concept of *safety* of an autonomous system (2.14) with respect to a set can be framed in the context of forward invariance of the system trajectories with respect to that set.

**Definition 2.4.2** (Safety)**.** The autonomous feedback system (2.14) is said to be *safe* with respect to set $\mathscr{C}$ if $\mathscr{C}$ is forward invariant.

In this work, we consider the framework of zeroing Control Barrier Functions, first proposed in [86] and in which the safe set $\mathscr{C}$ is defined as the superlevel set of some continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$:

$$\mathscr{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\} \tag{2.30}$$

$$\partial\mathscr{C} = \{x \in \mathbb{R}^n : h(x) = 0\} \tag{2.31}$$

In this framework, $\mathscr{C}$ in is known as the set of all states considered to be safe (i.e., the *safe* set), while its complement $\overline{\mathscr{C}} = \mathbb{R}^n \setminus \mathscr{C}$ is known as the set of unsafe states (the *unsafe* set).

Other historical frameworks for barrier functions exist, such as reciprocal CBFs [80, 7], where the barriers are defined as reciprocal functions of the zeroing barrier $h$ in (2.30). However, reciprocal CBFs get arbitrarily large when close to the boundary $\partial\mathscr{C}$, thereby suffering from numerical issues. For this reason, recent CBF literature focuses exclusively on the zeroing CBF framework. With these initial definitions, the concept of a Control Barrier Function (CBF) as akin to the concept of a CLF can be introduced as follows.

**Definition 2.4.3** (CBF [86])**.** Let $\mathscr{C} \in \mathbb{R}^n$ be defined by (2.30). Then $h(x)$ is a (zeroing) *Control Barrier Function* (CBF) for (2.13) if there exists a locally Lipschitz extended class $\mathscr{K}$ function $\beta$ such that

$$\sup_{u\in\mathbb{R}^m} \left( \underbrace{L_f h(x) + L_g h(x)\, u}_{\dot{h}} \right) \geq -\beta(h(x)) \quad \forall x \in \mathbb{R}^n . \tag{2.32}$$

Similarly to the definition of a CLF, Definition 2.4.3 means that there exists a set of *safe* state-feedback controls $\mathbb{K}_{\text{cbf}}(x) \subset \mathbb{R}^m$ that renders $\mathscr{C}$ forward invariant. This set can be defined as

$$\mathbb{K}_{\text{cbf}}(x) = \left\{ u \in \mathbb{R}^m : \underbrace{L_f h(x) + L_g h(x)\, u}_{\dot{h}} \geq -\beta(h(x)) \right\} \tag{2.33}$$

That means that any feedback control $u^\star \in \mathbb{K}_{\text{cbf}}(x)$ only allows the CBF to decrease its value when the state is in the interior of the safe set $\text{int}(\mathscr{C})$, but not on its boundary $\partial\mathscr{C}$, thus keeping the trajectories inside of the safe set $\mathscr{C}$.

**Remark.** If the state starts inside the unsafe set ($x(0) \in \overline{\mathscr{C}}$), the CBF is only allowed to increase its value (since $\dot{h} > 0$ in this case), thus making the trajectories diverge from the unsafe set $\overline{\mathscr{C}}$ and converge to the safe set $\mathscr{C}$, effectively making it asymptotically stable.

Thus, akin to the way the CLFs can enforce stability of a given point, CBFs can be used to enforce safety with respect to a safe set $\mathscr{C}$, meaning that if a CBF exists for the system (2.13), then a safe controller generating a control from (2.33) is realizable.

Figure 2.3 illustrates diagramatically how CLFs and CBFs can be used to enforce both stabilization and *safety* of trajectories in the form of *forward invariance* with respect to a given set $\mathscr{C}$ of safe states.

### 2.4.2 QP-based Minimum-Norm Controllers

As seem in Section 2.2, CLFs can be used to design a feedback stabilizable controller (2.17) for the nonlinear affine system system (2.13), using Sontag's formula (2.16). In [24], the following

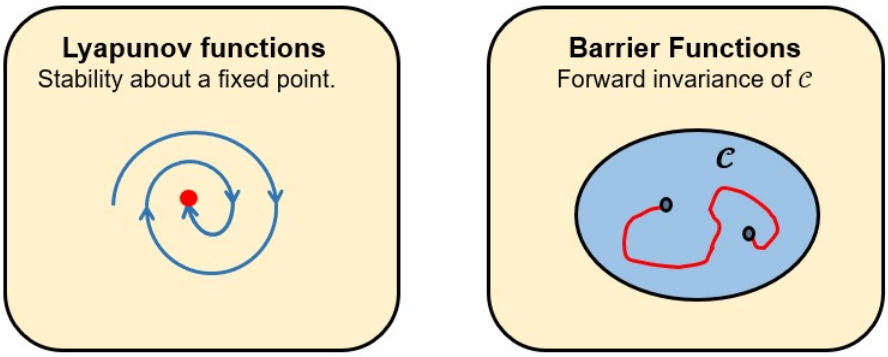



Figure 2.3: CLFs and CBFs as a way to enforce stability and safety, respectively. Source: https://biron.ece.ufl.edu/research/.

minimum-norm QP controller was proposed for feedback stabilization:

$$u^\star(x) = \underset{u\in\mathbb{R}^m}{\operatorname{argmin}} \|u\| \tag{2.34}$$
$$s.t.\ L_f V(x) + L_g V(x)\, u \leq -\alpha(V(x)) \tag{CLF}$$

with $\alpha$ being a class $\mathscr{K}$ function. The QP (2.34) is able to generate a feedback control from the set $\mathbb{K}_{\text{clf}}(x)$ defined in (2.15) with minimum norm $\|u\|$. That is, the QP (2.34) is able to stabilize the CLF minimum $x_0$ for system (2.13).

Instead of an stabilization objective, if only safety is required, the same QP framework can also be used to enforce safety with respect to the safe set $\mathscr{C}$ as defined in (2.30), with

$$u^\star(x) = \underset{u\in\mathbb{R}^m}{\operatorname{argmin}} \|u\| \tag{2.35}$$
$$s.t.\ L_f h(x) + L_g h(x)\, u \geq -\beta(h(x)) \tag{CBF}$$

with $\beta$ being an extended class $\mathscr{K}$ function [47]. Similarly, the QP (2.35) is able to generate a feedback control $u^\star(x) \in \mathbb{K}_{\text{cbf}}(x)$ defined in (2.33) with minimum norm $\|u^\star(x)\|$. That is, the QP (2.35) is able to guarantee that the state trajectories of the system (2.14) are forward invariant with respect to the safe set $\mathscr{C}$ (and therefore, safe). However, (2.35) is not able to joint guarantee safety *and* stability.

In [31], a different formulation was proposed to seek a safe controller that is closest to a nominal (and potentially unsafe) stabilizing controller $u_{\text{nom}}(x)$, also by means of a QP formulation:

$$u^\star(x) = \underset{u\in\mathbb{R}^m}{\operatorname{argmin}} \|u - u_{\text{nom}}(x)\| \tag{2.36}$$
$$s.t.\ L_f h(x) + L_g h(x)\, u \geq -\beta(h(x)) \tag{CBF}$$

Controller (2.36) is known as a *safety filter*, makes use of a CBF $h$ and can also guarantee the forward invariance of the state trajectory with respect to safe set $\mathscr{C}$. The difference between

(2.36) and (2.35) is that the former can also achieve stabilization of the system trajectory to an objective, defined by the nominal controller $u_{\text{nom}}(x)$. The safety-filter QP (2.36) simply generates the "closest" safe control $u^\star(x) \in \mathbb{K}_{\text{cbf}}(x)$ to the stabilizing control $u_{\text{nom}}(x)$ in order to guarantee that the trajectory is always safe.

None of these QP strategies use CLFs and CBFs jointly in an unified framework. One question can be asked: can this be done? This question was answered in [8], which proposed following QP for both stabilization and safety of trajectories:

$$u^\star(x), \delta^\star(x) = \underset{(u,\delta)\in\mathbb{R}^{m+1}}{\operatorname{argmin}} \; \|u\|^2 + p\delta^2 \tag{2.37}$$

$$s.t.\; L_f V(x) + L_g V(x)\, u \leq -\alpha(V(x)) + \delta \tag{CLF}$$

$$L_f h(x) + L_g h(x)\, u \geq -\beta(h(x)) \tag{CBF}$$

where $p > 0$, $\alpha$ and $\beta$ are of class $\mathscr{K}$ and extended class $\mathscr{K}$ functions, respectively. The objective of (2.37) is to minimize the norm of the control and auxiliary *relaxation* variables $u$ and $\delta$ while satisfying the CLF and CBF constraints.

The CBF constraint guarantees that $u \in \mathbb{K}_{\text{cbf}}(x)$, keeping the system trajectory safe. While safety is a *hard* constraint of (2.37), meaning that $\mathscr{C}$ is always forward invariant, the relaxation variable $\delta$ in the CLF constraint *softens* the stabilization objective, maintaining the feasibility of the QP, but also opening the possibility of hampering the stabilization objective. Actually, in the published in [64], we have shown that the presence of the relaxation variable $\delta$ in (2.37) hampers stabilization in such a level that undesirable equilibrium points other than the CLF minimum are introduced into the closed-loop system (2.14). These come in two different flavors: *interior* or *boundary* equilibrium points, depending on the region of the safe set $\mathscr{C}$ where they occur. The fact that some of these undesirable points can be asymptotically stable and can even occur on the safe set *boundary* $\partial\mathscr{C}$ is an important practical limitation of the CLF-CBF-based QP framework, since it could not only trap the system in an undesirable state but also expose it to close-to-failure situations, forcing the designer to opt for highly conservative safety margins when designing the system safety specifications.

Some works have attempted to address the problem of trajectory deadlocks (that is, equilibrium points) when combining stabilization and safety objectives with the safety-filter QP framework of (2.36). In this framework, deadlocks are caused by a conflict between the stabilization objectives of the feedback nominal controller $u_{\text{nom}}(x)$ and the safety constraint. In [83], the problem was managed by introducing a consistent perturbation into the QP constraint. Although efficient for solving some types of deadlocks, this method *softens* the safety constraint, allowing for the possibility of leaving the safe set if the deadlock situation happens to occur on the boundary, which could lead to unsafe behavior. [54] have proposed a method that solves the deadlock problem for the safety-filter QP (2.36) considering multiple non-convex unsafe regions modeled by CBFs. Undesirable stable equilibria are avoided by using a nonlinear transformation that maps the system state into a new domain called the "ball world", where all nonconvex CBF boundaries are

converted into moving $n$-spheres and dynamically avoid the state, reducing the stability region of stable equilibrium points to a set of zero measure. Although effective to solve deadlocks, the nonlinear transformation used by their proposed method can be highly dependent on the CBF geometry, and thus could be difficult to find for arbitrarely complex barrier shapes.

None of the methods proposed by [83, 54] address the problem of deadlocks considering the CLF-CBF-QP-based framework of (2.37). This topic was first addressed in [38], which has proposed a modification of (2.37) seeking to avoid *interior* equilibrium points, which are only one the types of undesirable equilibria that can appear in the formulation. Latter, [79] proposed a modified CLF-CBF-based QP controller in which undesirable *interior* equilibrium points and *boundary equilibrium points* with a certain particular property do not exist for the resulting closed-loop system. However, none of these two works address general types of boundary equilibrium points that could still occur in the general case.

Another challenge to be overcome in the field is the composition of multiple safety requirements. Consider $N$ continuously differentiable barriers $h_i : \mathbb{R}^n \to \mathbb{R}$ and their corresponding safe sets $\mathscr{C}_i \subset \mathbb{R}^n$ defined as the superlevel sets of $h_i$, $\mathscr{C}_i = \{x \in \mathbb{R}^n : h_i(x) \geq 0\}$ as in (2.30). The composite safe set associated to all the $N$ CBFs is

$$\mathscr{C} = \bigcap_{i=1}^{N} \mathscr{C}_i \tag{2.38}$$

and the set of safe controls rendering $\mathscr{C}$ forward invariant is

$$\begin{aligned} \mathbb{K}_{\text{cbf}}(x) &= \bigcap_{i=1}^{N} \mathbb{K}_{\text{cbf}_i}(x) \\ \mathbb{K}_{\text{cbf}_i}(x) &= \{u \in \mathbb{R}^m : L_f h_i(x) + L_g h_i(x)\, u \geq -\beta_i(h_i(x))\} \end{aligned} \tag{2.39}$$

Assume that distinct requirements are incorporated to the safety-critical problem by means of $N$ CBF constraints in the safety-filter QP with multiple CBFs below:

$$\begin{aligned} u^\star(x) = \underset{u \in \mathbb{R}^m}{\text{argmin}} \|u - u_{\text{nom}}(x)\|^2 & \qquad (2.40) \\ s.t.\ L_f h_i(x) + L_g h_i(x)\, u \geq -\beta_i(h_i(x)), \quad i \in \{1, \cdots, N\} & \qquad \text{(CBFs)} \end{aligned}$$

or in the minimum-norm CLF-CBF QP with multiple CBFs below:

$$\begin{aligned} u^\star(x), \delta^\star(x) = \underset{(u,\delta) \in \mathbb{R}^{m+1}}{\text{argmin}} \ \|u\|^2 + p\delta^2 & \qquad (2.41) \\ s.t.\ L_f V(x) + L_g V(x) u \leq -\alpha(V(x)) + \delta & \qquad \text{(CLF)} \\ L_f h_i(x) + L_g h_i(x)\, u \geq -\beta_i(h_i(x)), \quad i \in \{1, \cdots, N\} & \qquad \text{(CBFs)} \end{aligned}$$

The presence of multiple independent CBF constraints in the optimization problem adds complexity to the shape of feasible set associated to the corresponding QP. Oven though these con-

straints are affine and define half-spaces in the QP decision space, the intersection of these constitutes a general polytope which could be empty. Therefore, when multiple CBF constraints are present, (2.40)-(2.41) could be unfeasible. The explicit conditions under which the QPs (2.40)-(2.41) are feasible are given in Theorem 5.1.1. This fact highlights that the multiple CBF case is much more challenging than the single CBF case.

Some works have proposed ways of composing multiple safety objectives. For example, [52] gives results about the composition of many CBFs. From their results, considering $N$ CBFs $h_1, \cdots, h_N$, a single *continuous* composite CBF can be defined by

$$h_c(x) = -\frac{1}{\kappa} \ln \left( \sum_{i=1}^{N} \exp^{-\kappa h_i(x)} \right) \tag{2.42}$$

where $\kappa$ is a positive parameter. Then, the following can be said about the safe set $\mathscr{C}_c$ of the composite CBF $h_c$ in (2.42):

$$\mathscr{C}_c = \{x \in \mathbb{R}^n \mid h_c(x) \geq 0\} \subset \mathscr{C} \tag{2.43}$$

That is, using a single barrier $h_c$ in QP (2.37) yields a slightly more conservative safe region then using $N$ CBFs $h_i$ in QP (2.41). The level of conservativeness is tunable by the parameter $\kappa > 0$, since $\lim_{\kappa \to \infty} \mathscr{C}_c = \bigcap_{i=1}^{N} \mathscr{C}_i$ [52].

In summary, Path-Following controllers and their variants, such as Moving and Cooperative Moving Path-Following (MPF and CMPF, respectively) can be implemented directly by means of Lyapunov control theory. As will be explored in the next chapter, Lyapunov theory can also be used to design robust versions of the Moving Path-Following control strategies, providing the robot with the capacity to compensate for unknown disturbances that could degrade control performance. Additionally, important safety-critical requirements, such as obstacle avoidance in robotic navigation, can be implemented using Control Barrier Functions (CBFs) and can even be combined with explicit stabilization objectives using Control Lyapunov Functions (CLFs). This topic will be addressed in Chapter 4. However, the combination of these two requirements often leads to trajectory deadlocks (i.e., stable equilibrium points), where the system state remains safe but fails to satisfy its stabilization objective. Chapter 5 presents strategies to address this problem.

# Chapter 3

# Moving Path Following Control with Robustness Guarantees

In this chapter, we present novel results in Moving Path Following (MPF) and Cooperative Moving Path Following (CMPF) control with robustness guarantees to external disturbances for an autonomous navigation task with a team of surface underwater vehicles. We propose two distinct robust MPF and CMPF control strategies:

i Robust MPF/CMPF control with a First Order Sliding Mode (FOSM) term.

ii Robust MPF/CMPF control with a FOSM term and with a disturbance observer.

These two strategies for robust MPF control can be applied both in the case of a single vehicle following a target or in case of a team of robot vehicles. In the latter case, cooperation on the team of robot vehicles is ensured through the use of a consensus module. Experiments with three Autonomous Underwater Vehicles (AUVs) were performed, and the obtained results are discussed by the end of the chapter.

## 3.1 Problem Formulation

Consider a team of $N$ robotic vehicles with states, control signals and perturbed kinematic models as introduced in Section 2.3.3. The set of integer indexes for each of the $N$ robotic vehicles is defined as $\mathbb{I} = \{1, 2, ..., N\}$. We assume each vehicle has an inner-loop autopilot controller responsible to generate adequate forces and torques to control the vehicles at the dynamic level, effectively tracking linear and angular velocity commands $v_{f_i}$ and $\omega_i$ generated by the outer loop controller respondible for controlling the robotic vehicle at the kinematic level. Imperfect tracking by the inner-loop autopilot controller can further contribute to the velocity disturbances acting on the vehicle, that can be lumped into the disturbance terms $d_{v_i}$ and $d_{\omega_i}$.

In the CMPF control problem, the vehicles must follow a priori specified paths expressed with respect to a moving target whose position can be accurately estimated, while also maintaining some coordination objective. Define the (moving) target frame $\mathscr{T}$ with its origin attached to the

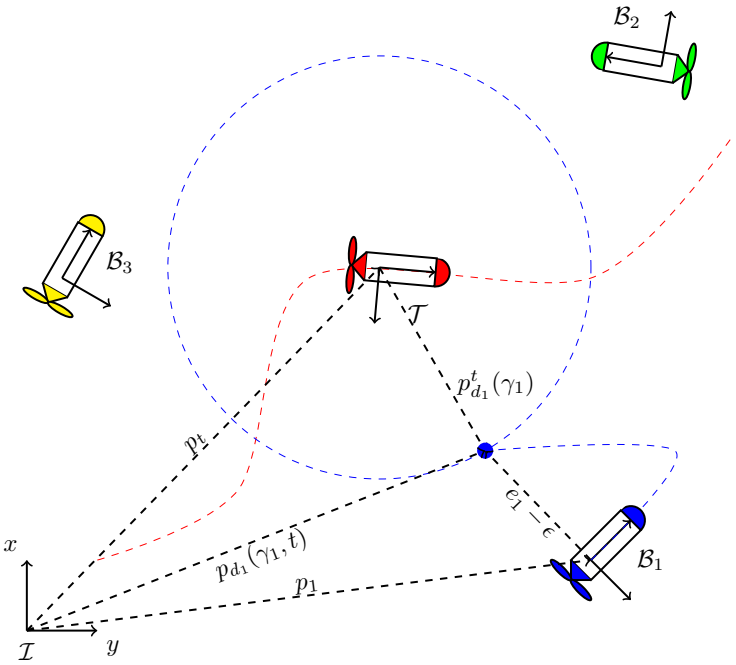



Figure 3.1: Coordinate frames and vector notation for $N = 3$ vehicles.

target center of mass. Then, the cooperative MPF problem can be divided in two sub-problems, namely (i) moving path following and (ii) cooperative control.

**Moving Path Following Problem.** Let $p_t(t) \in \mathbb{R}^n$ denote the position of the target with respect to the inertial frame $\mathscr{I}$, and $p^t_{d_i}(\gamma_i) \in \mathbb{R}^n$ be the desired path for the $i$-th vehicle, specified with respect to the local target frame $\mathscr{T}$ and parameterized by the path variable $\gamma_i \in \mathbb{R}$. As illustrated by Fig. 3.1, for a given $\gamma_i$ and time $t$, $p_{d_i}(\gamma_i, t)$ and $\dot{p}_{d_i}(\gamma_i, t)$ denote the position and velocity of the virtual reference point that must be followed by the $i$-th vehicle, with respect to the inertial frame $\mathscr{I}$:

$$p_{d_i}(\gamma_i, t) = p_t + R_t\, p^t_{d_i}(\gamma_i) \tag{3.1}$$

$$\dot{p}_{d_i}(\gamma_i, t) = v_t + R_t \left( \nabla p^t_{d_i}(\gamma_i)\dot{\gamma}_i + S(\omega_t)\, p^t_{d_i}(\gamma_i) \right) \tag{3.2}$$

where $R_t(t) \in \mathbb{SO}(n)$ is the rotation matrix of frame $\mathscr{T}$ with respect to $\mathscr{I}$ and $v_t(t) \in \mathbb{R}^n$, $\omega_t(t) \in \mathbb{R}^{n(n-1)/2}$ are the linear and angular target velocities.

Suppose we wish to control the position of the nose of the $i$-th vehicle, or more generically, an offset point from its center of mass $p_{\varepsilon_i} = p_i + R_i\, \varepsilon$, as defined in Section 2.28. Then, define the MPF error in body coordinates $\mathscr{B}_i$ as

$$e_i = R_i^{\mathsf{T}} \left( p_{\varepsilon_i} - p_{d_i}(\gamma_i, t) \right), \quad i \in \mathbb{I} \tag{3.3}$$

similarly to (2.26), but with a time-varying path $p_{d_i}(\gamma_i, t)$ due to the motion of the target as seen in

(3.1). The objective of the MPF control problem is to design a control law $u_i$ for the $i$-th vehicle such that the origin $e_i = 0$ is stable and $e_i \to 0$ as $t \to \infty$, $\forall i \in \mathbb{I}$. That is, it is desired to steer the vehicles towards their moving geometric paths, such that each vehicle offset position $p_{\varepsilon_i}$ stabilizes around $p_{d_i}(\gamma_i, t)$, $\forall i \in \mathbb{I}$.

In order to control the progression of the virtual points $p_{d_i}(\gamma_i, t)$ along the moving paths, the dynamics of the path variable $\dot{\gamma}_i$ should be explicitly controlled. This can be achieved by imposing the following dynamics for $\gamma_i$:

$$\dot{\gamma}_i = v_d + \vartheta_i, \quad i \in \mathbb{I} \tag{3.4}$$

where the scalar $v_d$ is the desired nominal speed for the path variable and $\vartheta_i$ is a bounded control signal, designed to achieve secondary objectives, such as: (i) faster convergence to the moving path and (ii) consensus over the path variables of the robotic vehicles to achieve a desired formation along the moving path. To move along the geometric paths with the desired velocity, the vehicles must satisfy the desired speed assignments $|\dot{\gamma}_i - v_d| \to 0$ as $t \to \infty$, $\forall i \in \mathbb{I}$.

**Cooperative Control Problem.** Assume that the $i$-th vehicle communicates with a fixed set $\mathbb{N}_i \subset \mathbb{N}$ of neighbor vehicles. Given the path variables $\gamma_i$ ($i \in \mathbb{I}$) for the $N$ vehicles and a given undirected, fixed communication topology among them, the objective of the cooperative motion control problem is to design a decentralized control law $v_{r_i}(t)$ such that the positions of the virtual points are synchronized, that is, $|\gamma_i - \gamma_j|$ converges to zero $\forall i, j \in \mathbb{I}$ as $t \to \infty$. To simultaneously achieve the speed assignment, coordination objective and also two other secondary objectives, function $\vartheta_i$ in (3.4) is decomposed as

$$\vartheta_i = v_{r_i}(t) + g_{e_i}(t) + g_{\omega_i}(t) \tag{3.5}$$

where $v_{r_i}(t)$ is the cooperative control signal (to be designed in the next section) that is responsible for achieving consensus between the vehicles, while $g_{e_i}(t)$ and $g_{\omega_i}(t)$ represent secondary objectives, where $g_{e_i}(t)$ is an error correction term, responsible for delaying the evolution of the path variable in case of momentary vehicle failure, and $g_{\omega_i}(t)$ is a rotation correction term, responsible for canceling the rotational motion induced on the path by the rotation of the target. These functions are to be properly defined in Section 3.2.3.

## 3.2 Robust Cooperative MPF Control Design

In this section, we present two distinct strategies for achieving robustness against disturbances in MPF control. In Section 3.2.1, we present a robust MPF controller using a sliding mode control strategy. In Section 3.2.2, we present a robust MPF controller using a disturbance observer for estimating the magnitude of the disturbances, and using this data to improve the controller performance. Additionally, in Section 3.2.3, we present the consensus controller used to solve the cooperative control problem for the team of robotic vehicles.

### 3.2.1 Robust MPF Control with Sliding Mode Term

In this section, we consider the kinematic controller proposed in [34] with a modification designed to ensure robustness against disturbances. For the $i$-th vehicle, the error dynamics is given by

$$\dot{e}_i = \dot{R}_i^{\mathsf{T}}(p_{\varepsilon_i} - p_{d_i}) + R_i^{\mathsf{T}}(\dot{p}_{\varepsilon_i} - \dot{p}_{d_i}) \tag{3.6}$$

Using model (2.20) with control signal (2.22) and MPF error (3.3), (3.6) can be rewritten as

$$\dot{e}_i = -S(\omega_i + d_{\omega_i})\, e_i + \Delta_\varepsilon u_i + d_i - R_i^{\mathsf{T}} v_t - R_i^{\mathsf{T}} R_t \left(\nabla p_{d_i}^t(\gamma_i)\, \dot{\gamma}_i + S(\omega_t) p_{d_i}^t(\gamma_i)\right) \tag{3.7}$$

with matrix $\Delta_\varepsilon$ as defined in (2.25). Vector $d_i \in \mathbb{R}^n$ is the total external disturbance acting on the $i$-th vehicle. In the planar case, it is given by

$$d_i = \begin{bmatrix} R_i^{\mathsf{T}} & s_\varepsilon \end{bmatrix} \begin{bmatrix} d_{v_i} \\ d_{\omega_i} \end{bmatrix}, \qquad s_\varepsilon = \begin{bmatrix} -\varepsilon_y \\ \varepsilon_x \end{bmatrix} \tag{3.8}$$

**Remark.** Notice that, by the triangle inequality, the total external disturbance $d_i$ is bounded by $\|d_i\| \leq \|d_{v_i}\| + \|d_{\omega_i}\|\, \|\varepsilon\|$.

**Assumption 2.** The norm of the total external disturbance $d_i$ for the $i$-th vehicle is bounded.

**Theorem 3.2.1** (Robust MPF with Sliding Mode Controller)**.** Consider an underactuated robotic vehicle described by with control signal given by (2.22). Let the MPF error kinematics be described by (3.7), and assume that the pose of the $i$-th vehicle $\{p_i, R_i\} \in \mathbb{R}^n \times \mathbb{SO}(n)$ and of the target frame $\{p_t, R_t\} \in \mathbb{R}^n \times \mathbb{SO}(n)$ are known. Under Assumptions 1 and 2, the feedback control law

$$u_i(e_i) = \Delta_\varepsilon^\dagger \left(-K_p e_i + R_i^{\mathsf{T}} \left(\widehat{v}_t + R_t\, S(\widehat{\omega}_t)\, p_{d_i}^t\right) + R_i^{\mathsf{T}} R_t \nabla p_{d_i}^t(\gamma_i) \dot{\gamma}_i - w_i(e_i)\right) \tag{3.9}$$

$$w_i(e_i) = \begin{cases} \rho_i \frac{e_i}{\|e_i\|} & , \quad \|e_i\| \geq \varepsilon_w \\ \rho_i \frac{e_i}{\varepsilon_w} & , \quad \|e_i\| < \varepsilon_w \end{cases} \tag{3.10}$$

ensures that all trajectories of the MPF error are globally uniformly ultimately bounded and converge to a ball around the origin $e_i = 0$ that can be made arbitrarily small. In (3.9), the matrix $\Delta_\varepsilon^\dagger$ is the Moore-Penrose pseudo-inverse of $\Delta_\varepsilon$, $K_p \in \mathbb{R}^{n \times n}$ is a positive-definite gain matrix and $\widehat{v}_t \in \mathbb{R}^n$, $\widehat{\omega}_t \in \mathbb{R}^{n(n-1)/2}$ are estimates of the target velocities. In (3.10), $\rho_i$ is a scalar such that

$$\rho_i \geq \|d_{v_i}\| + \|d_{\omega_i}\|\, \|\varepsilon\| + \|\widetilde{v}_t\| + \|\widetilde{\omega}_t \times p_{d_i}^t(\gamma_i)\| \tag{3.11}$$

where $\widetilde{v}_t = v_t - \widehat{v}_t$, $\widetilde{\omega}_t = \omega_t - \widehat{\omega}_t$ are bounded estimation errors on the target velocities, and $\varepsilon_w > 0$ is a small positive scalar.

*Proof.* Define the Lyapunov candidate $V(e_i) = \frac{1}{2}\|e_i\|^2$. Using the error dynamics in (3.7), its time derivative along the system trajectories is

$$\dot{V}(e_i) = e_i^{\mathsf{T}} \left( \Delta_\varepsilon u_i + d_i - R_i^{\mathsf{T}} v_t - R_i^{\mathsf{T}} R_t \left( \nabla p_{d_i}^t(\gamma_i)\dot{\gamma}_i + S(\omega_t) p_{d_i}^t(\gamma_i) \right) \right) \tag{3.12}$$

where $e_i^{\mathsf{T}} S(\omega_i + d_{\omega_i}) e_i = 0$ was used, since $S(\omega_i + d_{\omega_i})$ is skew-symmetric. Substituting control law (3.9) in (3.12) yields

$$\dot{V}(e_i) = -e_i^{\mathsf{T}} K_p e_i + e_i^{\mathsf{T}} \left( D_i - w_i \right), \tag{3.13}$$

where $D_i = d_i - R_i^{\mathsf{T}} \left( \widetilde{v}_t + R_t S(\widetilde{\omega}_t) p_{d_i}^t \right)$. Since $K_p > 0$, the first term is negative definite and bounded by $-\sigma_{\min}(K_p)\|e_i\|^2$. Next, we consider the two cases of (3.10), when $\|e_i\| \geq \varepsilon_w$ or $\|e_i\| < \varepsilon_w$.

- For $\|e_i\| \geq \varepsilon_w$ in (3.13), we have

$$\begin{aligned} \dot{V}(e_i) &\leq -\sigma_{\min}(K_p)\|e_i\|^2 + e_i^{\mathsf{T}} D_i - \rho_i \frac{e_i^{\mathsf{T}} e_i}{\|e_i\|} \\ &\leq -\sigma_{\min}(K_p)\|e_i\|^2 + \|e_i\| \left( \|D_i\| - \rho_i \right) \end{aligned}$$

  where the Cauchy-Schwarz inequality was employed on term $e_i^{\mathsf{T}} D_i$. By Assumption 2, it is always possible to design $\rho_i$ such that (3.11) is satisfied. Therefore, by Remark 3.2.1, choosing $\rho_i \geq \|D_i\|$ renders the second term on the right-hand side negative definite, which estabilishes that the trajectory $e_i(t)$ of the closed-loop system reaches the ball $\mathscr{B}_{\varepsilon_w}(0) := \{e_i \in \mathbb{R}^n : \|e_i\| \leq \varepsilon_w\}$ in finite time.

- When the trajectories are inside $\mathscr{B}_{\varepsilon_w}(0)$, we have $\|e_i\| < \varepsilon_w$, and (3.13) gets

$$\begin{aligned} \dot{V}(e_i) &\leq -\sigma_{\min}(K_p)\|e_i\|^2 + e_i^{\mathsf{T}} D_i - \rho_i \frac{e_i^{\mathsf{T}} e_i}{\varepsilon_w} \\ &\leq -(1-\theta)\sigma_{\min}(K_p)\|e_i\|^2 - \left( \theta \sigma_{\min}(K_p) + \frac{\rho_i}{\varepsilon_w} \right) \|e_i\|^2 + \|e_i\| \|D_i\| \end{aligned}$$

  where $0 < \theta < 1$. Then, using the inequality above, one can write:

$$\dot{V}(e_i) \leq -(1-\theta)\sigma_{\min}(K_p)\|e_i\|^2 < 0 \quad \forall \|e_i\| \geq \mu_i, \quad \mu_i = \frac{\|D_i\| \varepsilon_w}{\sigma_{\min}(K_p)\, \theta\, \varepsilon_w + \rho_i}$$

  Note that $\mu_i \leq \varepsilon_w$ for all $0 < \theta < 1$, which means that the trajectory of the closed-loop system $e_i(t)$ again reaches the ball $\mathscr{B}_{\mu_i}(0) \subseteq \mathscr{B}_{\varepsilon_w}(0)$ in finite time.

This establishes that the trajectories are globally ultimately uniformly bounded, since $V = \frac{1}{2}\|e_i\|^2$ is radially unbounded ([41]). Moreover, $e_i(t)$ converges to the ball $\mathscr{B}_{\mu_i}(0) \subseteq \mathscr{B}_{\varepsilon_w}(0)$, which can be made arbitrarily small when $\varepsilon_w \to 0$. □

Using similar arguments, [34] proves that the control law (3.9) with $w_i(e_i) = 0$ (no robustness term) and $\widehat{v}_t = v_t$ makes the system Input-to-State-Stable (ISS) with respect to the total disturbance in (3.11) (see Definition 2.2.2). In this case, $e_i$ converges to some bound of the order of the total disturbance. However, using (3.10), the scalar $\varepsilon_w$ can be designed to be much smaller than the total amount of disturbance, effectively compensating it.

### 3.2.2 Robust MPF Control with Sliding Mode Term and Disturbance Observer

In the presence of large amplitude disturbances, it may be difficult to tune the parameters $\rho_i$ and $\varepsilon_w$ so as to satisfy (3.11). In these situations, an observer can be designed to provide an estimate of the disturbance. Furthermore, this estimate can be used in the control law to compensate the real disturbance directly.

Without loss of generality, consider the planar problem. Consider that the vehicle pose $(p_i, R_i) \in \mathbb{R}^2 \times \mathbb{SO}(2)$ is known and that the vehicle orientation is parameterized by the planar angle $\phi_i \in [-\pi, \pi]$, such that $R_i = R_i(\phi_i) \in \mathbb{SO}(2)$. Then, the disturbance observer for the linear velocity disturbance is defined as

$$\begin{cases} \dot{\hat{p}}_i = R_i v_i + \hat{d}_{v_i} + K_1 \widetilde{p}_i \\ \dot{\hat{d}}_{v_i} = K_2 \widetilde{p}_i \end{cases} \tag{3.14}$$

where the estimation errors are defined as $\widetilde{p}_i = p_i - \hat{p}_i$ and $\widetilde{d_{v_i}} = d_{v_i} - \hat{d}_{v_i}$, and the positions $p_i$, $i \in \mathbb{I}$ are assumed to be accurately measured. For positive-definite matrices $K_1, K_2 \in \mathbb{R}^{2\times 2}$, the dynamics of the estimation errors $\widetilde{p}_i$, $\widetilde{d}_i$ can be proven to be Input-to-State Stable (ISS) with respect to the first time-derivative of $d_{v_i}$ [4].

Similarly, observers for the angular velocity disturbances $d_{\omega_i} \in \mathbb{R}$ can be designed as:

$$\begin{cases} \dot{\hat{\phi}}_i = \omega_i + \hat{d}_{\omega_i} + k_{\omega_1} \widetilde{\phi}_i \\ \dot{\hat{d}}_{\omega_i} = k_{\omega_2} \widetilde{\phi}_i \end{cases} \tag{3.15}$$

where the estimation errors are defined as $\widetilde{\phi}_i = \phi_i - \hat{\phi}_i$ and $\widetilde{d_{\omega_i}} = d_{\omega_i} - \hat{d}_{\omega_i}$, and the planar angles $\phi_i$ are assumed to be accurately measured. Again, for positive scalars $k_{\omega_1}, k_{\omega_2} \in \mathbb{R}_{>0}$, the dynamics of the estimation errors $\widetilde{\phi}_i$, $\widetilde{d}_{\omega_i}$ can be proven to be ISS with respect to the first time-derivative of $d_{\omega_i}$ [4].

**Theorem 3.2.2 (Robust MPF with Disturbance Observer).** Consider an underactuated robotic vehicle described by (2.20) and control signal given by (2.22). Let the MPF error kinematics be described by (3.7), and consider that the pose of the vehicle $\{p_i, R_i\} \in \mathbb{R}^n \times \mathbb{SO}(n)$ and of the target frame $\{p_t, R_t\} \in \mathbb{R}^n \times \mathbb{SO}(n)$ are known. Under Assumptions 1 and 2, the control law

$$u_i = \Delta_\varepsilon^\dagger \left( -K_p e_i + R_i^\mathsf{T} \left( \widehat{v}_t + R_t S(\widehat{\omega}_t) p_{d_i}^t \right) + R_i^\mathsf{T} R_t \nabla p_{d_i}^t(\gamma_i)\, \dot{\gamma}_i - w_i - \hat{d}_i \right) \tag{3.16}$$

ensures that all trajectories of the MPF error are globally uniformly ultimately bounded and converge to a ball centered on the origin $e_i = 0$ that can be made arbitrarily small. In (3.16), matrix $\Delta_\varepsilon^\dagger$ is the Moore-Penrose pseudo-inverse of $\Delta_\varepsilon$, $K_p \in \mathbb{R}^{n\times n}$ is a positive-definite gain matrix, $\widehat{v}_t \in \mathbb{R}^n$ and $\widehat{\omega}_t \in \mathbb{R}^{n(n-1)/2}$ are estimates of the target velocities and $\hat{d}_i$ is the total estimated external disturbance, which is a function of the states of the disturbance observers

$$\widehat{d}_i = \begin{bmatrix} R_i^\mathsf{T} & s_\varepsilon \end{bmatrix} \begin{bmatrix} \hat{d}_{v_i} \\ \hat{d}_{\omega_i} \end{bmatrix}, \qquad s_\varepsilon = \begin{bmatrix} -\varepsilon_y \\ \varepsilon_x \end{bmatrix} \tag{3.17}$$

The sliding mode control term $w_i(e_i)$ is defined by (3.10), with scalars $\rho_i$ satisfying

$$\rho_i \geq \|\widetilde{d}_{v_i}\| + |\widetilde{d}_{\omega_i}|\|\varepsilon\| + \|\widetilde{v}_t\| + \|\widetilde{\omega}_t \times p_{d_i}^t(\gamma_i)\| \tag{3.18}$$

*Proof.* The proof is very similar to Theorem 3.2.1, and can be performed by proposing the same Lyapunov candidate $V = \frac{1}{2}\|e_i\|^2$. Differentiating it in time and applying the error dynamics (3.7) with control law (3.9) yields

$$\dot{V}(e_i) = -e_i^\mathsf{T} K_p\, e_i + e_i^\mathsf{T} \left(\widetilde{D}_i - w_i\right) \tag{3.19}$$

where $\widetilde{D}_i = \widetilde{d}_i - R_i^\mathsf{T}\left(\widetilde{v}_t + R_t\, S(\widetilde{\omega}_t)\, p_{d_i}^t\right)$ and $\widetilde{d}_i$ is the total estimation error defined by $\widetilde{d}_i = d_i - \hat{d}_i$. Note that (3.19) is similar to (3.13), but with disturbance $\widetilde{D}_i$ instead of $D_i$. Therefore, using the same arguments for the proof of Theorem 3.2.1 with Assumption 2 and condition (3.18), one can conclude that the trajectories of the MPF error are globally uniformly ultimately bounded and $e_i(t)$ converges to the ball $\mathscr{B}_{\bar{\mu}_i}(0) \subseteq \mathscr{B}_{\varepsilon_w}(0)$, which can be made arbitrarily small when $\varepsilon_w \to 0$. □

**Remark.** Comparing conditions (3.11) and (3.18) for the choice of $\rho_i$, in (3.18) the gain $\rho_i$ must overcome only the norm of the disturbance estimation errors instead of the norm of the disturbance. Therefore, if the disturbance observer is properly designed, this method can reduce the necessary amount of control effort when compared to the previous method.

Both proposed control laws (3.9) and (3.16) employ estimates of the target velocities. Since the velocity estimation errors $\widetilde{v}_t$ and $\widetilde{\omega}_t$ appear as additional disturbances in $D_i$ and $\widetilde{D}_i$, they can be properly compensated by the proposed controllers as long as $\rho_i$ satisfies (3.11) or (3.18). In this case, the velocity estimation errors are implicitly assumed to be bounded. Furthermore, notice that in the case where no velocity estimators are employed ($\widehat{v}_t = 0$ and $\widehat{\omega}_t = 0$), the velocity estimation errors are simply $\widetilde{v}_t = v_t$ and $\widetilde{\omega}_t = \omega_t$, which are also bounded. These observations imply that velocity estimators are not absolutely required for the implementation of the proposed control laws. However, large velocity estimation errors would increase the lower bounds for the design of $\rho_i$, increasing the amount of control effort, which could lead to loss of performance. As a final remark, control laws (3.9) and (3.16) are essentially robust versions of the path following control law (2.28) with compensations for the velocities of the target and kinematic disturbances.

### 3.2.3 Cooperative Moving Path Following

Imposing specific dynamics for the path variables $\gamma_i$ allows to explicitly control the progression of the virtual points $p_{d_i}(\gamma_i,t)$ along the moving paths. As seem from (3.4), this dynamics imposes speed assignments for the robots and can also be used to impose secondary control objectives. This section deals with these secondary objetives, each one expressed by one of the terms in (3.5), aiming at achieving (i) faster convergence of the robotic vehicle to the moving path (the error correction term $g_{e_i}(t)$), (ii) minimizing the effect of the target rotational motion (rotation correction term $g_{\omega_i}(t)$) and (iii) achieving multi-robot cooperation through consensus (cooperative control law $v_{r_i}(t)$).

**Error correction term.** The term $g_{e_i}(t)$ is a bounded error correction term that acts as an external input to the path dynamics, enabling faster convergence of the robotic vehicle to the moving path. It can be designed to delay or to stop the motion of the virtual point if the vehicle is too far away from the path. This can be done by defining the gradient with respect to the path variable of the MPF error norm squared:

$$\eta_{e_i} = \frac{\partial}{\partial \gamma_i}\left(\frac{1}{2}\|e_i\|^2\right) = -e_i^{\mathsf{T}} R_i^{\mathsf{T}} R_t \nabla p_{d_i}^t(\gamma_i) \tag{3.20}$$

and then choosing a gradient descent law $g_{e_i} = -k_e\, sat(\eta_{e_i})$ with $k_e > 0$. The saturation function guarantees the boundedness for the correction term. Its effect is to effectively delay the evolution of the virtual point along the path by explicitly avoiding the evolution of $\gamma_i$ if the MPF error norm is too large.

**Path rotation correction term.** The term $g_{\omega_i}(t)$ is designed to delay the evolution of the virtual point $p_{d_i}$ in a such a way that minimizes the effect of the target rotational motion, which is evident from the term $S(\omega_t)\, p_{d_i}^t(\gamma_i)$ in (3.2). This effect is important since, for large target angular velocities $\omega_t$, the virtual point could move faster than the $i$-th vehicle could reach. Therefore, substituting (3.5) into the error dynamics (3.7), we seek to design a scalar $g_{\omega_i}$ such that

$$g_{\omega_i}(t) = \underset{g_i \in \mathbb{R}}{\operatorname{argmin}} \left\| \nabla p_{d_i}^t(\gamma_i) g_i + S(\omega_t) p_{d_i}^t(\gamma_i) \right\| \tag{3.21}$$

If the target angular velocity is known, the minimum can be achieved from the least squares solution:

$$g_{\omega_i}(\omega_t, \gamma_i) = -\frac{1}{\|\nabla p_{d_i}^t(\gamma_i)\|^2}\left(\nabla p_{d_i}^t(\gamma_i)^{\mathsf{T}} S(\omega_t)\, p_{d_i}^t(\gamma_i)\right) \tag{3.22}$$

**Assumption 3.** The path gradients are non-vanishing everywhere, i.e., $\nabla p_{d_i}^t(\gamma_i) \neq 0, \forall \gamma_i, i \in \mathbb{I}$.

From (3.22) and Assumption 3, the error correction term is bounded by

$$|g_{\omega_i}(t)| \leq \frac{\max_{\gamma_i}\|p^t_{d_i}(\gamma_i)\|}{\min_{\gamma_i}\|\nabla p^t_{d_i}(\gamma_i)\|}\|\omega_t\|\,.$$

**Cooperative Controller.** Consider the distributed consensus law [1]:

$$v_{r_i} = -k_{c_i}\sum_{j\in\mathbb{N}_i}(\gamma_i - \hat{\gamma}^i_j)\,, \qquad \forall i \in \mathbb{I} \tag{3.23}$$

where $k_{c_i} > 0$ are consensus gains and $\hat{\gamma}^i_j$ are estimates of the path variables of the neighbor vehicles $(\gamma_j\,, j \in \mathbb{N}_i)$ running inside the $i$-th vehicle computer. Assuming that the frequency of communication is low, its reasonable to assume that $\hat{\gamma}^i_j \neq \gamma_j\,, \forall t > 0$. Therefore, one can write $\hat{\gamma}^i_j = \gamma_j - \tilde{\gamma}^i_j$, where $\tilde{\gamma}^i_j$ is a path variable estimation error.

**Assumption 4.** Given a fixed, undirected communication topology between the vehicles, the $i$-th vehicle updates its path variable $\gamma_i$ to its $j \in \mathbb{N}_i$ neighbors in a fixed frequency. Additionally, assume that no data package is lost during communication. Consequently, the path variable estimation errors $\tilde{\gamma}^i_j\,, \forall i, j \in \mathbb{I}$ are always bounded.

Define the vectors $\gamma^\mathsf{T} = \begin{bmatrix}\gamma_1 & \gamma_2 & \cdots & \gamma_N\end{bmatrix}$, $g_e^\mathsf{T} = \begin{bmatrix}g_{e_1} & g_{e_2} & \cdots & g_{e_N}\end{bmatrix}$, $g_\omega^\mathsf{T} = \begin{bmatrix}g_{\omega_1} & g_{\omega_2} & \cdots & g_{\omega_N}\end{bmatrix}$ and $\mathbf{1}_N = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^\mathsf{T} \in \mathbb{R}^N$. Using (3.23) in (3.5) and stacking the dynamic equations yields

$$\dot{\gamma} = v_d\mathbf{1}_N - K_c L\gamma - K_c\tilde{\gamma} + g_e + g_\omega\,, \tag{3.24}$$

where $K_c = diag(k_{c_1}, k_{c_2}, \ldots, k_{c_N})$ is a positive definite matrix of consensus gains, $L = D - A \in \mathbb{R}^{N\times N}$ is the *Laplacian* of the network connection graph, defined by $D = diag(|\mathbb{N}_1|, |\mathbb{N}_2|, \ldots, |\mathbb{N}_N|)$ and the *adjacency matrix* $A = [a_{ij}]$, with $a_{ij} = 1$ if $j \in \mathbb{N}_i$ and $a_{ij} = 0$ otherwise. Vector $\tilde{\gamma}$ is defined as $[\tilde{\gamma}]_i = \sum_{j\in\mathbb{N}_i}\tilde{\gamma}^i_j$, i.e, its $i$-th element is the sum of all path variable estimation errors for the $i$-th vehicle.

**Theorem 3.2.3 (Cooperative Controller).** Consider a fleet of $N$ underactuated robotic vehicles with dynamics described by (2.20) and control signal given by (2.22). Then, control laws (3.9) or (3.16) with robustness term (3.10) guarantee that the origin of the $i$-th MPF error $e_i = 0$ is stable under the same conditions and assumptions of Theorems 3.2.1 and 3.2.2, respectively. Furthermore, under Assumption 4, the cooperative control law given by (3.23) ensures that $|\gamma_i - \gamma_j|, \forall i, j \in \mathbb{I}$ are Input-to-State Stable (ISS) (see Definition 2.2.2) with respect to the path variable estimation errors $[\tilde{\gamma}]_i$, error correction terms $g_{e_i}$ and rotation correction terms $g_{\omega_i}$, $\forall i \in \mathbb{I}$.

*Proof.* The first part of the Theorem was already proven in Theorems 3.2.1 and 3.2.2. The part related to the cooperative control follows the same core ideas from [34]. First, define the *disagreement vector* ([55]) as $\delta \equiv \gamma - \alpha\mathbf{1}_N$, with $\alpha = (1/N)\mathbf{1}_N^\mathsf{T}\gamma$. Note that the consensus condition $|\gamma_i - \gamma_j| = 0, \forall i, j \in \mathbb{I}$ is achieved if and only if $\delta = 0$. Additionally, the following two properties

hold: (i) $L\gamma = L\delta$ and (ii) $\mathbf{1}_N^\mathsf{T}\delta = 0$. Next, define the ISS Lyapunov function candidate

$$V_{cc}(\delta) = \delta^\mathsf{T} L\delta \geq 0$$

Taking its time-derivative and using (3.24), yields

$$\dot{V}_{cc} = -z^\mathsf{T} K_c z - z^\mathsf{T} K_c \tilde{\gamma} + z^\mathsf{T} g_e + z^\mathsf{T} g_\omega \tag{3.25}$$

with $z = L\delta$, where we used the properties (i) and (ii) introduced before. Using the Cauchy-Schwartz inequality, yields

$$\dot{V}_{cc} \leq -\sigma_{\min}(K_c)\|z\|^2 + \sigma_{\max}(K_c)\|z\|\,\|\tilde{\gamma}\| + \|z\|\,\|g_e\| + \|z\|\,\|g_\omega\| \tag{3.26}$$

Applying Young's inequality to the last three terms in (3.26), we have

$$\dot{V}_{cc} \leq -\left(\sigma_{\min}(K_c) - \frac{\sigma_{\max}(K_c)}{2c} - \frac{1}{2c} - \frac{1}{2c}\right)\|z\|^2 + \frac{c\sigma_{\max}(K_c)}{2}\|\tilde{\gamma}\|^2 + \frac{c}{2}\|g_e\|^2 + \frac{c}{2}\|g_\omega\|^2$$

with a scalar $c \in \mathbb{R}_{>0}$. Choosing any $c > \frac{\sigma_{\max}(K_c)}{2\sigma_{\min}(K_c)} + \sigma_{\min}^{-1}(K_c) > 0$ leaves the first term of the right-hand side strictly negative, which by Assumption 4 and by the boundedness of $g_{e_i}$, $g_{\omega_i}$ establishes that the disagreement vector $\delta$ is ISS with respect to the bounded disturbances $[\tilde{\gamma}]_i$, $g_{e_i}$ and $g_{\omega_i}$, $\forall i \in \mathbb{I}$ (see Definition 2.2.2).

□

## 3.3 Experimental Results with Underwater Vehicles

The proposed control strategies were tested and validated through experiments performed on Porto de Leixões (Porto, Portugal), using three Light Autonomous Underwater Vehicles (LAUVs) from the Underwater Systems and Technology Laboratory at the Faculty of Engineering of the University of Porto (FEUP), shown in Fig. 3.2. LAUVs are lightweight, portable vehicles that can be easily launched, operated and recovered with a minimal operational setup. The vehicles operate under the DUNE/Neptus environments, which are part of a software toolchain [60] developed and maintained by LSTS. DUNE is the on-board software running on the vehicles, comprising all the software needed for communications, navigation, control, maneuvering, plan execution and supervision of multiple types of robotic vehicles. The control algorithms were implemented on C++, using the available DUNE libraries. Neptus is a software used for command, control and monitoring, comprising many typical functions needed for a typical mission, such as planning, execution and post-mission analysis. Its user interface is shown in Fig. 3.3.

The experiments are divided into MPF and Cooperative MPF (CMPF) experiments. The MPF experimental results are presented at Section 3.3.2 and consist of: (i) nominal MPF without disturbance rejection, (ii) robust MPF with sliding mode term and (iii) robust MPF with sliding mode term and disturbance observer. The CMPF experimental results are presented at Section 3.3.3 and


Figure 3.2: Autonomous underwater vehicles used in the experiments.

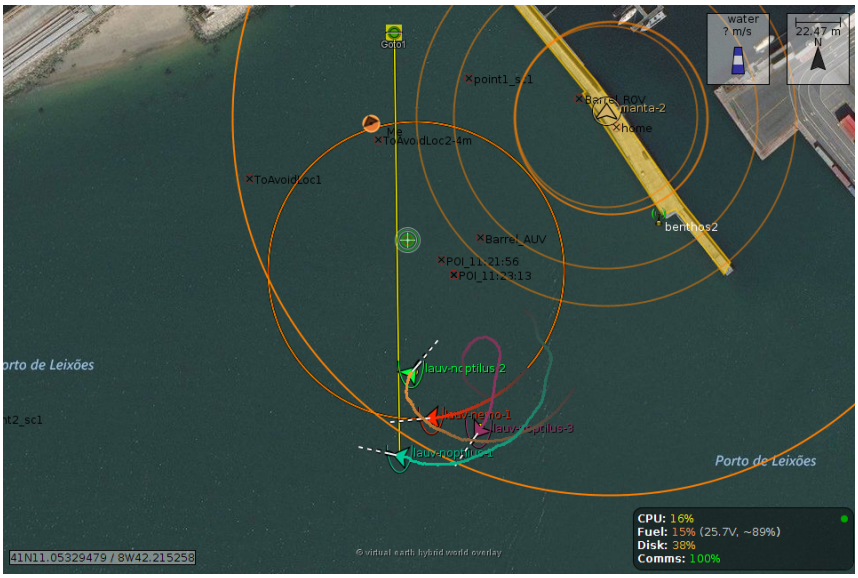



Figure 3.3: Neptus console.

consist of: (i) nominal CMPF without disturbance rejection, (ii) robust CMPF with sliding mode term and (iii) robust CMPF with sliding mode term and disturbance observer.

In all experiments, a target vehicle was *simulated* and continuously sends its position and orientation (computed from GPS/IMU measurements using an extended Kalman filter [13] to the follower vehicles through static UDP connections with a maximum frequency of $1\,Hz$. The control algorithm for the target vehicle is a vector field method [53] that is responsible to steer the vehicle along a circumference with radius equal to $60\,m$ in the clockwise direction at $0.5\,m/s$. With one single vehicle (MPF experiments) or with multiple vehicles (CMPF experiments), the desired

moving paths for the follower AUVs are circles centered at the target vehicle with phase difference of $2\pi/3$ between them:

$$p_{d_i}^t(\gamma_i) = R \begin{bmatrix} \cos(\gamma_i/R + \phi_i) \\ \sin(\gamma_i/R + \phi_i) \end{bmatrix} \tag{3.27}$$

where $R = 25\,m$, $\phi_1 = 0\,rad$, $\phi_2 = 2\pi/3\,rad$ and $\phi_3 = -2\pi/3\,rad$. In the CMPF experiments, each vehicle from the team sends its path variable to the neighbor vehicles with a frequency of $1\,Hz$ to maintain coordination, according to the consensus law (3.23) and Assumption 4. The consensus gains are $k_{c_i} = 0.1\,, \forall i \in \mathbb{I}$.

For the construction of the MPF errors $e_i$, the value $\varepsilon = [\,1 \quad 0\,]^\mathsf{T}$ was used. The controller gain matrices and error correction gains were chosen as $K_p = \text{diag}(0.2, 0.2)$ and $k_e = 2\,, \forall i \in \mathbb{I}$. The reference for the path variable velocity is $v_d = 1\,m/s$.

### 3.3.1 Kalman Filter for Target Velocity Estimation

To achieve accurate estimates for the target velocities, a simple linear Kalman filter was designed. Considering the planar case only, the target can be approximated as moving rigid body with a constant velocity, while its acceleration is considered a random variable drawn from a Gaussian distribution. Under this assumption, the target motion dynamics is given by the linear model:

$$\dot{p}_t = v_t\,, \dot{\phi}_t = \omega_t\,, \dot{v}_t = a_t\,, \dot{\omega}_t = \alpha_t \tag{3.28}$$

where $\phi_t$ is the target orientation angle on the plane, and $a_t$, $\alpha_t$ are the target linear and angular accelerations drawn from Gaussian distribuitions with zero mean and covariances $\Sigma_p^2 = diag(0.1, 0.1)$, $\sigma_p^2 = 10^{-4}$, respectively. The linear model (3.28) can be discretized using a zero order holder (ZOH), yielding a discrete linear, time-invariant model whose states are the target position, angle and velocities at each sample time. The measured outputs are the target position and orientation angle, sent through Wi-fi to the follower vehicle with a maximum frequency of $1\,Hz$.

The Kalman filter is implemented according to the usual *predict/update* strategy. Every control sample time, a new prediction is computed using the current target system state, and the update step is only computed when a new Wifi message containing a position/angle update is received from the target vehicle. The measure noise covariances were chosen as $\Sigma_m^2 = diag(0.1, 0.1)$ and $\sigma_m^2 = 10^{-4}$. The initial state for the target state covariances are: $2m$ for the position, $1\,rad$ for the orientation, $0.1\,m/s$ and $0.1\,rad/s$ for the linear and angular velocities.

**Remark.** We point out the fact that this particular kind of vehicles cannot generate reliable negative forward velocities due to their propeller design. Given the fact that control laws (3.9), (3.16) can generate negative forward velocities if the virtual point is behind the line-of-sight of the vehicle, a substitute controller was designed to override the original controller in case this happens. Therefore, while the forward velocity generated by (3.9) or (3.16) is negative ($v_{f_i} < 0$), the applied

control signal will be

$$u_i = \begin{bmatrix} v_C \\ -\operatorname{sign}[e_i]_y - \varepsilon_y \omega_C \end{bmatrix}$$

instead, until controls (3.9) or (3.16) generate a positive $v_{f_i}$ again. Constants $v_C$, $\omega_C \in \mathbb{R}$ are strictly positive. That means that the vehicle performs a "turning" maneuver with constant velocities until the virtual point is once again inside its line-of-sight. The direction of the turn is clockwise if the virtual point is to the right of the vehicle and counterclockwise if the virtual point is to the left of the vehicle. This strategy allows arbitrary initial configurations of the vehicles with respect to the initial position of the virtual point, and also allows the vehicles to recover from practical deadlock situations where their line-of-sight is kept facing away from the virtual point, which could happen, for example, in case of communication losses. In this case, $v_C = 1.7\,m/s$ and $\omega_C = 1\,rad/s$, which are approximately the upper saturation limits for the actuators.

### 3.3.2 Robust Moving Path Following Control

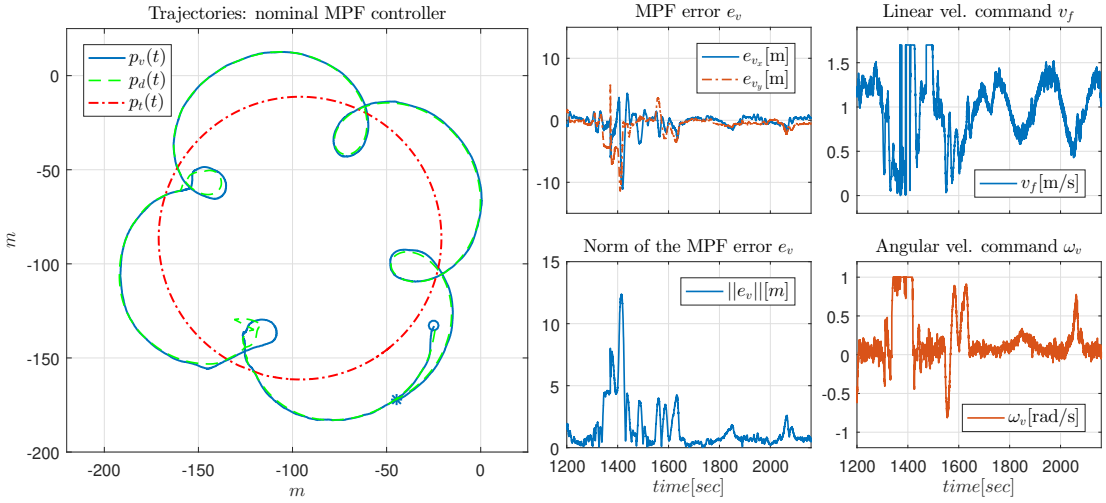
This section presents the experimental results obtained with the proposed strategies for Moving Path Following (MPF) control with a single vehicle following a moving target. We also present simulations comparing the performance of all three strategies: (i) nominal MPF control, (ii) robust MPF control with sliding mode term, (iii) robust MPF control with sliding mode term and disturbance observer. These results were published in [71]. Since only $N = 1$ vehicles is considered, the vehicle index is $i = 1$.

#### 3.3.2.1 MPF without Disturbance Rejection

First, we present the results obtained by using the nominal MPF controller with no robustness term ($\rho_1 = 0$) from [36, 37] that forms the base controller for performance evaluation of the proposed robust MPF schemes. The network router was located on the northeast part of Fig. 3.4a, close to the origin of the inertial frame. Data loss can be noticed in the trajectory the vehicle following the circular path $p^t_{d_1}(\gamma_1)$ on the distant southwest side on Fig. 3.4a, where the vehicle failed to receive the position of the target for a couple of moments due to the distance from the router. This can be noticed in the MPF error shown in Fig. 3.4b, around the time mark of $1400\,s$. Thereafter, the norm of the MPF error remains bounded within $3\,m$. The velocity control commands reach saturation during the transient and on occasional moments of communication losses (see Assumption 4), but remain within the actuation limits during most of the experiment.

#### 3.3.2.2 MPF with Sliding Mode Term

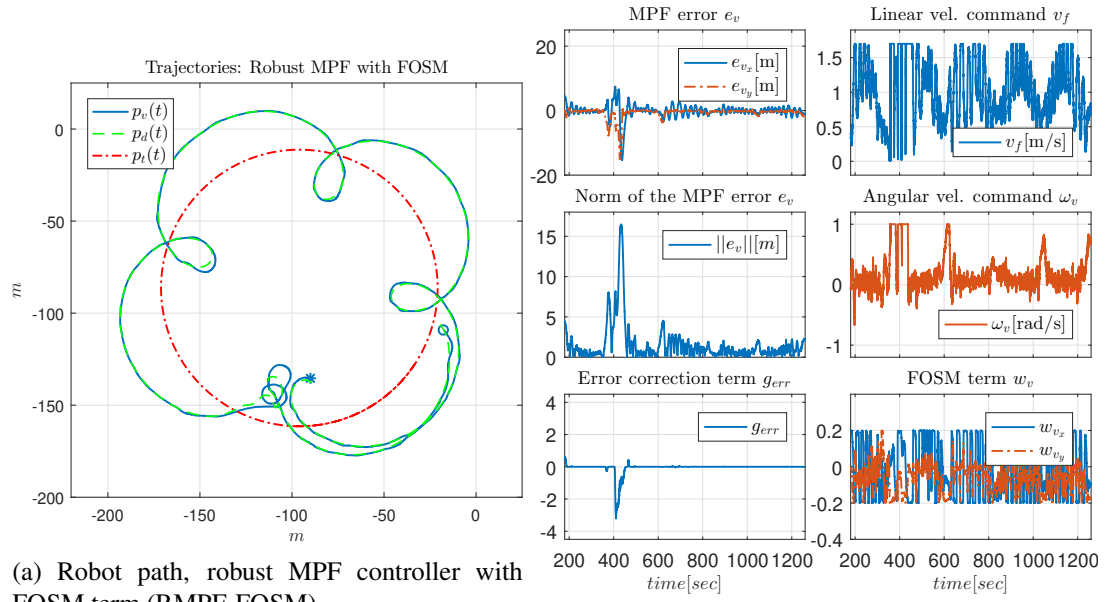
Figure 3.5a shows the vehicle trajectory obtained with the Robust MPF controller with FOSM (Theorem 3.2.1) with $\rho_1 = 0.2$ and $\varepsilon_w = 0.5$. Figure 3.5b illustrates the MPF error and the generated control signals. The norm of the MPF error remains bounded by less than $2\,m$, unless on




(a) Robot path, nominal MPF controller.

(b) Results for the nominal MPF controller.

Figure 3.4: Nominal MPF controller results.




(a) Robot path, robust MPF controller with FOSM term (RMPF-FOSM).

(b) Results for the robust MPF controller with FOSM term (RMPF-FOSM).

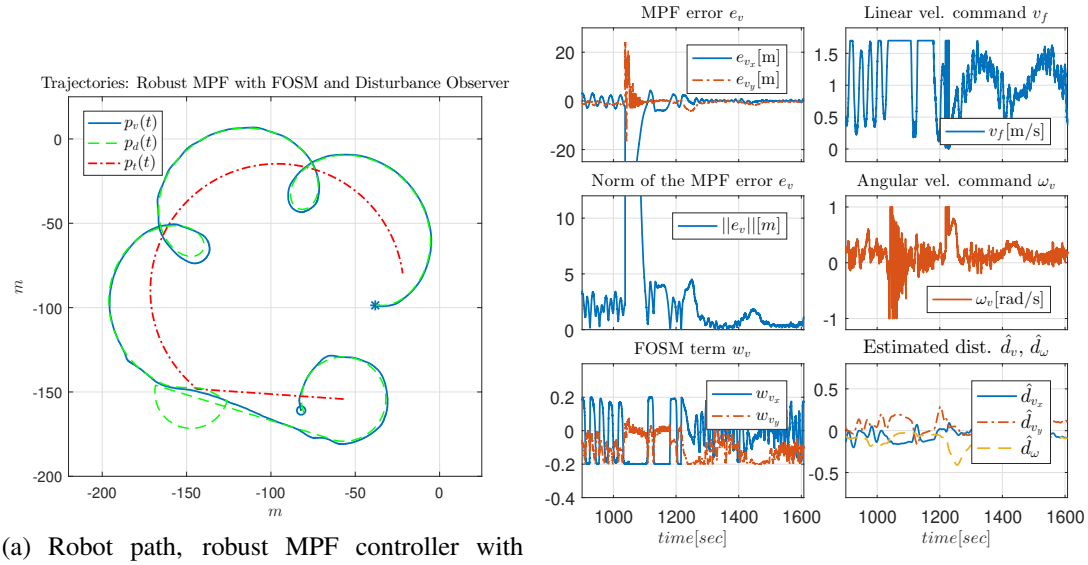
Figure 3.5: RMPF-FOSM controller results.

the farthest point from the router, where loss of data packages was observed. The MPF error $e_1$ converges to a neighborhood of the origin due to the implementation of the control law using (3.9) and (3.10). As expected, undesired chattering was observed in the linear and angular velocity commands. However, the control signals remain within their actuation limits, inducing a practical sliding mode behavior in the MPF error, as observed in Fig. 3.5b. The norm of the MPF error is smaller for the robust MPF with FOSM scheme. The dynamics during sliding ($e_1 = 0$) is *invariant* to disturbances bounded by $\rho_1$, provided that (3.11) is satisfied. Lastly, note the action of the error correction term $g_{e_1}$ during the moment around $400\,s$ where communications were lost. It controls

the progression of the virtual point along the moving path in order to enable faster convergence of the AUV. Once the MPF error starts to grow, $g_{e_1}$ slows down the evolution of $\gamma$, effectively waiting until the vehicle is able to follow the path.

#### 3.3.2.3 MPF with Sliding Mode Term and Disturbance Observer

Figures 3.6a and 3.6b show the results for the Robust MPF with FOSM and Disturbance Observer (Theorem 3.2.2) with $\rho_1 = 0.2$ and $\varepsilon_w = 0.5$. The gains for the disturbance observer were obtained using the pole placement method such that the desired observer response is critically damped with approx. $20\,s$ of rising time. It is evident from Fig. 3.6a that this experiment suffered significant




(a) Robot path, robust MPF controller with FOSM term and Disturbance Observer (RMPF-FOSM-DO).

(b) Results for the robust MPF controller with FOSM term and Disturbance Observer (RMPF-FOSM-DO).

Figure 3.6: RMPF-FOSM-DO controller results.

losses in the communications, with no data being received by the AUV during several minutes until the network was restored near the point $\{-150m, -150m\}$. In this case, since no data from the target was received for more than $10\,s$, the follower vehicle considers $p_t$ to be fixed, resulting in a stationary circular maneuver around the point $\{-150m, -50m\}$. After the communications were re-established around the time mark of $1250\,s$, the MPF error converges to the origin with a significantly smaller norm than in the nominal MPF (Fig. 3.4b) and in the robust MPF with FOSM without the disturbance observer (Fig. 3.5b), thereby demonstrating superior, robust performance to the two previous methods. There is an evident reduction in the chattering levels on the control signals when compared to the robust MPF with FOSM (Fig. 3.5b), which is a consequence of the use of the disturbance estimators along with the FOSM term, as mentioned in Remark 3.2.2. The improvement in the chattering levels can also be noticed from the FOSM term signal $w_1$ in Fig. 3.6b. The components of the estimated disturbances $\hat{d}_{v_1}$ and $\hat{d}_{\omega_1}$ are shown in Fig. 3.6b.

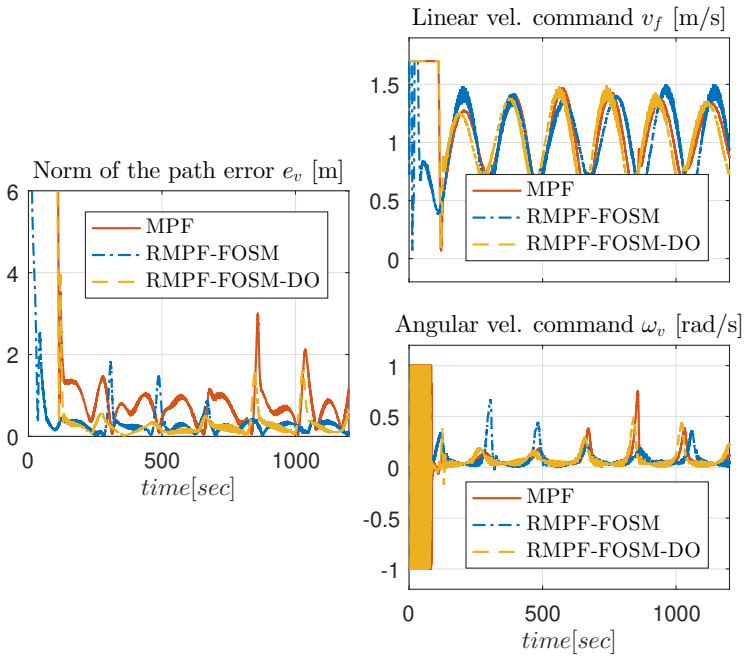



Figure 3.7: Comparison between the nominal MPF and the proposed robust MPF schemes.

#### 3.3.2.4 Performance Comparison

Here, we present simulation results in order to facilitate comparisons between the proposed robust MPF control laws and the nominal MPF control. These simulations were performed in the hardware-in-the-loop setup, where the simulation parameters were set identical to the experiments. Constant maritime currents of $0.1\,m/s$ to the North and East were simulated. DUNE has an in-built simulator for sensor and actuator dynamics, as well as the dynamic model of the AUV. The control commands generated by the MPF controllers are provided as velocity inputs to the detailed dynamic model. Figure 3.7 shows a comparison between the nominal MPF controller ($\rho = 0$) and the proposed robust MPF schemes. The value $\rho = 0.2$ is the same as the one used in the experiments and is enough to overcome the norm of the total disturbance due to the simulated maritime currents (see inequality (3.11)). Notice how the norm of the MPF error in the robust MPF schemes is noticeably smaller than in the nominal MPF controller, and how the chattering levels in simulation are much smaller than the values obtained in practice. This is expected because the chattering performance of sliding mode-based controllers tends to degrade when *unmodeled dynamics* are present in the control channel [72]. Furthermore, the amount of chattering in the control inputs in the Robust MPF with FOSM and Disturbance Observer (RMPF-FOSM-DO) is indeed smaller than in the Robust MPF with FOSM (RMPF-FOSM) scheme, the same result observed in the experiments.

### 3.3.3 Robust Cooperative Moving Path Following

This section presents the results obtained with the proposed robust strategies for Cooperative Moving Path Following (CMPF) control with three vehicles, namely: (i) CMPF without disturbance rejection, (ii) robust CMPF control with sliding mode term, (iii) robust CMPF control with sliding mode term and disturbance observer. These results were published in [70].

#### 3.3.3.1 Cooperative MPF without Disturbance Rejection

The first cooperative control experiment shows the results of the nominal CMPF controller, with target velocity compensation only and without any disturbance rejection, that is, $\rho_i = 0$ in (3.9), and $\hat{d}_{v_i} = 0$ and $\hat{d}_{\omega_i} = 0$ (disturbances are not compensated).

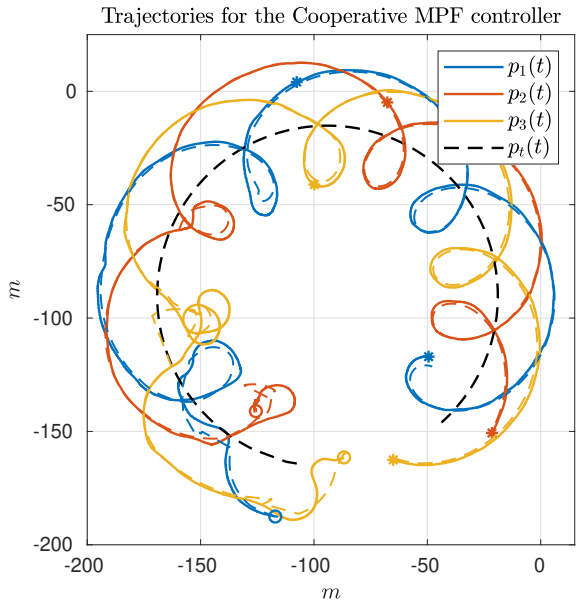



Figure 3.8: Robot paths, CMPF controller without disturbance rejection.

Figure 3.8 shows the trajectories of all three vehicles. The trajectory of the target is represented as the dashed black circle, in the clockwise direction. The small colored circles represent the beginning of the trajectory, while the colored asterisks represent its end. Noticeably, the three vehicles try to follow their respective paths (shown in dashed lines) around the rotating target, while maintaining their phase difference.

Figure 3.9 shows the obtained results. The initial position of the vehicles was distant from the network router (located closer to the northeast part of Fig. 3.8), which affected the wireless communications for a while. However, the initially large path variable errors rapidly decrease and remain bounded to less than $4m$ (Fig. 3.9(b)). Because of the communication losses and possibly the presence of ocean currents, the secondary controller described in Remark 3.3.1 had to recover some vehicles during the transient, resulting in some of the turning maneuvers we see in the beginning of the trajectories (Fig. 3.8). After that transient, the norm of the MPF errors converge to a small region of less than $3m$, while the control signal remains inside its linear region

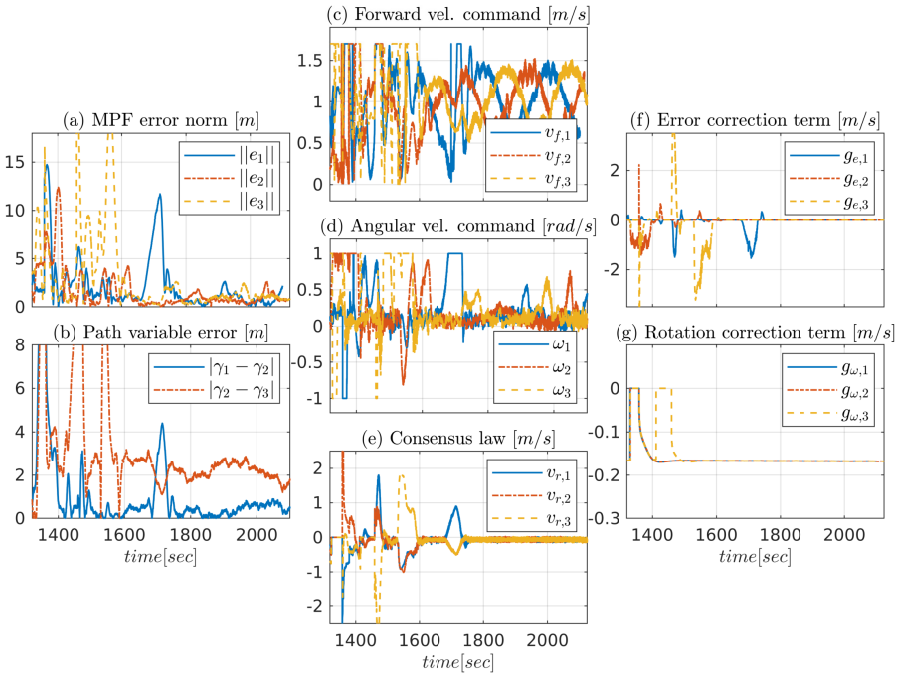



Figure 3.9: Experimental results for the CMPF controller without disturbance rejection.

(Figs 3.9(c) and 3.9(d)). As shown in Fig. 3.9(e), the consensus law for robot cooperation acts precisely when the path variable errors are high. Furthermore, the error correction terms acts when the MPF error norm is high (to prevent the evolution of the path variables) in Fig. 3.9(f), and how the rotation correction terms is bounded to a small value ($\approx 0.18\,m/s$) during the whole experiment Fig. 3.9(g). This is due to the fact that the target moves with constant angular velocity and the paths are circles to all three vehicles (3.22).

#### 3.3.3.2 Robust Cooperative MPF with Sliding Mode Term

The second CMPF experiment shows the results of the robust CMPF controller with velocity compensation and Sliding Mode term, with MPF control law given by (3.9) with $\rho_i = 0.2$ for the three vehicles and $\varepsilon_w = 0.5\,m$. The consensus law for cooperation among the vehicles is given by (3.23), as before.

Figure 3.10 shows the vehicle trajectories around the target, starting and ending in the southwest and southeast corners, respectively. Once more, due to communication losses and the presence of ocean currents in the southwest location, the secondary controller described in Remark 3.3.1 was activated for some of vehicles during the transient. However, the proposed controller was able to stabilize the error faster than the nominal controller. Besides, from Fig. 3.11(a), it is possible to notice the practical sliding mode phenomena around the origin $e_i \equiv 0$. That means that the controller is able to achieve better performance than the previous one, given that $\varepsilon_w$ can be designed to be arbitrarily small. However, from (3.10), small values of $\varepsilon_w$ can result in higher gains

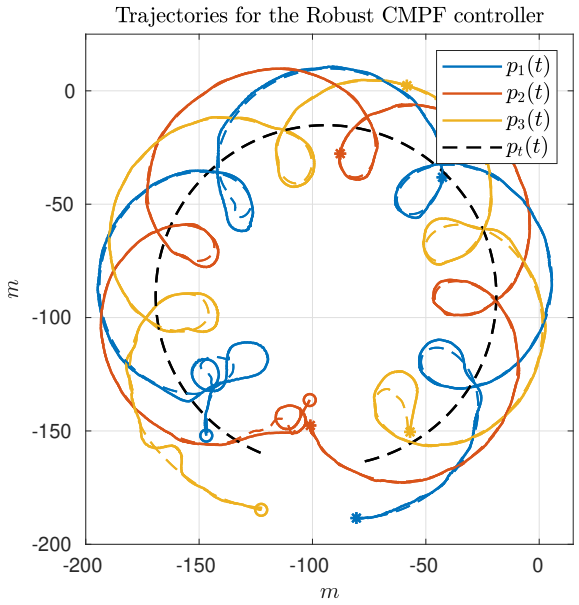



Figure 3.10: Robot paths, robust CMPF controller with Sliding Mode term (RCMPF-FOSM).

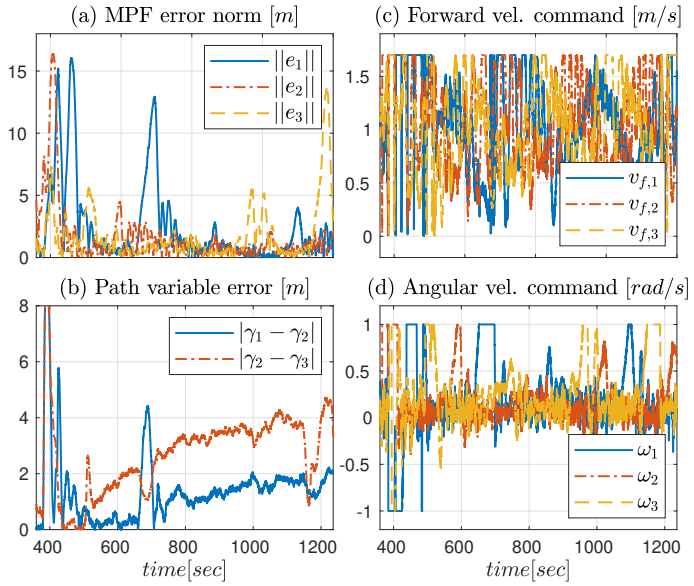



Figure 3.11: Results for the robust CMPF controller with Sliding Mode term (RCMPF-FOSM).

for $w_i$, which can potentially saturate the control inputs. In fact, sometimes the control saturation limits are reached after the transient, as shown in Figs 3.11(c) and 3.11(d), and practical sliding mode is momentarily lost. The reason is the limited velocity range allowed by the actuators, combined with our particular value choice for $\varepsilon_w$, and moments of occasional increase in the target velocity. Even so, performance is slightly better than in the previous case, and the amount of control chattering is acceptable. The consensus law, error correction signals and rotation correction signals are omitted, but are similar to those observed in Fig. 3.9.

#### 3.3.3.3 Robust Cooperative MPF with Sliding Mode Term and Disturbance Observer

The third and last experiment shows the results of the robust CMPF controller with velocity compensation, sliding mode FOSM term and direct disturbance compensation using the linear observers (3.14)-(3.15). The control law is given by (3.16) with $\rho_i = 0.2$ and $\varepsilon_w = 0.5\,m$, as before. Again, the consensus law for cooperation among the vehicles is given by (3.23).

Only the vehicles Noptilus 1 and 3 were used on this experiment, since the battery on Noptilus 2 was depleted. However, the results obtained by Noptilus 1 and 3 can still be compared to the previous results obtained for the same two vehicles. The chosen paths are the same circles defined in (3.27), but this time with $\phi_1 = 0$ and $\phi_3 = \pi\,rad$. This modification was used to guarantee that the two vehicles stay as far as possible from each another. Once again, notice the practical sliding

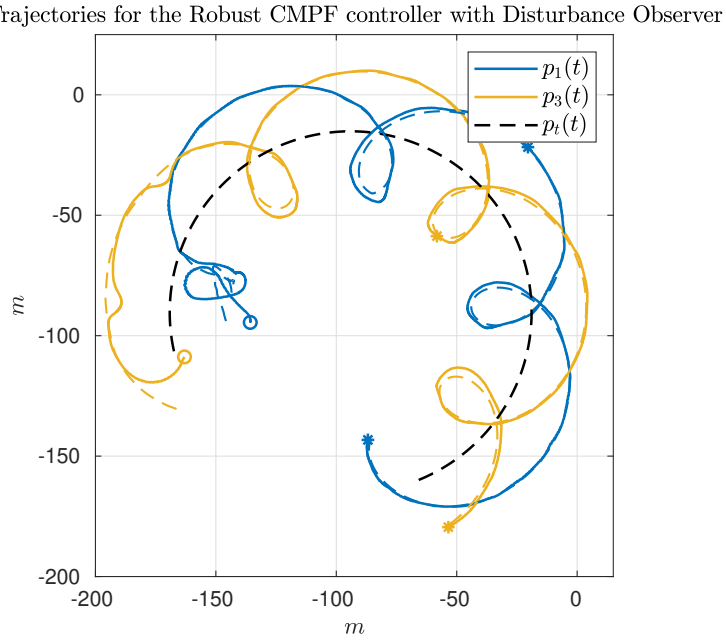



Figure 3.12: Robot paths, robust CMPF controller with with FOSM and disturbance observer.

mode phenomena around the origin $e_i \equiv 0$, except during the instants where the control inputs are saturated. However, in this case, the control chattering is significantly smaller than the one observed in Figs 3.11(c), 3.11(d), under the same experimental conditions. We explain this fact by the presence of the disturbance estimator. Since part of the disturbance is compensated, the sliding mode term can spend less effort compensating the remaining total disturbance, a result in agreement with the theoretical discussion by the end of Section 3.2.2. The estimated disturbances are shown in Fig. 3.14. The linear velocity disturbances remained bounded by less than $0.3 m s$ after the transient, while the angular velocity disturbances showed higher variation, but remained bounded to less than $0.5 rad\,s$ after the transient.

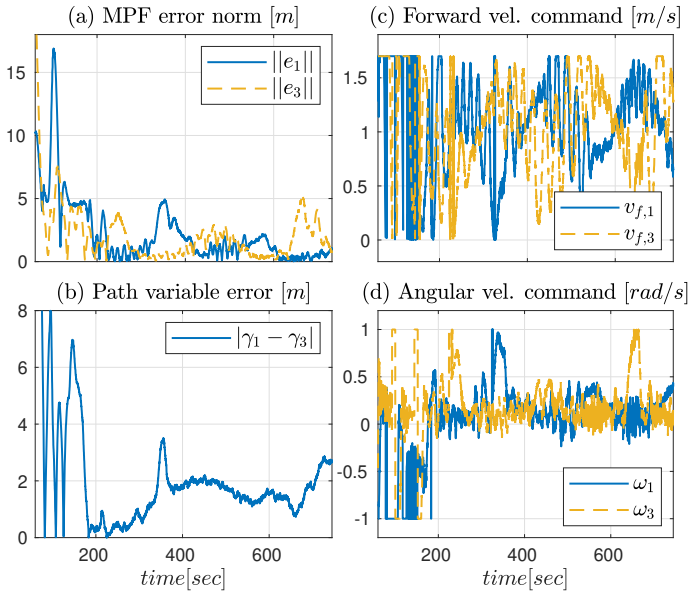



Figure 3.13: Results for the robust CMPF controller with FOSM and disturbance observer.

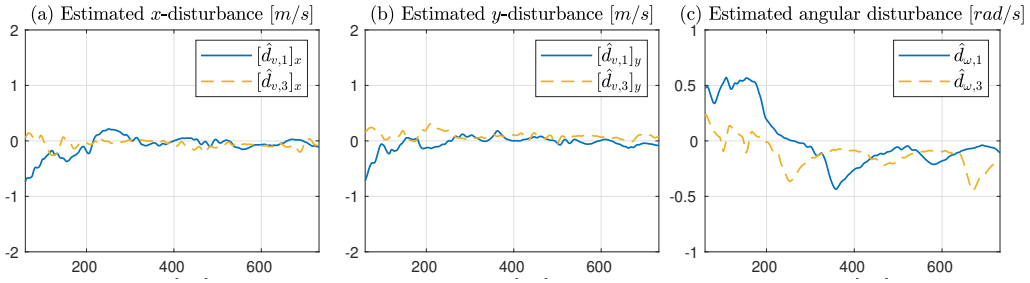



Figure 3.14: Results obtained with the disturbance estimator.

## 3.4 Concluding Remarks

This chapter has addressed the Robust MPF and CMPF problem for underactuated vehicles, such as AUVs. Regarding the robust MPF for a single vehicle, two robust MPF control schemes were proposed, ensuring asymptotic stability of the origin of the MPF error in the presence of bounded disturbances acting on the vehicle and inaccurate estimates for the target velocities. The proposed control schemes were validated experimentally using an AUV and it was demonstrated that their performance is superior to the nominal MPF control technique. Additionally, it was observed that the use of disturbance observers for active compensation of the bounded kinematic disturbances can reduce the level of chattering on the control signals by reducing the total amount of disturbance that the FOSM term must compensate. Satisfactory performance was achieved in the presence of unmodeled dynamics of the inner-loop controller for the actuators and unknown kinematic disturbances acting on the vehicle.

Regarding the task of cooperative MPF with a team of marine vehicles, we have demonstrated that the origin of all MPF errors are stable with the two proposed robust CMPF control schemes in the presence of bounded disturbances acting on the vehicles. Furthermore, it was theoretically

demonstrated that the cooperative control scheme is ISS with respect to the path variable estimation errors and to two other bounded, auxiliary input variables, named error correction term and rotation correction terms. The proposed robust controllers (3.9) and (3.16) guarantee that the MPF error is globally uniformly bounded to a small neighborhood of the origin while maintaining acceptable control chattering. The narrow linear region of the actuators imposes limits on how small $\varepsilon_w$ can be designed in practice. Lastly, we conclude that control law (3.16) actually improved the control chattering in practice, corroborating the theoretical insight of Remark 3.2.2.

# Chapter 4

# Safety-Critical Robotic Navigation Systems

In this chapter, we present three applications of path following control for multi-robot navigation with collision avoidance guarantees using the framework of CLFs and CBFs.

1. Section 4.1 presents an application of safe path following for the control of autonomous marine vessels. The results were published in [67].

2. Section 4.2 presents an application of safe path following using spline-based paths for autonomous vehicles. The results were published in [68].

3. Section 4.3 presents an application of safe Cooperative Path Following for robotic vehicles with deadlock avoidance. The results were published in [66].

## 4.1 Safety Filter for Autonomous Marine Vessels

In this section, we present a safety-critical control architecture for autonomously maneuvering a vessel from an initial position to a goal position while estimating the position of static nearby obstacles and evading them. The development and implementation of the system architecture was a team project fully developed by masters and PhD students associated with the Cyber-Physical Control Systems and Robotics Laboratory (C2SR), for the *Njord - The Autonomous Ship Challenge*, a pilot student-driven competition organized by the Norwegian University of Science and Technology (NTNU) [1] in 2021. Due to the worldwide COVID-19 pandemic, the 2021 edition of the competition was fully digital, and has proposed several challenges ranging from testing the vessel's situational awareness capabilities in trafficked water to testing the vessel's autonomous maneuvering capabilities.

The software architecture was developed within the Robot Operating System (ROS) framework, and comprises three main software modules: Situational Awareness (SA), Sensor Fusion

[1] https://www.njordchallenge.com/

and Mapping (SM), and Planning and Control (PC) modules. These modules are composed of other sub-modules, which are introduced as follows (see Fig. 4.1).

**[SA] Object detection**: The object detection module is responsible for detecting objects of interest (e.g., other vessels) in the surroundings of the ship. Using vision as input, the detection task is described by two sub-tasks: object *localization* and object *classification*.

**[SA] Object tracking**: The object tracking module is responsible for keeping track of objects previously detected. The need for tracking arises from two reasons: 1) reducing the module computational cost by reducing the search space in the object detection module, and 2) allowing accurate distance between the vessel and the detected object.

**[SA] Object state estimation**: This module is responsible for recovering the distance between the vehicle and objects of interest, e.g., other vessels.

**[SM] Sensor fusion and Mapping:** Sensor fusion is responsible for estimating pose, velocity, and vehicle acceleration using noisy measurements provided by the simulator (position and velocity). Mapping manages the obstacles detected by the situational awareness.

**[PC] Motion planning**: The motion planning module is responsible for generating the collision-free trajectories (position, velocity, and acceleration) that the vessel must follow to achieve a given spatial goal, given the positions and velocities of nearby obstacles.

**[PC] Path-following control**: This module consists of an kinematic, outer loop path following controller for the vehicle's planar position and orientation. It is responsible for steering the vehicle along a specified geometric path (provided by the motion planning module).

**[PC] Safety control**: This module implements a minimally invasive controller that modifies the command velocities generated by the path-following control module in case a potentially unsafe configuration for the vessel is imminent, such as collisions with neighboring obstacles. Therefore, it ensures that the velocity commands will not result in potentially dangerous behavior during the tasks.

**[PC] Dynamic Control**: The purpose of this module is controlling the vessel at the dynamic level by sending force/torque commands to the vessel actuators. The designed inner loop controller must be able to track velocity commands sent by the path following controller.

A complete description for the modules can be found in the published work [67]. In this section, we focus on the design description of two submodules of the Planning and Control (PC) module: (i) the outer loop for vehicle position control using path following and (ii) the CBF-based safety-filter controller to avoid collisions against other vehicles. To illustrate the efficacy of the proposed methods, simulations results using the Gemini-Unity simulator [2] are also presented and discussed by the end of the section.

### 4.1.1 Path Following Controller

In this section, we describe the path following (PF) control algorithm implemented for steering a vessel towards a planar path $p_d : \mathbb{R} \to \mathbb{R}^2$ described by a B-spline and generated by the motion

[2] https://njord.gitbook.io/njord/installation-guide/unity

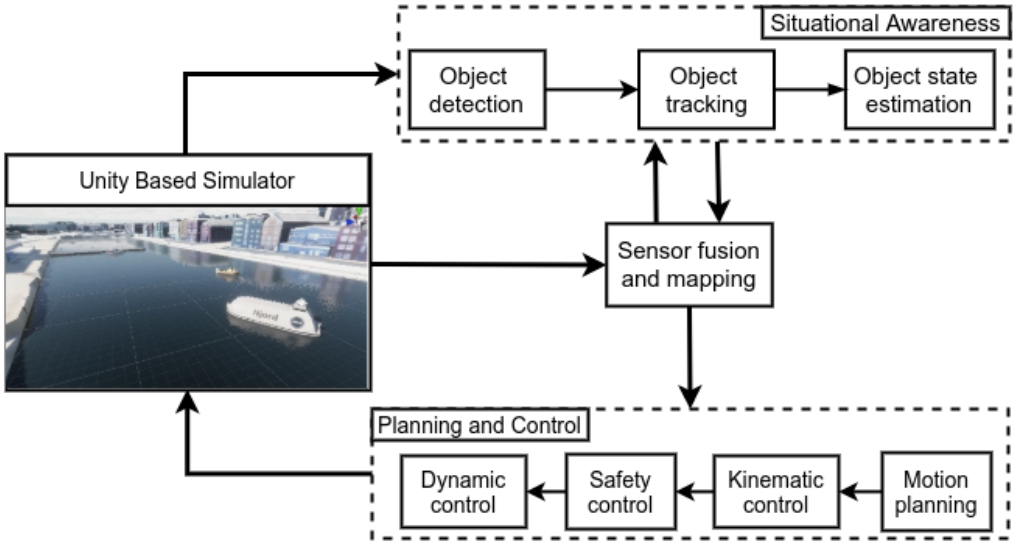



Figure 4.1: Software system architecture.

planner module [67]. Since B-splines are class $C^1$ functions, Assumption 1 is satisfied (the path is a smooth function). Then, the PF controller (2.28) described in Section 2.3.3 computes the required velocities for the vessel to converge to the path $p_d(\gamma)$, effectively controlling the vehicle at the kinematic level.

Let the state $x$ of the vessel be defined by its inertial 2D pose $x = (p, \phi) \in \mathbb{R}^2 \times [-\pi, \pi]$ of the vessel's stern with respect to an inertial 2D reference frame $\mathscr{I}$, with $p \in \mathsf{R}^2$ being the planar center position of the stern and $\phi \in [-\pi, \pi]$ being the vehicle's planar angle. We assume the vehicle kinematics to be described by the unicycle model in (2.21). With the hypothesis of *kinematic control*, the control signal $u = [\, v_f \quad \omega \,]^\mathsf{T}$ is given by the longitudinal linear velocity $v_f$ and angular velocity $\omega = \dot{\phi}$ of the vessel as in (2.22), which are assumed to be accurately controlled by the MRAC inner controller loop. The controlled position of the vessel is located at the center point of the vehicle, slightly in front of the vehicle's stern, that is, a point $\bar{p} = p + R(\phi)\varepsilon$ placed at a constant position $\varepsilon = [\, \varepsilon_x \quad 0 \,]^\mathsf{T}$ from the origin of its body frame $\mathscr{B}$, as defined in Section 2.28. Then, the derivative of $\bar{p}$ with respect to time is given precisely by (2.24), and the path following controller is given by (2.28). The path variable dynamics is defined as

$$\dot{\gamma} = v_d - k_e \tanh \eta_e \,, \qquad k_\eta > 0 \tag{4.1}$$

$$\eta_e = \frac{\partial}{\partial \gamma} \left( \frac{1}{2} \|e\|^2 \right) = -e^\mathsf{T} R(\phi)^\mathsf{T} \nabla p_d(\gamma) \tag{4.2}$$

where the second term in the right-hand side of (4.1) with (4.2) is akin to the error correction term in (3.20), having the effect of delaying the evolution of the desired point along the path by explicitly avoiding the evolution of $\gamma$ if the path following error is too large [71]. Since the path is fixed in this case, there's no need for using a path rotation correction term as in (3.22). In (4.1), $v_d$ represents a desired speed assignment for the vessel.

### 4.1.2 Safety Filter Design

Although the motion planning module is able to compute reliable paths in most situations, many sources of inaccuracies could increase the risk of collisions: unmodeled vessel dynamics, imprecise information about the real dimensions of the vessel or obstacles, etc. These factors could lead to the computation of potentially almost unsafe paths, with minimum distance to a nearby obstacle being dangerously small. In order to address this issue, a safety-critical controller was designed with the objective of adding an extra level of safety to the navigation task, ensuring that the vehicle does not collide with nearby obstacles even if: (i) the motion planning module computes an unsafe path, (ii) the PF controller convergence is hampered.

Consider a marine vessel moving the surface of the water, with state $x = (p_\varepsilon, \phi) \in \mathbb{R}^2 \times [-\pi, \pi]$, where $p_\varepsilon \in \mathbb{R}^2$ is an offset position to the vessel's center of mass as given by (2.23). Consider a set of $N$ non-intersecting obstacles with set of indexes $\mathbb{O} = \{1, \cdots, N\}$. For the $i$-th obstacle detected by the obstacle detection algorithm, consider the *safe* set $\mathscr{C}_i = \{p_\varepsilon \in \mathbb{R}^2 : h_i(p_\varepsilon) \geq 0\}$ defined as the superlevel set of the following quadratic CBF:

$$h_i(p_\varepsilon) = \frac{1}{2}(p_\varepsilon - p_i)^\mathsf{T} H_i (p_\varepsilon - p_i) - \frac{1}{2} \tag{4.3}$$

$$\nabla h_i(p_\varepsilon) = H_i(p_\varepsilon - p_i) \tag{4.4}$$

where $p_i \in \mathbb{R}^2$ is the center position of the $i$-th obstacle and $H_i = H_i^\mathsf{T} \in \mathbb{S}^2_{\geq 0}$ is a constant Hessian matrix defining the elliptical shape of the $i$-th barrier boundary surrounding the $i$-th obstacle, as illustrated in Fig. 4.2.

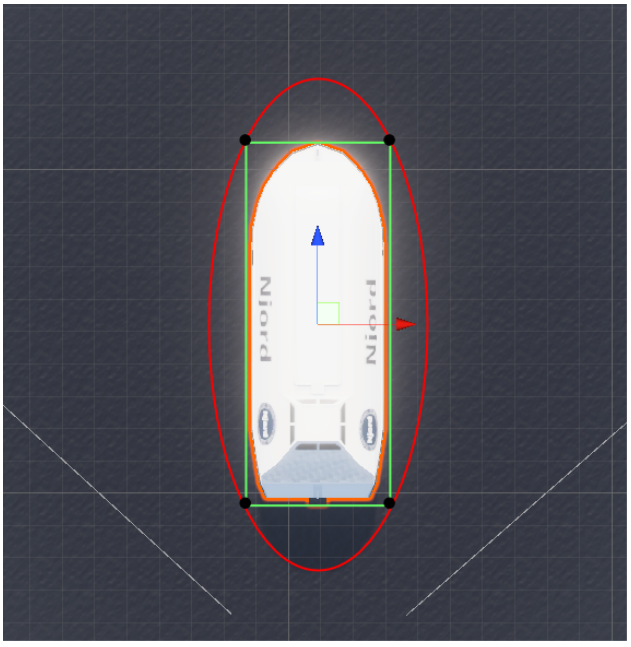


Figure 4.2: A vessel-like obstacle: an elliptical unsafe region is fitted by four corner points, with fixed minor-major axis ratio (eccentricity).

**Remark.** In (4.3), $H_i$ is computed by fitting the ellipse $h_i(p_\varepsilon) = 0$ (with fixed eccentricity) to the four corner points defining the bounding box corresponding to the $i$-th obstacle (see Fig. 4.2), according to Fig. 4.2. Since $p \in \mathbb{R}^2$ is the center position of the vessel's stern, the origin of the frame marked in the center of the vessel at Fig. 4.2 is precisely at the offset position $p_\varepsilon \in \mathbb{R}^2$.

A controller based on the *safety filter* of (2.36) is proposed as follows. Considering the $N$ obstacles modeled by the CBFs (4.3) ($i \in \mathbb{O}$), we propose a modification of the safety-filter QP in (2.36) with multiple CBFs:

$$u^\star(x) = \underset{u\in\mathbb{R}^2}{\operatorname{argmin}} \ \|u - u_{\text{pf}}(x)\|^2 \tag{4.5}$$

$$s.t. \quad \nabla h_i(p_\varepsilon)^\top \underbrace{R(\phi)\Delta_\varepsilon u}_{\dot{p}_\varepsilon} + \beta h_i(p_\varepsilon) \geq 0\,, \quad i \in \mathbb{O} \tag{CBFs}$$

where the extended class $\mathscr{K}$ function was chosen as the linear function $\beta(h_i) = \beta h_i$, $\beta > 0$. In the CBF constraints of (4.5), the time derivative of $p_\varepsilon$ defined in (2.24) as used since $\dot{h}_i = \nabla h_i^\top \dot{p}_\varepsilon$. Controller (4.5) effectively modifies the nominal velocity command $u_{\text{pf}}(x)$ computed by the PF controller (2.28), computing the closest velocity command $u^\star(x)$ to $u_{\text{pf}}(x)$ such that the safety constraints are not violated. In particular, if all the constraints are inactive, the QP solution is given by $u^\star(x) = u_{\text{pf}}(x)$, which means that the velocity commands sent by the PF controller simply bypasses the safety-critical module. The safe set $\mathscr{C} = \bigcap_{i=1}^{N} \mathscr{C}_i$ is rendered forward invariant by controller (4.5), and since the rectangle defined by the $i$-th obstacle is completely contained in the elliptical region defined by the unsafe set $\overline{\mathscr{C}}_i$, the vehicle's position $\bar{p}$ never enters the obstacle rectangular area, even in case the path planner produces a B-spline path that is too close to the actual obstacle.

### 4.1.3 Simulation Results

The modules were designed using the Robot Operating System (ROS) middleware suite, a set of open source software libraries for building robot applications. The ROS-based design philosophy consists of designing separate ROS nodes for performing specific tasks, and then managing the exchange of information between nodes by means of a publishing/subscribing system of *topics*.

Figure 4.3 shows the overall structure of our ROS package: the node **adapter_server** (provided by Gemini) is responsible for publishing the vessel latitude/longitude simulated coordinates from Gemini-Unity, vessel velocities and raw images from the onboard RGB camera. The **sensor_fusion** node publishes the transformed pose and velocities of the vessel. The **perception** node receives the raw RGB camera image and vessel pose, and publishes to a topic describing the corner points of detected rectangular objects. Figure 4.4 shows an example of detection of an obstacle (vessel) and extraction of its features, which will latter be used to generate the spline paths and to create the safety barriers. The **outer_loop** node is responsible for: (i) generating a new B-spline path every time a new obstacle message or goal position is received, (ii) implementing the PF controller for steering the vessel towards the path, as described in Section 4.1.1. The **safety_control**

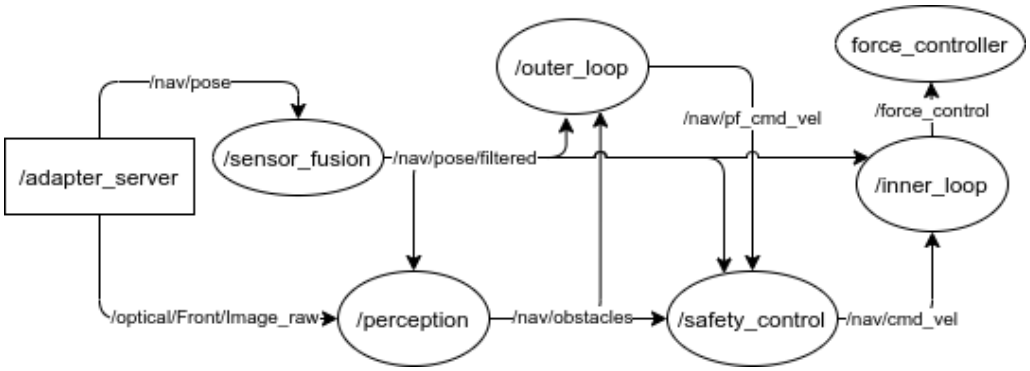



Figure 4.3: ROS architecture graph: structure of interconnections between ROS nodes. Nodes and topics are represented by ellipses and arrows, respectively. Nodes with outward arrows are publishing to that corresponding topic, while nodes with inward arrows are subscribed to that topic and receive information from it.

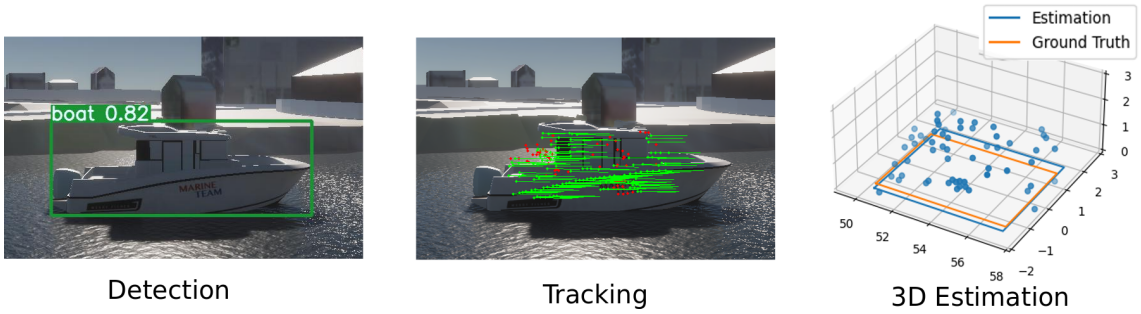



Figure 4.4: Results of the situational-awareness module: the boat is detected using the YOLO algorithm [81], which searches for objects in the scene that are likely to correspond to a boat (left image). Features are extracted from the detected region and tracked across consecutive frames (middle image). After tracking two or more frames, the depth of the 3D points is recovered and footprint of the boat is estimated (right image).

node receives the vessel pose and information about detected obstacles, as well as the command velocity previously computed by the safe path following controller (4.5), and using the CBF-based controller described in Section 4.1.2, computes a safer velocity command for the inner loop. The **inner_loop** node is responsible for tracking the command velocity sent from the PF controller (4.5), using the MRAC strategy [67]. Finally, it publishes the forces and torques needed to control the vessel at the dynamic level, sending it to the **force_controller** node (provided by Gemini) that interacts with the vessel model in Unity.

The following experimental setup was designed: (i) vehicle placed at the same initial position in all experiments, (ii) other vehicles in the scene acting as obstacles, and (iii) single and multiple goal waypoints placed in the same position in all experiments. All Unity models used in the simulations were provided by the competition. Figure 4.5 shows the navigation result for the single obstacle, multiple waypoints scenario. The vehicle leaves the marker “0" towards the goal position at marker “1". Initially, the generated spline path is a straight line from “0" to “1", which the vehicle starts following. After 500$s$ of simulation, the perception module detects the obstacle

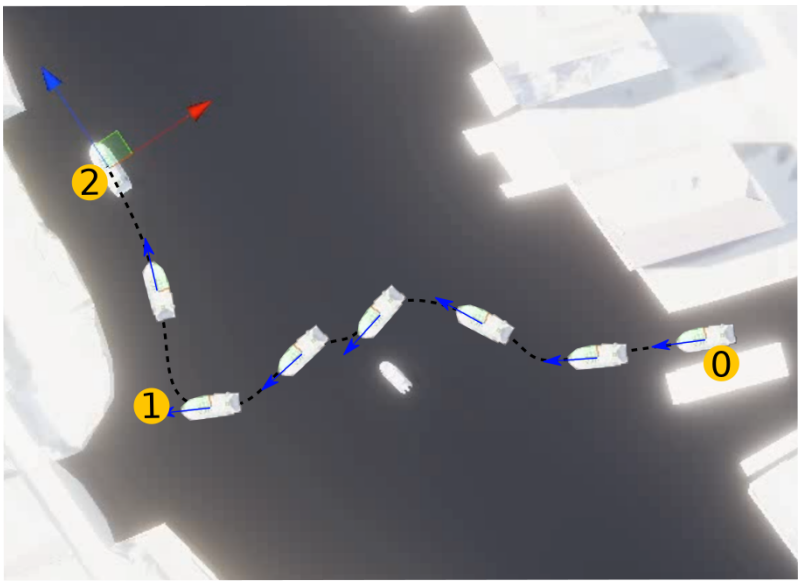



Figure 4.5: Single obstacle, multiple waypoint scenario.

vehicle, sending an obstacle message to the **outer_loop** and **safety_control** nodes. A new spline is generated by the motion planner module, allowing the vehicle to reach its destination safely (after approximately 1500$s$ of simulation) by performing a successful avoidance maneuver around the obstacle (see Fig. 4.5). After the vehicle reaches marker "1", a new goal is sent (marker "2") and a new spline path is generated, allowing the vehicle to reach its final goal. Figure 4.6 illustrates

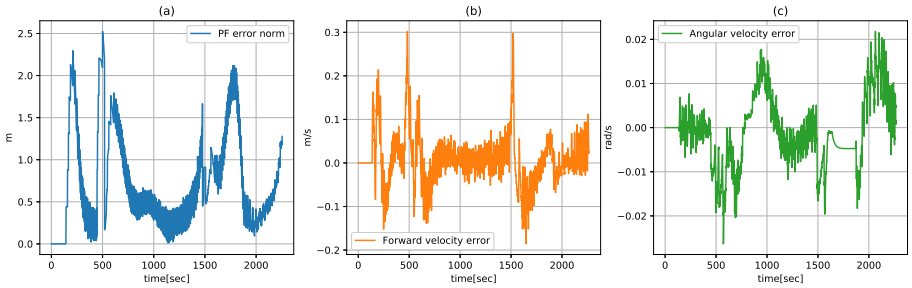



Figure 4.6: Control results with a single obstacle, multiple waypoint scenario.

the performance of the control modules at the kinematic and dynamic level. Figure Fig. 4.6(a) shows the PF error norm $\|e\|$ as defined in (2.26). Notice the path error transients at 500 and 1500$s$ timestamps, corresponding to discontinuities in the path position and gradient due to the resetting of the path variable $\gamma$, precisely when the motion planner computes a new path. Figures 4.6(b)-(c) illustrate the performance of the inner loop MRAC at tracking the vehicle velocities, with peaks at the same timestamps due to saturation of the vessel actuators at the transients (see Fig. 4.6(b)). Figure 4.7 shows the navigation results for a single goal waypoint (marker "1") and several scattered obstacle vehicles. Figures 4.7(b)-(e) show the corner points of the detected obstacle and the generated spline after detection at several timestamps. Notice how the perception module was able to successfully detect the most critical obstacle in the scene and reach its final

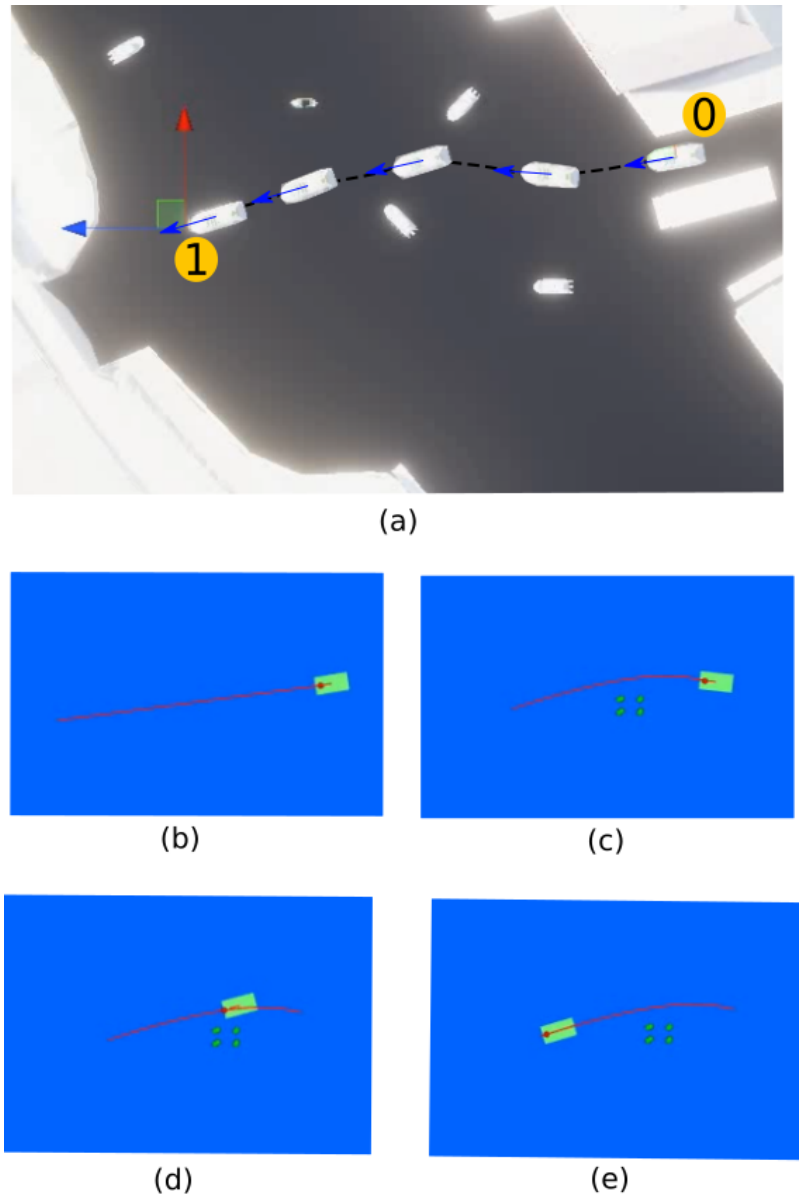



Figure 4.7: Single waypoint, multiple obstacles scenario: (a) shows the snapshots of the vessel in the Gemini-Unity simulator. The forward direction is shown by a blue arrow. (b)-(e) illustrate the B-spline trajectory at different timestamps.

destination safely.

## 4.2 CLF-CBF-based Safe Path Following Control

In this section, we present a path following control framework for autonomous vehicles using a CLF and multiple CBFs, based on the QP (2.41). The proposed controller assumes a centralized communication system that provides to the agents (vehicles) the knowledge of their relative positions and the geometrical data of the lane (assumed to be available from digital maps), in order

to determine a set of B-spline-based paths and barrier constraints, effectively avoiding collisions between pairs of vehicles and between each vehicle and the lane limits. The main contributions are twofold:

(i) An expansible, safety-guaranteed autonomous navigation control system for multiple vehicles, based on the framework of Path Following control and Control Barrier functions for collision avoidance;

(ii) A software package in Python containing tools for the simulation of autonomous vehicles operating in a multi-lane environment, including the creation and editing of B-spline curves. The project is publicly available at `https://github.com/CaipirUltron/spline_cbfs`.

Simulation results are presented and discussed, aiming to demonstrate the viability and effectiveness of the proposed framework as a possible solution for the mobility problem concerning the design of future smart cities. The work presented in this section was published in [68].

### 4.2.1 Problem Formulation

Consider $N$ robotic vehicles with positions $p_i = [\, x_i \;\; y_i \,]^\top \in \mathbb{R}^2$ and planar angle $\phi_i \in \mathbb{R}$ in an inertial frame $\mathscr{I}$, as illustrated by Fig. 4.8. The dynamic equations for the $i$-th vehicle are given by the unicycle model of (2.21), with kinematic control input $u_i$ given by the vehicle's controllable linear and angular velocities, as in (2.22). The $i$-th vehicle state is defined as its planar pose $\xi_i = [\, x_i \; y_i \; \phi_i \,]^\top$. Point $p_{\varepsilon_i} = [\, x_{\varepsilon_i} \;\; y_{\varepsilon_i} \,]^\top = p_i + R(\phi_i)\varepsilon$ is defined as in (2.23) with $\varepsilon = [\, \varepsilon_x \;\; 0 \,]^\top$,

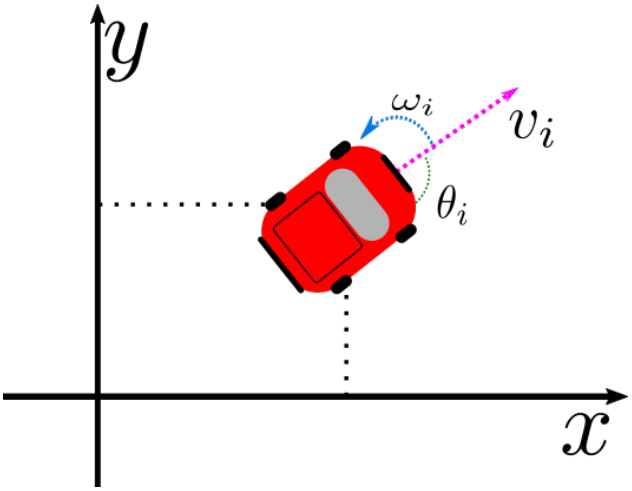



Figure 4.8: Vehicle kinematic model.

that is, a point at a distance $\varepsilon_x$ to the front of the vehicle, as before. The time derivative of $p_{\varepsilon_i}$ is given by $\dot{p}_{\varepsilon_i} = R(\phi_i)\Delta_\varepsilon u_i$, as in (2.24). In this section, the path error is defined in the inertial frame $\mathscr{I}$ as $e_i = p_{\varepsilon_i} - p_{d_i}(\gamma_i)$, where $p_{d_i} : \mathbb{R} \to \mathbb{R}^2$ is the B-spline the $i$-th vehicle is tasked to follow.

**Definition 4.2.1** (Safe Navigation)**.** Assume the geometry of the $i$-th vehicle is modeled by an elliptical region $\mathscr{V}_i \subset \mathbb{R}^2$, and define an equivalent parameterized elliptical curve $E_i : \mathbb{R} \to \mathbb{R}^2$ such

that $\partial\mathscr{V}_i = \{E_i(\rho), \rho \in \mathbb{R}\}$, that is, $E_i(\rho)$ covers the boundary of the elliptical region. Then, define $\mathscr{C} = \cap_{k=1}^{M} \mathscr{C}_i$ as the intersection of $M$ arbitrary regions $\mathscr{C}_i \in \mathbb{R}^2$. The autonomous navigation task is said to be *safe* for the $i$-th vehicle if $\mathscr{V}_i \subset \mathscr{C}$ for all time.

The previous definition means that the elliptical region defining the vehicle's geometry is required to remain inside a certain subset of $\mathbb{R}^2$, avoiding collisions against forbidden regions modeled by the unsafe set $\overline{\mathscr{C}}$ (e.g. neighboring vehicles and environmental limits). As seen before, we can use CBF-based controllers to make the system trajectory forward invariant with respect to $\mathscr{C}$.

### 4.2.2 CLF-CBF-based Controller for Safe Path Following

While CLFs can be used to implement path following controllers, CBFs define a very natural framework for avoiding collisions among neighboring vehicles and between vehicles and more general obstacles in the environment. Next, we propose an optimization-based controller for safe path following with multiple vehicles, using CLFs and CBFs.

#### 4.2.2.1 CLF-based Path Following Control

Considering the $i$-th vehicle, define the CLF candidate $V(\xi_i) = \frac{1}{2}\|e_i\|^2$. Its time derivative is

$$\dot{V}(\xi_i, u_i, \gamma_i, \dot{\gamma}_i) = e_i^{\mathsf{T}} R(\phi_i) \Delta_\varepsilon u_i - e_i^{\mathsf{T}} \nabla p_{d_i}(\gamma_i)\, \dot{\gamma}_i \tag{4.6}$$

Since the controller must continuously steer the robot towards the path, the dynamics of the path variable is designed as $\dot{\gamma}_i = \gamma_{d_i} + g(e_i)$, where $\gamma_{d_i}$ is a given path speed and $g(e_i)$ is an error correction term for the $i$-th vehicle. Then, considering a known path variable dynamics, the set of controllers that stabilize the origin of the path error $e_i = 0$ is given by

$$\mathbb{K}_{\text{clf}}(\bar{\xi}_i) = \{u_i \in \mathbb{R}^2 \mid \dot{V}(\xi_i, u_i, \gamma_i, \dot{\gamma}_i) + \alpha V(e_i) \leq 0\} \tag{4.7}$$

where here the class $\mathscr{K}$ function for the CLF was simply chosen as the linear function $\alpha(\tau) = \alpha\tau$.

#### 4.2.2.2 Vehicle Barriers

Here, we propose barrier function candidates to avoid collisions between pairs of vehicles. Define the quadratic form $h_i : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$, which is associated to the $i$-th vehicle:

$$h_i(\delta_i, \phi_i) = \frac{1}{2}\delta_i^{\mathsf{T}} H_i(\phi_i)\delta_i - \frac{1}{2}, \qquad \delta_i = \delta - p_i \tag{4.8}$$

with $H_i(\phi_i) = R(\phi_i)\Lambda_i R(\phi_i)^{\mathsf{T}}$, $\Lambda_i^{-1} = \text{diag}\{a_i^2, b_i^2\} \in \mathbb{R}^2$ is a diagonal matrix with constant positive entries, $a_i \neq b_i$. Equation (4.8) defines a convex quadratic polynomial in $\delta \in \mathbb{R}^2$, whose level sets are ellipses centered on the $i$-th vehicle center $p_i \in \mathbb{R}^2$ and oriented by the $i$-th vehicle angle $\phi_i$, with their shapes defined by the eigenvalues $a_i^2, b_i^2$ of $\Lambda_i^{-1}$. Therefore, $h_i$ is dependent on the

complete vehicle pose $\xi_i$. The partial derivatives of $h_i$ with respect to $\delta_i$ and $\phi_i$ are given by

$$\frac{\partial h_i}{\partial \delta_i}(\delta_i, \phi_i) = H_i(\phi_i)\delta_i \tag{4.9}$$

$$\frac{\partial h_i}{\partial \phi_i}(\delta_i, \phi_i) = \frac{1}{2}\left(\frac{1}{a_i^2} - \frac{1}{b_i^2}\right)\delta_i^\top R(\phi_i)\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}R(\phi_i)^\top\delta_i \tag{4.10}$$

**Remark.** If (4.10) vanishes, the vehicles loses steering controllability, meaning that it becomes impossible to actuate its angular velocity. Since (4.10) vanishes when $a_i = b_i$, we must choose ellipses with $a_i \neq b_i$ (they cannot be circles) for the vehicle to be steerable.

Define the following control barrier candidate for the vehicle pair $\{i, j\}$ as

$$h_{ij}(\xi_i, \xi_j) = \min_{\rho\in\mathbb{R}} h_i(E_j(\rho)) \tag{4.11}$$

where $E_j(\rho)$ defines the parametrized ellipse described by the equation $h_j(\delta) = 0$, that is, the elliptical boundary of the superlevel set corresponding to the geometry of the $j$-th neighbor vehicle, and given by the following parametric equation:

$$E(\rho) = \begin{bmatrix} a_j\cos(\rho)\cos(\phi_j) - b_j\sin(\rho)\sin(\phi_j) + x_j \\ a_j\cos(\rho)\sin(\phi_j) + b_j\sin(\rho)\cos(\phi_j) + y_j \end{bmatrix} \tag{4.12}$$

where $\xi_j = [\, x_j \;\; y_j \;\; \phi_j \,]^\top$ is the state of the $j$-th vehicle. Besides, from (4.8), notice that $h_j(E_j(\rho)) = 0\ \forall \rho \in \mathbb{R}$, showing that (4.12) in fact describes a general ellipse corresponding to $\partial\mathscr{V}_j$.

**Remark.** Problem (4.11) consists of minimizing the value of the quadratic form (4.8) over the elliptical boundary $\partial\mathscr{V}_j$. Notice that $h_{ij}(\xi_i, \xi_j) < 0$ if and only if $(E_j(\rho^\star) - p_i)^\top H_i(\phi_i)(E_j(\rho^\star) - p_i) < 1$ at the optimal solution $\rho^\star = \operatorname{argmin}_{\rho\in\mathbb{R}} h_i(E_j(\rho))$, which means that $\mathscr{V}_i \cap \mathscr{V}_j \neq \emptyset$, meaning that the vehicle pair $\{i, j\}$ is colliding. Therefore, the set

$$\mathscr{C}_{ij} = \{\xi_i, \xi_j \in \mathbb{R}^3 \times \mathbb{R}^3 \mid h_{ij}(\xi_i, \xi_j) \geq 0\} \tag{4.13}$$

defines the safe set where the system trajectories are allowed to remain.

**Remark.** Problem (4.11) is a non-convex optimization. That means that gradient-based solvers can get stuck at local minima. Therefore, a specific solver able to compute all the minimum candidates was developed.

The partial derivatives of (4.12) with respect to the position $p_j = [\, x_j \;\; y_j \,]^\top$ and orientation $\phi_j$ of the $j$-th vehicle are:

$$\frac{\partial E_j(\rho)}{\partial p_j} = I_2\,, \quad \frac{\partial E_j(\rho)}{\partial \phi_j} = \begin{bmatrix} -a_j\cos(\rho)\sin(\phi_j) - b_j\sin(\rho)\cos(\phi_j) \\ a_j\cos(\rho)\cos(\phi_j) - b_j\sin(\rho)\sin(\phi_j) \end{bmatrix} \tag{4.14}$$

Using (4.9), (4.10) and (4.12), the derivatives of (4.11) with respect to the poses of vehicles $i$ and $j$ are given by, respectively:

$$\frac{\partial h_{ij}}{\partial p_i} = -\frac{\partial h_i}{\partial \delta_i}(\delta_i,\phi_i)\Big|_{\delta_i = E_j(\rho^\star)-p_i}, \qquad \frac{\partial h_{ij}}{\partial \phi_i} = \frac{\partial h_i}{\partial \phi_i}(\delta_i,\phi_i)\Big|_{\delta_i = E_j(\rho^\star)-p_i} \tag{4.15}$$

$$\frac{\partial h_{ij}}{\partial p_j} = \frac{\partial h_i}{\partial \delta_i}(\delta_i,\phi_i)\Big|_{\delta_i = E_j(\rho^\star)-p_i}, \qquad \frac{\partial h_{ij}}{\partial \phi_j} = \left(\frac{\partial E_j(\rho^\star)}{\partial \phi_j}\right)^{\mathsf{T}} \frac{\partial h_i}{\partial \delta_i}(\delta_i,\phi_i)\Big|_{\delta_i = E_j(\rho^\star)-p_i} \tag{4.16}$$

with $\rho^\star$ being the argmin of optimization (4.11). Using the partial derivatives in (4.15) and (4.16), the time derivative of $h_{ij}$ as a function of the control inputs $u_i$ and $u_j$ yields

$$\dot{h}_{ij}(\xi_i,\xi_j,u_i,u_j) = \left(\frac{\partial h_{ij}}{\partial p_i}\right)^{\mathsf{T}} \begin{bmatrix}\cos\phi_i \\ \sin\phi_i\end{bmatrix} v_{f_i} + \frac{\partial h_{ij}}{\partial \phi_i}\omega_i + \left(\frac{\partial h_{ij}}{\partial p_j}\right)^{\mathsf{T}} \begin{bmatrix}\cos\phi_j \\ \sin\phi_j\end{bmatrix} v_{f_j} + \frac{\partial h_{ij}}{\partial \phi_j}\omega_j \tag{4.17}$$

Therefore, the set of kinematic controls that prevent the $i$-th vehicle from colliding against the $j$-th vehicle is given by

$$\mathbb{K}_{\text{vcbf}_j}(\xi_i,\xi_j) = \left\{(u_i,u_j) \in \mathbb{R}^2 \times \mathbb{R}^2 : \dot{h}_{ij}(\xi_i,\xi_j,u_i,u_j) + \beta h_{ij}(\xi_i,\xi_j) \geq 0\right\} \tag{4.18}$$

where here the extended class $\mathscr{K}$ function was simply chosen as the linear function $\beta(h_{ij}) = \beta h_{ij}$, $\beta > 0$.

#### 4.2.2.3 Spline-based Barriers

Next, consider $M$ B-splines $B_k : \mathbb{R} \to \mathbb{R}^2$, $k = 1,\cdots,M$, with $B_k(\rho)$, $\rho \in \mathbb{R}$ being an arbitrary point on the $k$-th B-spline curve, and gradient given by $\nabla B_k(\rho) \in \mathbb{R}^2$. Considering the $i$-th vehicle and the $k$-th B-spline $B_k$, similarly to the barrier for the vehicle pair (4.11), define the following spline-based barrier candidate:

$$h_{s_{ik}}(\xi_i) = \min_{\rho \in \mathbb{R}} h_i(B_k(\rho)) \tag{4.19}$$

Notice how $h_{s_{ik}}$ only depends on the state of the $i$-th vehicle $\xi_i$. An illustration of vehicle barriers and B-spline barriers can be seen in Figure 4.9.

**Remark.** The optimization problem (4.19) consists of minimizing the value of the quadratic form (4.8) over the B-spline curve. Notice that $h_{s_{ik}} < 0$ only occurs when $(B_k(\rho^\star)-p_i)^{\mathsf{T}} H_i(\phi_i)(B_k(\rho^\star)-p_i) < 1$ at the solution $\rho^\star = \text{argmin}_{\rho\in\mathbb{R}} h_i(B_k(\rho))$, which means that the $k$-th B-spline and $\mathscr{V}_i$ intersect in more than just a tangential point. That means that the $i$-th vehicle is colliding with the $k$-th B-spline.

Using (4.8) and the B-spline gradient, the partial derivatives of (4.19) with respect to $p_i$ and $\theta_i$ are given by:

$$\frac{\partial h_{s_{ik}}}{\partial p_i} = -\frac{\partial h_i}{\partial \delta_i}(\delta_i,\phi_i)\Big|_{\delta_i = B_k(\rho^\star)-p_i}, \qquad \frac{\partial h_{s_{ij}}}{\partial \phi_i} = \frac{\partial h_i}{\partial \theta_i}(\delta_i,\phi_i)\Big|_{\delta_i = B_j(\rho^\star)-p_i} \tag{4.20}$$

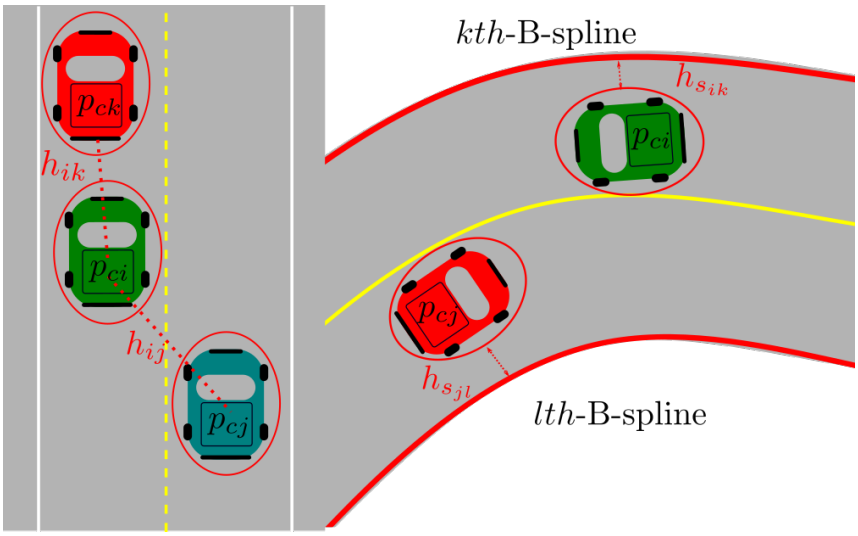



Figure 4.9: Proposed barrier candidates: $h_{ij}$ prevents collisions between vehicles $i$ and $j$, while $h_{s_{ik}}$ prevents collisions between vehicle $i$ and the $k$-th B-spline curve.

with $\rho^\star$ being the argmin of optimization (4.19). Using the partial derivatives in (4.20), the time derivative of $h_{s_{ij}}$ as a function of the control input $u_i$ is

$$\dot{h}_{s_{ik}}(\xi_i, u_i) = \left(\frac{\partial h_{s_{ij}}}{\partial p_i}\right)^{\mathsf{T}} \begin{bmatrix} \cos\phi_i \\ \sin\phi_i \end{bmatrix} v_{f_i} + \frac{\partial h_{s_{ij}}}{\partial \phi_i}\omega_i \tag{4.21}$$

Therefore, the set of kinematic controls that prevent the $i$-th vehicle from colliding against the $k$-th B-spline is given by

$$\mathbb{K}_{\text{scbf}_k}(\xi_i) = \left\{u_i \in \mathbb{R}^2 : \dot{h}_{s_{ik}}(\xi_i, u_i) + \beta h_{s_{ik}}(\xi_i) \geq 0\right\} \tag{4.22}$$

#### 4.2.2.4 Proposed QP Controller for Safe Path Following

**Assumption 5.** The $i$-th vehicle has access to its own state $\xi_i$ and to estimates $\hat{\xi}_j$, $\hat{u}_j$ $(j \neq i)$ of the states and control signals of other nearby vehicles, respectively.

Assumption 5 considers the existence of a centralized communication system that provides the exchange of data between vehicles, enabling the knowledge of their relative positions. Additionally, this system also provides to the controllers with geometric information about the lane limits and paths to be followed. Ideally, smart cities are required to provide reliable data about the vehicle's surroundings to its autonomous control system.

Each vehicle is controlled by the minimum-norm QP controller with multiple CBFs, similarly

to (2.41), combining path following stabilization and multiple collision avoidance objectives:

$$u_i^\star = \underset{(u_i,\mu)\in\mathbb{R}^3}{\operatorname{argmin}} \|u_i\|^2 + q\mu^2 \tag{4.23}$$

$$s.t.\ \dot{V}(\xi_i, u_i, \gamma_i, \dot{\gamma}_i) \leq -\alpha V(\xi_i) + \mu \qquad \textbf{(CLF)}$$

$$\dot{h}_{ij}(\xi_i, \hat{\xi}_j, u_i, \hat{u}_j) \geq -\beta h_{ij}(\xi_i, \hat{\xi}_j), \quad j \in \{1, \cdots, N\} \qquad \textbf{(vCBF-j)}$$

$$\dot{h}_{s_{ik}}(\xi_i, u_i) \geq -\beta h_{s_{ik}}(\xi_i), \quad k \in \{1, \cdots, M\} \qquad \textbf{(sCBF-k)}$$

where $q, \alpha, \beta > 0$. The proposed controller (4.23) is a Quadratic Program on the decision variables $(u_i, \mu) \in \mathbb{R}^3$. It minimizes the control effort $\|u_i\|$ while trying to keep $u_i \in \mathbb{K}_{\text{clf}}$ to stabilize the path following error by means of the first (CLF) constraint. The $N$ vCBF constraints keep $u_i \in \mathbb{K}_{\text{vcbf}_1} \cap \cdots \cap \mathbb{K}_{\text{vcbf}_N}$, avoiding collisions between nearby vehicles: notice that (4.23) needs information about the state $\xi_j$ and control signal $u_j$ of the $j$-th neighbor vehicle, thus making Assumption 5 necessary. Similarly, the $M$ sCBF constraints keep $u_i \in \mathbb{K}_{\text{scbf}_1} \cap \cdots \cap \mathbb{K}_{\text{scbf}_M}$, avoiding collisions between vehicle $i$ and the B-splines defining the lane limits.

### 4.2.3 Simulation Results

The simulation environment was designed in Python and is publicly available at `https://github.com/C2SR/spline_cbfs`. It consists of a complete framework for the simulation of any number of vehicles, paths, spline barriers, and controllers. The simulation setup can be fully specified by a Python script defining objects for the vehicles, paths, and barriers used by the QP controllers, among other simulation parameters. Special *.json* files for the splines can be created and dynamically modified, favoring the design of arbitrary geometries for the paths and the spline barriers defining the lane limits. The results of a simulation are also saved in a *.json* file for later use. The repository also implements graphical functionality for animating a predefined simulation and plotting the results, using its corresponding saved *.json* files.

Next, we present and discuss the results obtained with the simulation of a navigation task composed of two vehicles following spline-based paths on a three-way road, whose limits are defined by three spline barrier functions. The trajectories of the vehicles (blue rectangles) for $10s$ of simulation at four specific timestamps are shown in Fig. 4.10.

Vehicle 1 is the one following the blue path (blue dashed line) $p_{d_1}$, while vehicle 2 follows the purple path (purple dashed line) $p_{d_2}$. The lane limits are represented by the red B-spline curves. The red point in front of each robot at each path corresponds to the virtual point position $p_{d_i}(\gamma_i)$ (for the $i$-th vehicle). The blue ellipse around vehicle $i$ represents the superlevel set corresponding to $h_i(\delta_i, \phi_i) = 0$. The green points on the ellipses represent $E_j(\rho^\star)$, where

$$\rho^\star = \underset{\rho\in\mathbb{R}}{\operatorname{argmin}} h_i(E_j(\rho)),$$

that is, the point on the ellipse of the $j$-th vehicle that minimizes the barrier $h_i$ (relative to the $i$-th vehicle). The controller for the $i$-th vehicle is given by (4.23), which has access to the $i$-th vehicle

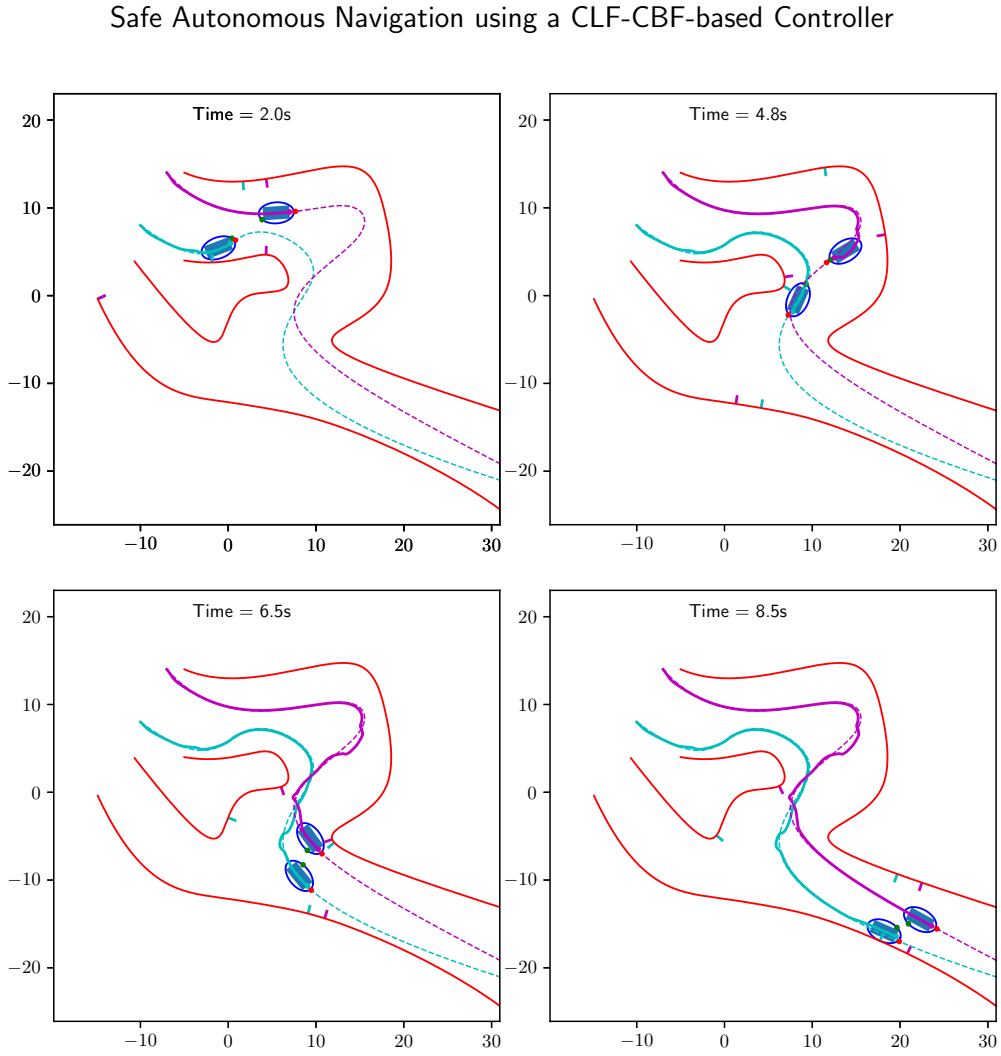



Figure 4.10: Simulation environment with two vehicles and three B-spline barriers.

pose and to the poses of other nearby vehicles. For implementing the spline barrier constraints, the controller (4.23) also has access to all the lane splines. Each vehicle is assumed to have a high-performance velocity controller that receives velocity commands from the QP-based kinematic controller (4.23). These controllers use saturations to keep the velocity commands within the vehicle operational limits.

From the previous theoretical developments, the ellipses delimiting the vehicles cannot overlap due to the vehicle barrier constraints. However, notice how the *paths* overlap in the center of the environment: the vehicles are required to follow their specific paths without colliding with each other (and the lane limits). At $t = 2s$, vehicle 2 has already converged and remained following its corresponding path, while vehicle 1 initially converged but deviated slightly upon approaching close to one of the lane limits. This a consequence of the activation of one of the **sCBF** constraints on the QP (4.23). At $t = 4.8s$, vehicle 1 is at the path intersection, and we can notice how the trajectory of the second vehicle got affected. Some moments before, vehicle 2 has deviated from

its path due to the activation of the **vCBF** constraints relative to vehicle 1 ($h_{21}$), but returned to its path shortly after. At $t = 6.5s$, the trajectories of both vehicles got affected by the proximity with the lane limits (due to the shrinking of available space at the center of the road), which means that the **sCBF** constraints have activated for both vehicle controllers. However, both converged back to their specific paths after leaving the intersection. At $t = 8.5s$, vehicle 1 has again deviated slightly from its path due to the proximity to the lower lane limit.

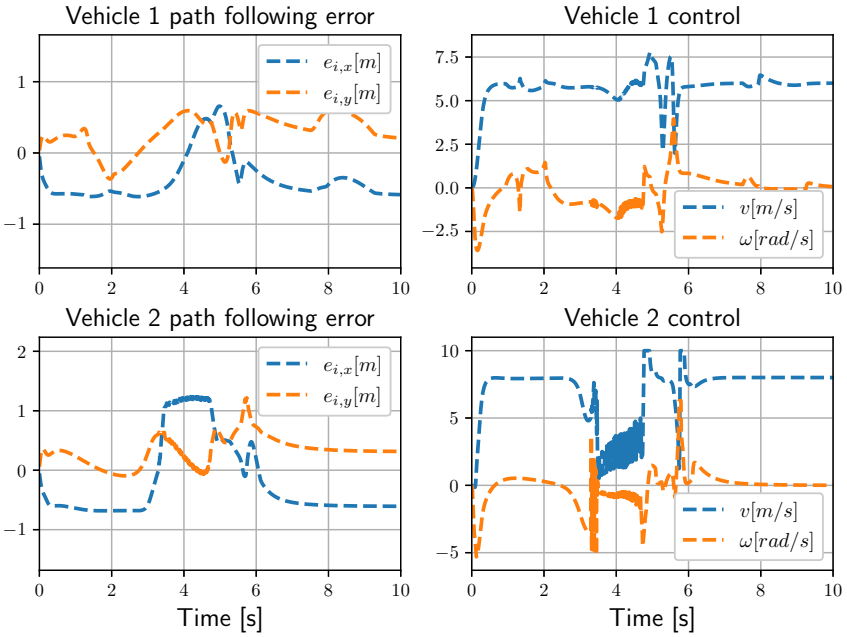



Figure 4.11: Path following errors and control signals.

The path following errors and control signals are shown in Fig. 4.11. The path-following errors remain bounded during the whole simulation, only increasing when the vehicles are deviating from their paths to avoid some imminent collision. The vehicle control signals remain within the operational speed limits during the whole simulation. Extensive simulations under many different configurations (more vehicles, paths, spline barriers, and with different parameters) were performed, and the proposed controller has showed satisfactory performance in a wide variety of scenarios.

## 4.3 CLF-CBF-based Safe Cooperative Path Following Control

In this section, a novel framework to solve the Cooperative Path-Following (CPF) control problem with collision avoidance guarantees is presented, even in case two or more paths intersect. The methodology is based on the QP-based control framework with CLFs and multiple CBFs to achieve path stabilization, swarm consensus and collision avoidance guarantees. To avoid the problem of (possible) deadlocks arising from trying to fulfill these (possible) inconsistent objectives, we propose a technique that assigns dynamic priorities to the vehicles. This is the first attempt made on this thesis to solve the issue of stable equilibrium points (deadlock) that can arise

under the CLF-CBF formulation. A general theoretical treatment for the problem is presented at Chapter 5, but in this section we address the problem in a specific application. Numerical simulations demonstrate the effectiveness of the proposed approach, in the case of a swarm of vehicles in a 2D environment.

### 4.3.1 Problem Formulation

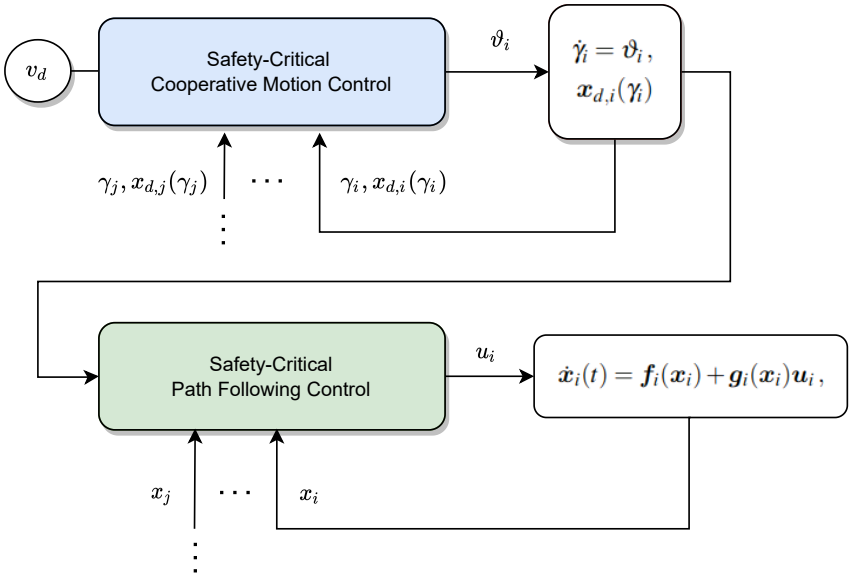



Figure 4.12: Control diagram for safety-critical Cooperative Path Following: an inner loop safety-critical PF controller handles safe path following, while an outer loop safety-critical cooperative motion controller handles the coordination objective for the team of robots while maintaining a common speed assignment $v_d$ for the team.

Consider a team of $N$ vehicles, with set of integer indexes for the vehicles defined as $\mathbb{I} = \{1,2,...,N\}$, as in Section 2.3.3. The $i$-th vehicle state is given by its planar inertial position $x_i = p_i \in \mathbb{R}^2$ in an inertial frame $\mathscr{I}$, with dynamical equation being a simple integrator $\dot{x}_i = u_i \in \mathbb{R}^2$, where $u_i$ are the robot's velocities in the $x$ and $y$ directions, respectively.

**Remark.** Compared to Section 4.2, here the model of the vehicles is significantly simplified, for simplicity. However, similar results can be achieved by using the unicycle model of Section 4.2.

As shown by Fig. 4.12, the safety-critical Cooperative Path Following (CPF) control problem can be divided in two sub-problems, which thereby can be solved by two controllers in cascade, namely a cooperative motion control module and a PF control module.

#### 4.3.1.1 Safety-Critical Path Following Control

The first sub-problem is safety-critical Path Following (PF) control, and consists of steering the vehicles towards the desired paths while avoiding collisions. Let $x_{d_i} : \mathbb{R} \to \mathbb{R}^2$ be $C^1$ functions representing the desired paths for each vehicle, which are parameterized by a path variable $\gamma_i \in \mathbb{R}$.

**Remark.** In this section, we assume that the path functions can intersect at a finite number of points. That is, for a pair of vehicles $i, j \in \mathbb{I}$, there may exist a finite number of pairs $\{\gamma_i, \gamma_j\} \in \mathbb{R} \times \mathbb{R}$ such that $x_{d_i}(\gamma_i) = x_{d_j}(\gamma_j), i, j \in \mathbb{I}$.

Defining the PF error for the $i$-th vehicle as $e_i = x_i - x_{d_i}(\gamma_i)$, the formal objectives of safety-critical PF control are twofold:

i to design feedback control laws $u_i$ for the vehicles such that the origin of the path following errors $e_i = 0$, $\forall i \in \mathbb{I}$ are asymptotically stable. That is, it is desired to steer the vehicles towards their paths, such that $x_i$ stabilizes around $x_{d_i}(\gamma_i)$, $\forall i \in \mathbb{I}$.

ii to satisfy safety conditions of the kind $\|x_i - x_j\| \geq D_s$ for $i, j \in \mathbb{I}$ and some positive distance $D_s$, designed to avoid collisions between neighboring vehicles.

#### 4.3.1.2 Safety-Critical Cooperative Motion Control

The second sub-problem is safety-critical cooperative motion control, consisting of achieving a coordination objective on the vehicle swarm while sending non-colliding virtual point references $p_{d_i}(\gamma_i)$ and $p_{d_j}(\gamma_j)$ to a pair of vehicles $\{i, j\} \in \mathbb{I}$. This can be done by explicitly controlling the path variables to coordinate the vehicles along their paths, achieving a consensus objective while also being able to take necessary actions on the path dynamics to avoid collisions.

The progression of each virtual point $x_{d_i}(\gamma_i)$ along the paths can be controlled by imposing a dynamic law for the path variable $\gamma_i$:

$$\dot{\gamma}_i = \vartheta_i, \quad i \in \mathbb{I}$$

where $\vartheta_i$ is a control signal to be designed, as shown in Fig. 4.12. The virtual points should also satisfy the desired speed assignments $\dot{\gamma}_i = v_d$, $\forall i \in \mathbb{I}$, where $v_d$ is a desired nominal speed assignment for the vehicles. For convenience, define vectors $\boldsymbol{\gamma} = [\ \gamma_1\ \cdots\ \gamma_N\ ]^\mathsf{T}$ and $\boldsymbol{\vartheta} = [\ \vartheta_1\ \cdots\ \vartheta_N\ ]^\mathsf{T}$, comprising all path variables and their respective path control signals, respectively.

**Remark.** Note that $\|\dot{x}_{d_i}(\gamma_i)\| = \|\nabla x_{d_i}(\gamma_i)\|\,|\dot{\gamma}_i|$. Therefore, the magnitude of $\dot{\gamma}_i$ is equivalent to $\|\dot{x}_{d_i}(\gamma_i)\|$ if the norm of the path derivative is unitary, that is, $\|\nabla x_{d_i}(\gamma_i)\| = 1$. Particularly, this is the case if $\gamma_i$ represents the arc length of the $i$-th path.

Assume that all vehicles communicate with each other. Given the path variable vector $\boldsymbol{\gamma} \in \mathbb{R}^N$, the formal objectives of the safety-critical cooperative motion control problem are twofold:

i to design a decentralized feedback control law $\vartheta_i$ such that the positions of the virtual points are synchronized, that is, $|\gamma_i - \gamma_j| = 0$, $\forall i, j \in \mathbb{I}$ as $t \to \infty$.

ii to satisfy the following safety-critical conditions: $\|x_{d_i}(\gamma_i) - x_{d_j}(\gamma_j)\| \geq D_s$, $i, j \in \mathbb{I}$, designed to keep the virtual points of neighboring vehicles from a safe distance $D_s$ from each other.

Notice that objective (ii) effectively avoids collisions between neighbour virtual points at path intersections or at regions where the paths are very close together.

**Remark.** Notice that objectives (i), (ii) and the speed assignments $\vartheta_i = \vartheta_d$, $i \in \mathbb{I}$ can be conflicting. In this case, the consensus condition (i) or the speed assignments would have to be broken in order to satisfy the safety constraints (ii).

### 4.3.2 CLF-CBF-based Controller for Safe Cooperative Path Following

Next, we present the proposed safety-critical controllers for solving the presented sub-problems illustrated in Fig. 4.12. First, define the barrier-like function $h : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$:

$$h(y_i, y_j) = \|y_i - y_j\| - D_s\,, \qquad y_i \neq y_j \tag{4.24}$$

where $D_s$ is a positive constant and $y_i$ are $y_j$ are $C^1$ arbitrary signals. Then, the time derivative of (4.24) is given by

$$\dot{h}(y_i, y_j, \rho_i, \rho_j) = \frac{(y_i - y_j)^\mathsf{T}}{\|y_i - y_j\|}(\dot{y}_i - \dot{y}_j) \tag{4.25}$$

#### 4.3.2.1 CLF-CBF-based Safe Path Following Control

Define the CLF candidate for the stabilization of the $i$-th vehicle as $V(e_i) = \frac{1}{2}\|e_i\|^2$. Similarly to the previous section, its time derivative is given by $\dot{V}(x_i, u_i, \gamma_i, \dot{\gamma}_i) = e_i^\mathsf{T} u_i - e_i^\mathsf{T} \nabla x_{d_i}(\gamma_i)\dot{\gamma}_i$.

Regarding the CBF candidates for the vehicle safety objectives, using signals $y_i = x_i$ and $y_j = x_j$ in (4.24), notice that condition $h(x_i, x_j) \geq 0$ adequately represents safety requirement (ii) from Section 4.3.1.1, making $h(x_i, x_j)$ a suitable CBF candidate for avoiding collisions for the vehicle pair $\{i, j\} \in \mathbb{I}$. Equation (4.25) can be used to compute the time derivative of $h(x_i, x_j)$, with $\dot{x}_k = u_k$ $(k = i, j)$.

**Assumption 6.** Assume that the $i$-th vehicle PF controller has access to estimates $\hat{x}_j$ and $\hat{u}_j$ for the positions $x_j$ and control actions $u_j$ from neighbour vehicles $j \in \mathbb{I}$.

Under Assumption 6, if feasible, the QP

$$\begin{aligned} u_i^\star, \delta^\star = \underset{(u_i, \delta) \in \mathbb{R}^{m+1}}{\operatorname{argmin}} \; & \|u_i\|^2 + p_1 \delta^2 \\ s.t.\ & \dot{V}(e_i, u_i, \gamma_i, \vartheta_i) + \bar{\alpha}_1 V(e_i, \gamma_i) \leq \delta \\ & C_{ij} \geq 0\,, \qquad \forall j \in \mathbb{I} \end{aligned} \tag{4.26}$$

$$C_{ij} = \dot{h}(x_i, \hat{x}_j, u_i, \hat{u}_j) + \beta_1 h(x_i, x_j) \tag{4.27}$$

with $\bar{\alpha}_1, \beta_1 > 0$, solves the safety-critical PF control problem for the $i$-th vehicle, ensuring the stabilization of the path error $e_i$ to a neighborhood of the origin and satisfying the safety requirements (ii) from Section 4.3.1.1. Equation (4.26) describes the inner loop controller from Fig. 4.12.

**Remark.** Assumption 6 is needed in order to implement the CBF-like constraints $C_{ij} \geq 0$ in (4.26), since the distances $h(x_i, x_j)$ and their time derivatives must be computed for each neighbor

vehicle $j \in \mathbb{I}$. Additionally, $x_{d_i}(\gamma_i)$ is also needed for the computation of the path error $e_i$ (refer to Fig. 4.12).

Finally, note that (4.26) might not be feasible due to the multiple CBF constraints, which might be incompatible at some configurations.

### 4.3.3 CLF-CBF-based Safe Cooperative Motion Control

Define the CLF candidate for cooperative consensus for the $i$-th vehicle as

$$V_{c_i}(\gamma) = \sum_{j \in \mathbb{I}} (\gamma_i - \gamma_j)^2, \qquad j \in \mathbb{I}. \tag{4.28}$$

Regarding the CBF candidates for the path safety objectives, using signals $y_i = x_{d_i}(\gamma_i)$ and $y_j = x_{d_j}(\gamma_j)$ in (4.24) to define $h(\gamma_i, \gamma_j) = h(x_{d_i}(\gamma_i), x_{d_j}(\gamma_j))$, notice that condition $h(\gamma_i, \gamma_j) \geq 0$ adequately represents the safety requirement (ii) from Section 4.3.1.2, making $h(\gamma_i, \gamma_j)$ a suitable CBF candidate for avoiding collisions between the pair of virtual points $x_{d_i}(\gamma_i)$, $x_{d_j}(\gamma_i)$. Equation (4.25) with $\dot{x}_{d_i}(\gamma_i, \vartheta_i) = \nabla x_{d_i}(\gamma_i)\, \vartheta_i$ and $\dot{x}_{d_j}(\gamma_j, \vartheta_j) = \nabla x_{d_j}(\gamma_j)\, \vartheta_j$ can be used to compute the time derivative of $h(\gamma_i, \gamma_j)$, written where as $\dot{h}(\gamma_i, \gamma_j, \vartheta_i, \vartheta_j)$.

**Assumption 7.** We assume that the $i$-th vehicle cooperative controller receives estimates of the path variables $\hat{\gamma}_j$ and path control actions $\hat{\vartheta}_j$ from neighbour vehicles $j \in \mathbb{I}$. These estimates are collected in vectors $\hat{\gamma} = [\, \hat{\gamma}_1 \;\cdots\; \gamma_i \;\cdots\; \hat{\gamma}_N \,]^\mathsf{T} \in \mathbb{R}^N$ and $\hat{\vartheta} = [\, \hat{\vartheta}_1 \;\cdots\; \vartheta_i \;\cdots\; \hat{\vartheta}_N \,]^\mathsf{T} \in \mathbb{R}^N$.

Then, with Assumption 7, if feasible, the QP

$$\begin{aligned} \vartheta_i^\star, \mu^\star = \; & \underset{(\vartheta_i, \mu) \in \mathbb{R}^2}{\operatorname{argmin}} \; \|\vartheta_i - v_d\|^2 + p_2 \mu^2 && (4.29) \\ & s.t.\; \dot{V}_{c_i}(\hat{\gamma}, \hat{\vartheta}) + \bar{\alpha}_2 V_{c_i}(\hat{\gamma}) \leq \mu \\ & \qquad\qquad C_{\gamma_{ij}} \geq 0, \qquad \forall j \in \mathbb{I} \\ C_{\gamma_{ij}} = \; & \dot{h}(\gamma_i, \hat{\gamma}_j, \vartheta_i, \hat{\vartheta}_j) + \beta_2 h(\gamma_i, \gamma_j) && (4.30) \end{aligned}$$

with $\bar{\alpha}_2, \beta_2 > 0$, guarantees consensus among the vehicles, satisfies the safety requirements (ii) from Section 4.3.1.2 and tries to fulfill the speed assignment for the vehicles by minimizing the control cost $\|\vartheta_i - v_d\|^2$. Equation (4.29) describes the outer loop controller from Fig. 4.12.

**Remark.** Assumption 7 is needed in order to implement the CBF-like constraints $C_{\gamma_{ij}} \geq 0$ in (4.29), since the distances $h(\gamma_i, \gamma_j)$ and their time derivatives must be computed for each neighbor vehicle $j \in \mathbb{I}$.

### 4.3.4 Priority Assignment for Safety-Critical CPF Control

For the closed-loop system resulting from control laws (4.26) and (4.29), if two or more paths are intersecting or close to each other, the vehicles could reach a deadlock configuration, where they

all stop close to an intersection to avoid colliding. In this section, we show that this issue can be solved in this specific problem by assigning priorities to the vehicles.

First, notice that given a pair $(i, j)$ of communicating vehicles with perfect knowledge about each other's positions and path variables, we have $h(x_i, x_j) = h(x_j, x_i)$, which means that if $h(x_i, x_j)$ is positive invariant for vehicle $i$, then $h(x_j, x_i)$ will also be positive invariant for vehicle $j$. Similarly, $h(x_{d_i}(\gamma_i), x_{d_j}(\gamma_j)) = h(x_{d_j}(\gamma_j), x_{d_i}(\gamma_i))$. Therefore, between pairs of vehicles, assuming that the positions of the pair are accurately known by the two vehicles, only one of the pairwise barriers has to be effectively implemented to avoid collision.

Define a priority vector for the swarm of vehicles as:

$$\alpha = \{\alpha_1, \cdots, \alpha_N\} \in \mathbb{R}^N \tag{4.31}$$

where $\alpha_i$, $i \in \mathbb{I}$ are *priorities* assigned to each vehicle (not to be confused with $\bar{\alpha}_1$ and $\bar{\alpha}_2$, that is, the class $\mathscr{K}$ CLF constants). Vehicle $i$ has priority over vehicle $j$ if $\alpha_i > \alpha_j$. Additionally, $\alpha_i \neq \alpha_j, i \neq j$.

**Remark.** The swarm priorities (4.31) are assumed to be given by some external supervisor that assigns unique priorities to each vehicle. In a context of self-driving cars, for example, this supervisor would be an algorithm that determines the priority of each vehicle according to the current traffic configuration.

Considering the $i$-th vehicle, we propose a modification to (4.26) and (4.29), aiming to solve the deadlock problem in these specific configurations when the virtual points are close to a path intersection. Define a modified Lyapunov candidate for consensus as:

$$V_{c_i}(\gamma) = \sum_{j \in \mathbb{I}} (\gamma_i - \gamma_j)^2 \mathbf{1}_{\Delta\alpha_{ji} \geq 0}, \tag{4.32}$$

where $\Delta\alpha_{ji} = \alpha_j - \alpha_i$ and $\mathbf{1}_{x \geq 0}$ is an indicator function for the set of positive reals. Then, the following modified versions of QPs (4.26), (4.29)

$$\begin{aligned} u_i^\star, \delta^\star = \underset{(u_i, \delta) \in \mathbb{R}^{m+1}}{\operatorname{argmin}} \; & \|u_i\|^2 + p_1 \delta^2 \\ s.t. \; & \dot{V}(e_i, u_i, \gamma_i, \vartheta_i) + \bar{\alpha}_1 V(e_i, \gamma_i) \leq \delta \\ & C_{ij} \geq 0, \quad \forall j \in \mathbb{I} \; s.t. \; \alpha_i < \alpha_j \end{aligned} \tag{4.33}$$

$$\begin{aligned} \vartheta_i^\star, \mu^\star = \underset{(\vartheta_i, \mu) \in \mathbb{R}^2}{\operatorname{argmin}} \; & \|\vartheta_i - v_d\|^2 + p_2 \mu^2 \\ s.t. \; & \dot{V}_{c_i}(\hat{\gamma}, \hat{\vartheta}) + \bar{\alpha}_2 V_{c_i}(\hat{\gamma}) \leq \delta \\ & C_{\gamma_{ij}} \geq 0, \quad \forall j \in \mathbb{I} \; s.t. \; \alpha_i < \alpha_j \end{aligned} \tag{4.34}$$

solve the deadlock problem at path intersections, enabling the vehicles to continue moving through their assigned paths in an order according to their assigned priorities (the one with the highest priority goes first, followed by the rest in decreasing priority order).

**Remark.** From QPs (4.33), (4.34), given pair of agents $i, j \in \mathbb{I}$, the CBF constraints $C_{ij} \geq 0$, $C_{\gamma_{ij}} \geq 0$ from (4.27) and (4.30) are only implemented for agent $i$ if $\alpha_i < \alpha_j$. Besides, from (4.32), a contribution $|\gamma_i - \gamma_j|$ from vehicle $j$ is only summed up to $V_{c_i}(\hat{\gamma})$ if $\alpha_i < \alpha_j$. This strategy removes redundant collision constraints and redundant contributions to the consensus Lyapunov candidate, according to the priorities of the swarm.

#### 4.3.4.1 Consistency Graphs for Solving Conflicts

This section proposes a simple algorithm for dynamic priority assignment, with the objective of modifying the desired vehicle priorities in order to ensure that a maximum of only one active barrier constraint is implemented by QPs (4.33) and (4.34). Then, if a vehicle with lower priority implements two or more CBF constraints, only one of them will be active, thus ensuring the feasibility of the corresponding QP.

For the outer loop QP (4.34), define the *swarm consistency graph* $\mathbb{C} = \{\mathbb{I}, \mathbb{E}\}$, whose nodes are formed by the vehicles and the set of edges $\mathbb{E} \subset \mathbb{I} \times \mathbb{I}$ is defined as:

$$\mathbb{E} = \{(i, j) \in \mathbb{I} \times \mathbb{I} \,|\, C_{\gamma_{ij}} = C_{\gamma_{ji}} = 0\} \tag{4.35}$$

Therefore, considering a neighbour agent $j$, the swarm consistency graph has a connection between nodes $(i, j)$ if the corresponding $C_{\gamma_{ij}} = C_{\gamma_{ji}}$ constraint is active. A similar consistency graph can be defined for the inner loop QP (4.33), by changing $C_{\gamma_{ij}}$ to $C_{ij}$ on (4.35). Figure 4.13 illustrates the concept by means of an example with three vehicles.

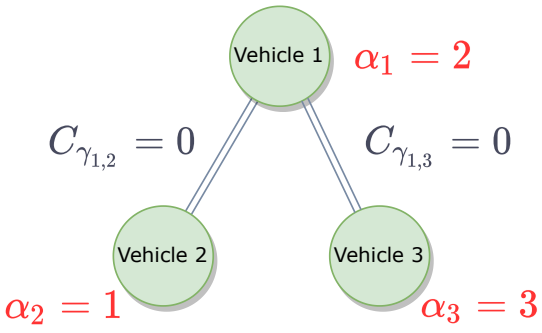



Figure 4.13: Swarm consistency graph for the cooperative QP, showing the active constraints on a given moment (with three vehicles). Starting from vehicle 3 (highest priority), a search on the graph visits the nodes in a decreasing order of priority.

Starting from the node with maximum priority in $\mathbb{C}$, we must ensure that the priorities of the nodes being discovered by a *depth-first-search* (DFS) algorithm are decreasing as the graph is explored. This way, at a given time, vehicles with smaller priorities will always have just one active barrier constraint. For example, at Fig. 4.13, starting with maximum priority of $\alpha_3 = 3$ (vehicle 3), the swarm priority vector that does not result in risk of inconsistent barrier constraints is $\alpha = \{\alpha_1, \alpha_2, \alpha_3\} = \{2, 1, 3\}$. Then, for this priority vector, in QP (4.34), vehicle 3 does not implement any barriers (moves freely, without being affected by the others), vehicle 1 implements

$C_{\gamma_{13}} \geq 0$ (is only affected by vehicle 3), and vehicle 2 implements $C_{\gamma_{21}} \geq 0$, $C_{\gamma_{23}} \geq 0$ (is affected by vehicles 1 and 3). However, in Fig. 4.13, only $C_{\gamma_{21}} \geq 0$ is active ($C_{\gamma_{21}} = 0$) in vehicle 2, which means it is only being affected immediately by vehicle 1. This idea is explained in a precise manner with the aid of Algorithm 1 and a sorting Depth First Search (DFS) algorithm.

**Algorithm 1** Priority Assignment Algorithm

Start with $\alpha = \alpha_d$, where $\alpha_d = [\ \alpha_{d_1}\ \cdots\ \alpha_{d_N}\ ]$ is a given vector of *desired* vehicle priorities.
**for** each connected component $G \in \mathscr{C}$ of size $m$ formed by nodes $\{i_1, i_2, \cdots, i_m\}$ **do**
  $i_{\max} = \text{argmax}\{\alpha_{i_1}, \alpha_{i_2}, \cdots, \alpha_{i_m}\}$
  **Sorting DFS**(G, $i_{\max}$, $\alpha$)
**end for**
Return the resulting vector of consistent priorities $\alpha$.

Algorithm 1 performs a sorting Depth First Search (DFS) algorithm on each connected component of the swarm consistency graph $\mathbb{C}$, each starting at the maximum priority node of the corresponding connected component. The sorting DFS algorithm explores each connected component many times, swapping priorities between pairs of nodes as they are visited. If the priority of the previously visited node is smaller than the currently visited one, their priorities are swapped. This strategy effectively sorts the priorities in decreasing order, starting from the first visited node (maximum priority) to the deepest layers of nodes in the graph, to which smaller priorities will be assigned.

### 4.3.5 Simulation Results

A Python-based simulation environment was developed and is publicly available at `https://github.com/C2SR/QP-CPF`. A simulation with three communicating vehicles following straight line paths intersecting in a central point was set up. This configuration for the simulation was chosen in such a way that, if the vehicles satisfy their speed assignments $\vartheta_1 = \vartheta_2 = \vartheta_3 = v_d = 2\,m/s$ and maintain consensus on their path variables for the entire simulation, their corresponding virtual points would meet in the intersection point, resulting in a collision.

The behaviour of the swarm using controllers (4.26), (4.29) is shown in Fig. 4.14. Initially the vehicles converge to their paths while maintaining consensus, but eventually they reach a deadlock configuration, avoiding collisions among them, but without satisfying their speed assignments. In fact, without assigning priorities to the vehicles, the safety-critical CPF problem cannot be solved due to the presence of intersecting paths.

Figure 4.15 shows many simulation timestamps using the proposed controllers (4.33)-(4.34) with prioritization for the vehicles, under the same simulation conditions. Figure 4.16 shows the results for the vehicle path variables and speeds. The controllers follow the cascade structure shown in Fig. 4.12. Vehicles 1, 2 and 3 are the light blue, purple and green vehicles, respectively, and their desired priorities are given by $\alpha_{d_1} = 2$, $\alpha_{d_2} = 3$ and $\alpha_{d_3} = 1$ (the rightmost vehicle has the highest desired priority, and priorities decrease for the other vehicles in the clockwise direction). As shown in Fig. 4.15(a) and Fig. 4.16, before $2\,s$ of simulation, the vehicles initially converge

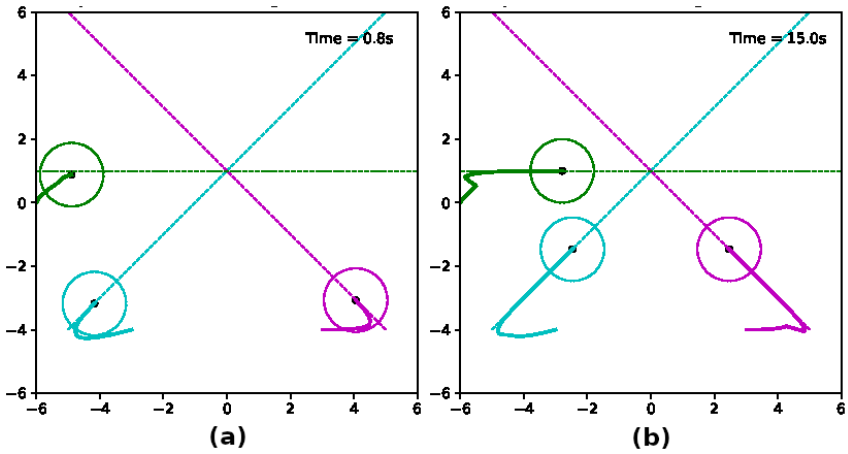



Figure 4.14: Results for the safety-critical CPF control without prioritization. The vehicles meet a deadlock configuration when close to the path intersection. Consensus is maintained and collision is avoided, but the speed assignments cannot be satisfied, and the evolution of the vehicles along their paths is halted.

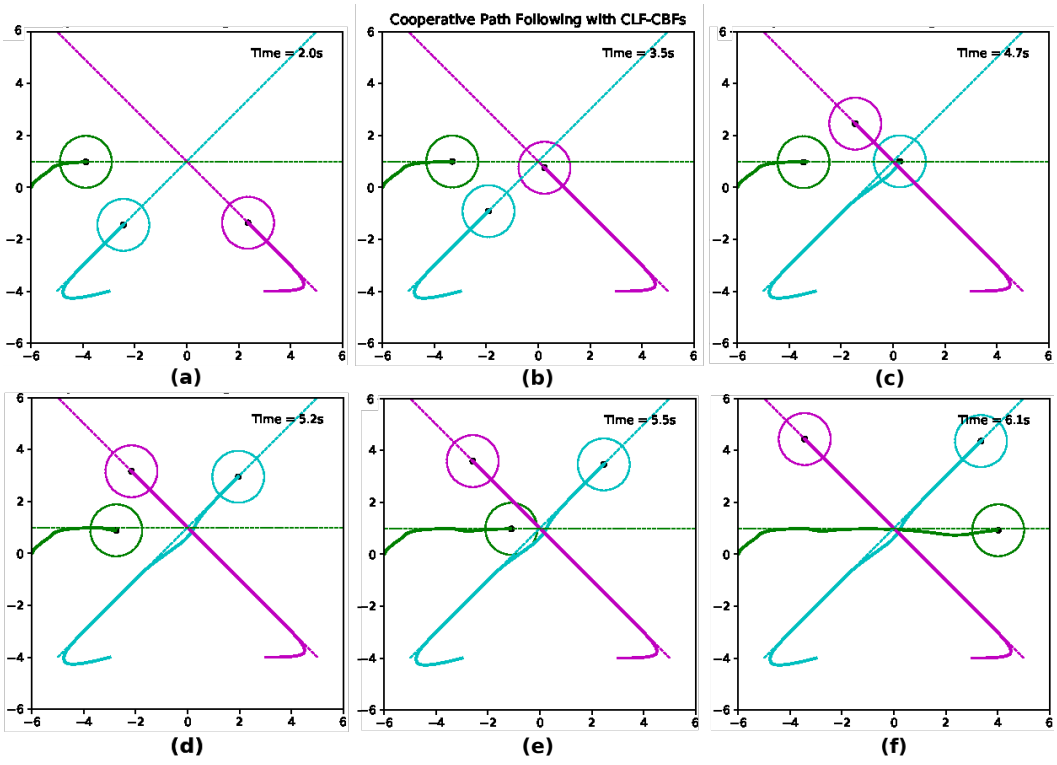



Figure 4.15: Results for the safety-critical CPF control with prioritization. In (a), the maximum priority is assigned to the rightmost vehicle, with priorities decreasing in the clockwise direction.

to their paths while maintaining consensus on their corresponding path variables. Eventually, at $3.5\,s$ of simulation, vehicle 2 (the one with highest priority, $\alpha_{d_2} = 3$) crosses the path intersection while the others wait, avoiding collision. After vehicle 2 has passed, between $4.7\,s$ and $5.2\,s$ [s] of simulation (Figs 4.15(c)-(d)), vehicle 1 ($\alpha_{d_1} = 2$) crosses and then, when its safe, vehicle 3

($\alpha_{d_3} = 1$) finally starts moving (Fig. 4.15(e)) and crosses the intersection between between $5.5\,s$ and $6.1\,s$ of simulation (Figs 4.15(e)-(f)).

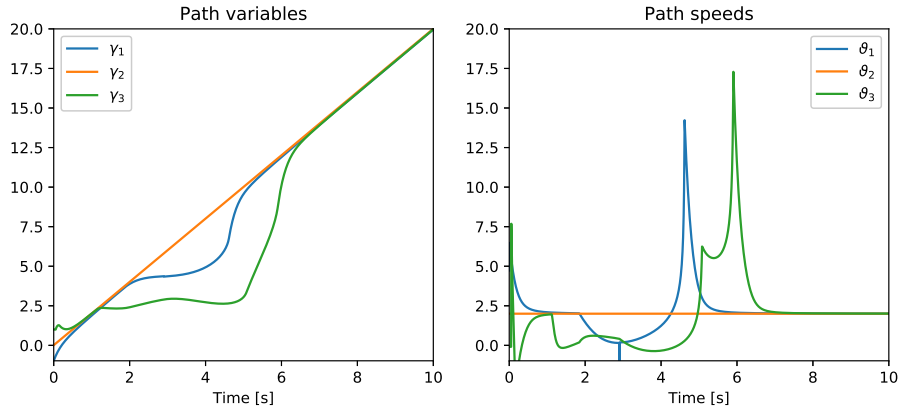



Figure 4.16: Path variables and speeds for the safety-critical CPF control with prioritization.

In Fig. 4.16, we notice that between $1.5\,s$ and $6\,s$ of simulation, the consensus on the path variables must be abandoned to avoid collisions at the intersection. Similarly, the path speed assignments for the lowest priority vehicles (1 and 3) must be abandoned, since they must wait for the highest priority vehicle (2) to keep moving. Notice the first peak for $\vartheta_1$ at $4.7\,s$, since it has to accelerate along its path to retrieve consensus with vehicle 2. A similar peak can be observed for $\vartheta_3$ at approximately $6.0\,s$, since it has to accelerate along its path to retrieve consensus with vehicle 1 and 2. After $6.0\,s$ of simulation, the vehicles are not at risk of collision and are able to satisfy their speed assignments $\vartheta_1 = \vartheta_2 = \vartheta_3 = v_d = 2.0\,m/s$, and also retrieve consensus among them.

The achieved behaviour is completely safe and in accordance with the desired assigned priorities for the vehicles. Furthermore, the priority assignment algorithm introduced in Section 4.3.4.1 ensures that the actual priorities used for implementation of the CBF constraints in controllers (4.33)-(4.34) are consistent and maintain the QPs feasible.

## 4.4 Concluding Remarks and Discussions

In this chapter, we started by designing a safety-filter controller for safe vehicle navigation, which was able to introduce an extra layer of safety in situations where the motion planner module fails to generate a reliable (safe) path. However, the safety-filter controller does not give explicit guarantees that stabilization will be maintained when the CBF constraint is active. The natural step forward was to directly incorporate CLFs in the path following control design. That was done in Sections 4.2 and 4.3, where we combine the CLF for achieving stabilization of the path following and coordination errors, and the CBFs to avoid collisions. Finally, in Section 4.3, we have acknowledged the problem of trajectory deadlock, when the conflicting requirements for vehicle

coordination and collision avoidance impedes the vehicle from advancing along their paths. We have addressed this issue by using an strategy of assigning dynamic priorities to the vehicles.

### 4.4.1 The safety filter strategy

In Section 4.1, we have presented a navigation framework for autonomous surface vessels consisting of two main blocks: situational awareness and maneuvering (motion planning/control). The former detects and estimates the position of nearby boats while the latter generates collision-free trajectories and drives the vessels through the desired path computing low level commands (force and torque). Additionally, a safety-critical controller adds an extra layer of safety to the system by ensuring that the vessel does not collide with obstacles due to errors in the motion planner or path following modules.

Future works include: (i) improving of the inner loop tracking performance by using more accurate descriptions for the vehicle dynamics; (ii) improving the computational performance of the YOLO-based object detection algorithm; (iii) taking into account the vehicle dynamics in the design of the safety-critical controller.

### 4.4.2 CLF-CBF-based Path Following

In Section 4.2, a safe PF controller for vehicle navigation with explicit safety guarantees was proposed. Provided that the paths are generated by means of an external path planning algorithm, the vehicles are able to converge and remain on their paths when no collision is imminent. When a collision is imminent (with another vehicle or with a lane limit), a CBF constraint from the optimization-based controller is activated (**vCBF** or **sCBF**, respectively) and the controller effectively modifies the vehicle behavior to avoid the collision from happening.

The proposed framework can also be employed for different models of vehicles, and the use of B-spline curves allows for a significant amount of freedom in the design of non-structured paths and lane geometries, which is a desirable feature if the framework is to be applied in real environments, such as the curved geometries of lanes and highways usually found in real cities.

Future works include (i) computational analysis of the algorithm performance under different scenarios, including more vehicles and lane limits; (ii) improving the optimization algorithm for computing the spline-based barrier at (4.19); (iii) taking into account the vehicle dynamics into the controllers, allowing for the explicit control of the vehicle accelerations.

### 4.4.3 Deadlock Avoidance in Cooperative Path Following

In Section 4.3, we have proposed a scheme for solving the safety-critical CPF control problem using the framework of CLFs and CBFs with prioritization for the vehicles. We have designed a QP-based CLF-CBF controller with a dynamic number of CBF constraints, effectively solving the problem of deadlock situations that usually arrives when multiple vehicles meet close to a path intersection. Additionally, we have proposed an algorithm to modify the assigned priorities for the

swarm, aiming to maintain feasibility of the QPs at all times. The results obtained with numerical simulations illustrate the efficiency and viability of the proposed methodology.

The proposed method can be combined with path planning algorithms for multiple vehicles, effectively reducing the computational burden of having to take path intersections into account when generating new paths for the swarm. This is specially important in applications where optimal paths have to be computed considering only local information for each vehicle.

# Chapter 5

# Developments on CBF-Based Safety-Critical Control

In this chapter, we provide a throughout analysis for two CBF-based QP controllers: (i) the safety-filter QP introduced in (2.40) and (ii) the CLF-CBF QP introduced in (2.41), both with multiple CBFs, each modeling distinct safe regions in the system state space. Our objective is to provide a complete characterization for the equilibrium points arising in the closed-loop system (2.14) with these CBF-based controllers, aiming at the development of a theory for effectively avoiding the convergence of the state trajectory towards undesirable equilibrium points.

This chapter is organized as follows:

1. Section 5.1 presents novel theoretical results regarding the existence and stability of all equilibrium points arising from the (i) safety-filter and (ii) CLF-CBF QP-based framework, as well as the concept of *CLF compatibility*, regarding the compatibility between the stabilization and safety objectives in the case of the CLF-CBF QP.

2. Section 5.2 proposes a controller using CLFs without rotational symmetry, aiming at avoiding the *existence* of boundary equilibrium points. Results are discussed, as well as the limitations of the proposed method.

3. Section 5.3 further develops the theory of CLF-compatibility for certain classes of dynamic systems and CLF-CBF pairs. We present a method for computing a CLF that is compatible with the safety-objectives from a non-compatible one, and simulation results for certain applications are presented and discussed.

## 5.1 Analysis of the CBF-based QP Framework

This section explores the existence and stability properties for equilibrium points arising in the QP-based framework for safe control using CBFs. We focus on the closed-loop system solutions

of the nonlinear affine control system (2.14) with the following generalized QP controller:

$$u^\star(x), \delta^\star(x) = \underset{(u,\delta)\in\mathbb{R}^{m+1}}{\operatorname{argmin}} \; (u-u_{\text{nom}}(x))^\mathsf{T} H(x)(u-u_{\text{nom}}(x)) + p\delta^2 \tag{5.1}$$

$$s.t. \; L_f V(x) + L_g V(x)\, u \leq -\alpha(V(x)) + \delta \tag{CLF}$$

$$L_f h_i(x) + L_g h_i(x)\, u \geq -\beta_i(h_i(x)), \quad i \in \{1, \cdots, N\} \tag{CBFs}$$

where $H : \mathbb{R}^n \to \mathbb{S}^m_{>0}$ is a symmetric positive definite matrix function of the system state, $p > 0$ and the $\alpha$ and $\beta_i$ are classes $\mathscr{K}$ and $\mathscr{K}_\infty$ functions, respectively. The structure of controller (5.1) allows for the generalization of the safety-filter QP and the CLF-CBF QP into a single framework:

1. with $V(x) = 0$ (without the CLF), the safety-filter QP controller is recovered. In this case, the optimal solution for the slack variable is always $\delta^\star = 0$. That means that the CLF constraint is always active.

2. with $u_{\text{nom}}(x) = 0$ (without the nominal stabilizing controller), the CLF-CBF QP controller is recovered.

This generalized optimization-based controller minimizes a general norm of the difference $\|u - u_{\text{nom}}(x)\|_{H(x)}$, seeking to achieve a weak form of stabilization by "softening" the CLF constraint (by introducing the slack variable $\delta$) and a hard form of safety by never relaxing the CBF constraints. This logic makes sense considering tasks where the safety requirements are non-negotiable. However, stability is seen as a secondary objective, since due to the slack variable, it is not possible to strictly guarantee that $u^\star(x) \in \mathbb{K}_{\text{clf}}(x) \cap \mathbb{K}_{\text{cbf}}(x)$. That means that, if the QP (5.1) is feasible, safety with respect to $\mathscr{C}$ is achieved, but stabilization is possibly hampered.

**Theorem 5.1.1.** The QP (5.1) is feasible for all $x \in \mathscr{C}$ in (2.38) if and only if

$$\sum_{i=1}^{N} \lambda_i L_g h_i(x) = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i \left(L_f h_i(x) + \beta_i(h_i(x))\right) \geq 0 \tag{5.2}$$

holds for all $x \in \mathbb{R}^n$ and $\lambda_1, \ldots, \lambda_N \geq 0$.

*Proof.* A proof for the case of the safety-filter can be found in [52], but can be easily adapted to the general case. Consider the Lagrangian of the feasibility problem corresponding to the QP (5.1):

$$\mathscr{L}_f(x, u, \delta, \bar{\lambda}) = \lambda_0(\dot{V}(x,u) + \alpha(V(x)) - \delta) - \sum_{i=1}^{N} \lambda_i(\dot{h}_i(x,u) + \beta_i(h_i(x)))$$

with KKT multipliers $\bar{\lambda} = \begin{bmatrix} \lambda_0 \lambda_1 \ldots \lambda_N \end{bmatrix}^\mathsf{T}$, $\lambda_i \geq 0 \forall i \in \{0, 1, \cdots, N\}$. The QP (5.1) is feasible if and only if $\exists (u, \delta) \in \mathbb{R}^{m+1}$ such that $\mathscr{L}_f(x, u, \delta, \bar{\lambda}) \leq 0, \forall x \in \mathbb{R}^n, \forall \lambda_i \geq 0$. With the Lagrange dual function defined as $\mathscr{D}(x, \bar{\lambda}) = \inf_{(u,\delta)\in\mathbb{R}^{m+1}} \mathscr{L}_f(x, u, \delta, \bar{\lambda})$, this condition means $\mathscr{D}(x, \bar{\lambda}) \leq 0, \forall x \in$

$\mathbb{R}^n, \forall \lambda_i \geq 0$. Notice that

$$\mathscr{D}(x,\bar{\lambda}) = \begin{cases} -\sum_{i=1}^{N} \lambda_i (L_f h_i(x) + \beta_i(h_i(x))) & \text{for } \lambda_0 = 0, \sum_{i=1}^{N} \lambda_i L_g h_i(x) = 0 \\ -\infty & \text{otherwise} \end{cases}$$

Then, (5.2) is equivalent to $\mathscr{D}(x,\bar{\lambda}) \leq 0$ and provides feasibility. □

The feasibility of the generalized QP (5.1) is guaranteed under the conditions of Theorem 5.1.1. If feasible, (5.1) generates a safe control, that is, $u^\star(x) \in \mathbb{K}_{\text{cbf}}(x)$, and possibly close to the stabilizing set $\mathbb{K}_{\text{clf}}(x)$ in some sense. If condition (5.2) is not satisfied, it is possible that some states exist such that two or more of the CBF constraints are incompatible, that is, $\exists x \in \mathbb{R}^n$ s.t. $\mathbb{K}_{\text{cbf}}(x) = \emptyset$ in (2.39). A weaker, but also important form of feasibility is *recursive feasibility*, a property that guarantees that if the QP is feasible at one time instant, it remains feasible for all future time $t \geq 0$.

**Definition 5.1.1** (Recursive Feasibility)**.** Given an initial condition $x(0) \in \mathbb{R}^n$, the QP (5.1) is said to recursively feasible if it is feasible at $x(0)$ and if

$$\text{feasible for } x(0) \in \mathbb{R}^n \Rightarrow \text{feasible for } x(t) \in \mathbb{R}^n \ \forall t \geq 0$$

Even though it is a weaker condition than full feasibility, recursive feasibility is important because it guarantees that the optimization-based controller will remain solvable if at least one initially feasible state exists.

**Assumption 8.** Considering the $N$ CBFs $h_i$ $(i = 1, \cdots, N)$, the initial state $x(t)\Big|_{t=0} \in \mathbb{R}^n$ is contained in the safe set $\mathscr{C}$ from (2.38), that is, $x(t)\Big|_{t=0} \in \mathscr{C}$.

Assumption 8 is natural, since it is not desired for the state trajectories to ever enter the unsafe region, and therefore it is somewhat natural to assume that they already start in a safe configuration. For example, in a safe multi-robot navigaton task, where a robotic vehicle is required to keep a safe distance from their neighbors and from obstacles in the environment, it is naturally assumed that all robots start in a non-collision configuration.

**Corollary 5.1.1.1.** Under Assumption 8, the QP (5.1) is recursively feasible if at least one of the following conditions hold:

1. Condition (5.2) holds for all $x \in \mathscr{C}$.
2. The non-linear system (2.13) is driftless at the safe set, that is, $f(x) = 0 \ \forall x \in \mathscr{C}$.

*Proof.* Under Assumption 8, $x(0) \in \mathscr{C}$.
**Proof of 1.** If (5.2) holds for all $x(t) \in \mathscr{C}, t \geq 0$, then it holds for the initial condition $x(0)$, which means the QP is initially feasible. Since the CBF constraints guarantee that $u^\star \in \mathbb{K}_{\text{cbf}}$, $x(t)$ is forward invariant with respect to $\mathscr{C}$. Therefore, the QP (5.1) is recursively feasible.

**Proof of 2.** If the non-linear system (2.13) is driftless at the safe set, then condition (5.2) becomes

$$\sum_{i=1}^{N} \lambda_i L_g h_i(x) = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i \beta_i(h_i(x)) \geq 0\,,$$

which automatically holds for all $x \in \mathscr{C}$ and $\lambda_1, \ldots, \lambda_N \geq 0$, since $\beta_i(h_i(x)) \geq 0$ for all $x \in \mathscr{C}$. Since the CBF constraints guarantee that $u^\star \in \mathbb{K}_{\text{cbf}}$, $x(t)$ is forward invariant with respect to $\mathscr{C}$. Therefore, the QP (5.1) is recursively feasible. □

As pointed out by the works [64, 79, 65, 69], neither the safety-filter QP (2.40) nor the CLF-CBF QP (2.41) can guarantee global stabilization of trajectories towards the origin for the closed-loop system (5.17), meaning that trajectories could converge towards undesirable equilibrium points and even towards other types of attractors, such as limits cycles.

Next, we investigate the conditions for existence and stability of equilibrium points formed by the closed-loop system (2.14) with $u^\star(x)$ given by the generalized QP-based controller (5.1), considering multiple CBF constraints. The results are conditioned to the feasibility of the QP (5.1): that is, it is assumed that the conditions of Theorem 5.1.1 or Corollary 5.1.1.1 holds. Particularly, as will be shown in the next two subsections, the next definition will be useful to unify the properties of existence and stability of all equilibrium points of the closed-loop system into a single mathematical object.

**Definition 5.1.2** (Equilibrium Field)**.** Let the set $\mathscr{A} = \{a_1, \cdots, a_r\} \subset 2^{\{1,\cdots,N\}}$ represent $0 \leq r \leq N$ CBF indexes. Then, define the *equilibrium vector field* as $f_a : \mathbb{R}^n \times \mathbb{R}^r_{\geq 0} \to \mathbb{R}^n$:

$$f_a(x, \lambda_a) = f_{\text{nom}}(x) + G(x)\left(-p\alpha(V(x))\nabla V(x) + \sum_{i \in \mathscr{A}} \lambda_i \nabla h_i(x)\right) \tag{5.3}$$

where $f_{\text{nom}}(x) = f(x) + g(x)u_{\text{nom}}(x)$, $G(x) = g(x)H(x)^{-1}g(x)^\mathsf{T}$, and $\lambda_a = \begin{bmatrix} \lambda_{a_1} & \cdots & \lambda_{a_r} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^r_{\geq 0}$. In particular, if $\mathscr{A}$ is empty, $r = 0$, the sum on (5.3) vanishes and $f_a$ is a pure function of the state $x \in \mathbb{R}^n$. The Jacobian matrix of (5.3) with respect to the state variable $x \in \mathbb{R}^n$ is given by

$$J_{f_a}(x, \lambda_a) = \frac{\partial f_{\text{nom}}}{\partial x} - p\alpha(V)\frac{\partial (G\nabla V)}{\partial x} - p\alpha'(V)G\nabla V\nabla V^\mathsf{T} + \sum_{i \in \mathscr{A}} \lambda_i \frac{\partial (G\nabla h_i)}{\partial x} \tag{5.4}$$

As will be demonstrated in the next sections, the vector field (5.3) and its Jacobian (5.4) will be of central importance to characterize the existence conditions and stability properties of all equilibrium points formed on the closed-loop system, respectively. Notice that (5.4) depends on the partial derivatives of the vector fields $f_{\text{nom}}(x)$ and $g(x)$ defining the system dynamics and on the CLF and CBF Hessian matrices $H_V(x)$ and $H_{h_i}(x)$, respectively. In particular, the Jacobian of $f_{\text{nom}}(x)$ also depends on the Jacobian of the nominal control $u_{\text{nom}}(x)$ from the safety-filter.

### 5.1.1 Existence of Equilibria on CBF QP Framework

In this section, we examine general conditions for existence of equilibrium points within the CBF QP framework.

**Definition 5.1.3** (Equilibrium Manifold)**.** Let (2.14) be the closed-loop system formed by the nonlinear system (2.13) and controller (5.1). Given a set of $0 \leq r \leq N$ CBF indexes $\mathscr{A} = \{a_1, \cdots, a_r\}$ as in Definition 5.1.2, define the following sets

$$\mathbb{A}_a = \left\{ x \in \mathbb{R}^n \,|\, \exists \lambda_a \in \mathbb{R}^r_{\geq 0} \text{ s.t. } f_a(x, \lambda_a) = 0 \right\} \tag{5.5}$$

$$\mathbb{A}_0 = \left\{ x \in \mathbb{R}^n \,|\, f_a(x, 0) = 0 \right\} \tag{5.6}$$

where $f_a$ is the equilibrium vector field defined in (5.3).

In Definition 5.1.3, (5.5) is associated to $\mathscr{A}$, that is, to a set of $r$ CBFs $\{h_{a_1}, \cdots, h_{a_r}\}$. Notice that, if $f_{\text{nom}}(x) = 0 \ \forall x \in \mathbb{R}^n$, (5.5) is simply the set of points in $\mathbb{R}^n$ for which the vectors $L_G V(x)$ and $L_G h_{a_i}(x)$ are co-directed. This is the case for driftless systems ($f(x) = 0$) with the CLF-CBF controller ($u_{\text{nom}}(x) = 0$). Furthermore, (5.6) is a subset of $\mathbb{R}^n$ in which $f_{\text{nom}}(x)$ and $L_G V(x)$ are co-directed. Also notice that $\mathbb{A}_0 \subset \mathbb{A}_a$, no matter the dimension of $\lambda$ in (5.5), by simply picking up $\lambda = 0 \in \mathbb{R}^r$ in (5.5). We can now state the following result.

**Theorem 5.1.2** (Existence of Equilibrium Points)**.** Let (2.14) be the closed-loop system formed by the nonlinear system (2.13) and controller (5.1). Let $\partial \mathscr{C}_{\mathscr{I}} = \bigcap_{i \in \mathscr{I}} \partial \mathscr{C}_i \neq \emptyset$ be the intersection of $1 \leq r \leq N$ CBF boundaries identified by a set of $r$ indexes $\mathscr{I} \subset 2^{\{1,\cdots,N\}}$. Then, the equilibrium points of (5.17) fall into two distinct types, namely the sets of *boundary* and *interior* equilibrium points, respectively:

$$\mathscr{E}_{\partial \mathscr{C}_{\mathscr{I}}} = \{ x \in \Omega^{clf}_{\mathscr{I}} \cap \partial \mathscr{C}_{\mathscr{I}} \,|\, f_a(x, \lambda_a(x)) = 0 \} \tag{5.7}$$

$$\mathscr{E}_{int(\mathscr{C})} = \{ x \in \Omega^{clf}_{cbf} \cap int(\mathscr{C}) \,|\, f_{\text{nom}}(x) = p\alpha(V) G \nabla V \} \tag{5.8}$$

where $\Omega^{clf}_{\mathscr{I}}, \Omega^{clf}_{cbf} \subset \mathbb{R}^n$ are sets of states where the CLF constraint is active: for the first one, only the $r$ CBF constraints with indexes $\mathscr{I}$ are simultaneously active, and for the second one, all CBF constraints are inactive. Equation (5.7) is the set of *boundary equilibrium points* occurring at the boundary intersection $\partial \mathscr{C}_{\mathscr{I}}$. Given any $x_e \in \mathscr{E}_{\partial \mathscr{C}_{\mathscr{I}}}$, define the maximum subset $\mathscr{A} \subset \mathscr{I}$ containing all $r_0 \leq r$ CBF indexes such that $L_g h_i(x_e) \neq 0, \ \forall i \in \mathscr{A} \subset \mathscr{I}$. Then, $x_e$ satisfies $f_a(x_e, \lambda) = 0$ for some $\lambda \in \mathbb{R}^{r_0}_{\geq 0}$, meaning that $\mathscr{E}_{\partial \mathscr{C}_{\mathscr{I}}} \subset \partial \mathscr{C}_{\mathscr{I}} \cap \mathbb{A}_a$, with $\mathbb{A}_a = \left\{ x \in \mathbb{R}^n \,|\, \exists \lambda_a \in \mathbb{R}^{r_0}_{\geq 0} \text{ s.t. } f_a(x, \lambda_a) = 0 \right\}$. Equation (5.8) is the set of *interior* equilibrium points, satisfying $\mathscr{E}_{int(\mathscr{C})} \subset int(\mathscr{C}) \cap \mathbb{A}_0$. The set containing all equilibrium points, including interior points and boundary points occurring at all possible boundary intersections is:

$$\mathscr{E} = \mathscr{E}_{int(\mathscr{C})} \cup \underbrace{\left( \bigcup_{\mathscr{I}}^{2^{\{1,\cdots,N\}}} \mathscr{E}_{\partial \mathscr{C}_{\mathscr{I}}} \right)}_{\mathscr{E}_{\partial \mathscr{C}}} \tag{5.9}$$

*Proof.* The Lagrangian associated to QP (2.37) is given by

$$\mathscr{L}(x,u,\delta,\bar{\lambda}) = \frac{1}{2}\|u-u_{\text{nom}}(x)\|^2_{H(x)} + \frac{1}{2}p\delta^2 + \lambda_0(L_fV(x)+L_gV(x)\,u+\alpha(V(x))-\delta) \\ -\sum_{i=1}^{N}\lambda_i(L_fh_i(x)+L_gh_i(x)\,u+\beta_i(h_i(x))) \tag{5.10}$$

where $\lambda_i \geq 0$ and $\lambda = \begin{bmatrix}\lambda_1 & \cdots & \lambda_N\end{bmatrix}^\mathsf{T} \in \mathbb{R}^N_{\geq 0}$, $\bar{\lambda} = \begin{bmatrix}\lambda_0 & \lambda^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{N+1}_{\geq 0}$ are vectors of KKT multipliers associated to the optimization problem. The stationarity KKT conditions are:

$$\frac{\partial \mathscr{L}}{\partial u} = H(x)\,(u-u_{\text{nom}}(x)) + \lambda_0\, g(x)^\mathsf{T}\nabla V(x) - \sum_{i=1}^{N}\lambda_i\, g(x)^\mathsf{T}\nabla h_i(x) = 0 \tag{5.11}$$

$$\frac{\partial \mathscr{L}}{\partial \delta} = p\delta - \lambda_0 = 0 \tag{5.12}$$

and the complementary slackness conditions are:

$$\lambda_0(L_fV(x)+\alpha(V(x))+L_gV(x)\,u-\delta) = 0 \tag{5.13}$$

$$\lambda_i(L_fh_i(x)+\beta_i(h_i(x))+L_gh_i(x)\,u) = 0 \quad i\in\{1,\cdots,N\} \tag{5.14}$$

Using (5.11)-(5.12), the QP solutions are given by:

$$u^\star(x) = u_{\text{nom}}(x) + H(x)^{-1}g(x)^\mathsf{T}\left(-\lambda_0\nabla V(x) + \sum_{i=1}^{N}\lambda_i\nabla h_i(x)\right) \tag{5.15}$$

$$\delta^\star(x) = p^{-1}\lambda_0\,, \tag{5.16}$$

with $\lambda_i \geq 0 \in \{0,1,\cdots,N\}$. Substituting (5.15) on (2.14) yields the following expression for the closed-loop system:

$$f_{cl}(x) = f_{\text{nom}}(x) + G(x)\left(-\lambda_0\nabla V(x) + \sum_{i=1}^{N}\lambda_i\nabla h_i(x)\right) \tag{5.17}$$

At an equilibrium point $x_e \in \mathscr{E}$, $f_{cl}(x_e) = 0$. Applying this condition to (5.17) yields

$$f_{\text{nom}}(x_e) = G(x_e)\left(\lambda_0\nabla V(x_e) - \sum_{i=1}^{N}\lambda_i\nabla h_i(x_e)\right) \tag{5.18}$$

Regarding the complementary slackness conditions (5.13)-(5.14), different cases must be considered, depending on the activation of the constraints. First, consider the two cases regarding the CLF constraint:

**Case 1.** Consider the region of the state space where the CLF constraint is *inactive*, $\Omega^{\overline{clf}} = \{x \in \mathbb{R}^n \mid L_{f_{cl}(x)}V(x) + \alpha(V(x)) - \delta^\star(x) < 0\}$. From (5.13), $\lambda_0 = 0$. Then, using (5.16), notice that $\delta^\star(x) = 0$. At an equilibrium point $x_e \in \mathscr{E} \cap \Omega^{\overline{clf}}$, $L_{f_{cl}(x_e)}V(x_e) = 0$, and therefore we obtain $\alpha(V(x_e)) < \delta^\star(x_e) = 0$, implying that $V(x_e) < 0$, which is a contradiction since $V$ is a positive

definite function and $\alpha$ is of class $\mathscr{K}$. Therefore, any equilibrium point must necessarily lie on the region where the CLF constraint is *active*.

**Case 2.** Consider the region where CLF constraint is *active*, $\Omega^{clf} = \{x \in \mathbb{R}^n \mid L_{f_{cl}(x)}V(x) + \alpha(V(x)) = \delta^\star(x)\}$. At an equilibrium point $x_e \in \mathscr{E} \cap \Omega^{clf}$, $L_{f_{cl}(x_e)}V(x_e) = 0$. Therefore, using (5.16), $\alpha(V(x_e)) = \delta^\star(x_e) = p^{-1}\lambda_0$. Then, at any equilibrium point $x_e \in \mathscr{E}$, the KKT multiplier associated to the CLF constraint is $\lambda_0(x_e) = p\alpha(V(x_e)) \geq 0$. Therefore, equation (5.18) yields:

$$f_{\text{nom}}(x_e) = G(x_e)\left(p\alpha(V(x_e))\nabla V(x_e) - \sum_{i=1}^{N}\lambda_i \nabla h_i(x_e)\right) \tag{5.19}$$

For the safety-filter, $V = 0$ and $\nabla V = 0$ are valid in (5.19). Next, we consider the cases regarding the activation of the CBF constraints. For the next two cases, the CLF constraint is assumed to be active.

**Case 3.** First, consider the region the CLF constraint is active, but all CBF constraints are *inactive*:

$$\Omega^{clf}_{cbf} = \Omega^{clf} \cap \{x \in \mathbb{R}^n \mid L_{f_{cl}(x)}h_i(x) + \beta_i(h_i(x)) > 0,\ i = 1, \cdots, N\}$$

From (5.14), $\lambda_1 = \cdots = \lambda_N = 0$. At an equilibrium point $x_e \in \mathscr{E} \cap \Omega^{clf}_{cbf}$, $L_{f_{cl}(x_e)}h_i(x_e) = 0$, implying that $h_i(x_e) > 0$. Therefore, equilibrium points occurring in this region must lie on the interior of the safe sets, that is, $x_e \in int(\mathscr{C}) = \bigcap_{i=1}^{N} int(\mathscr{C}_i)$. Additionally, (5.19) must be satisfied with $\lambda_1 = \cdots = \lambda_N = 0$, which means that $f_{\text{nom}}(x_e) = p\alpha(V(x_e))G(x_e)\nabla V(x_e)$, which is equivalent to $f_a(x_e, 0) = 0$. This shows that $x_e \in int(\mathscr{C}) \cap \mathbb{A}_0$.

**Case 4.** Consider the region where the CLF constraint is active and exactly $1 \leq r \leq N$ CBF constraints are simultaneously active. Their corresponding set of indexes is denoted by $\mathscr{I} \in 2^{\{1,\cdots,N\}}$. This region is expressed by

$$\Omega^{clf}_{\mathscr{I}} = \Omega^{clf} \cap \{x \in \mathbb{R}^n \mid L_{f_{cl}(x)}h_i(x) + \beta_i(h_i(x)) = 0,\ i \in \mathscr{I}\} \tag{5.20}$$

At an equilibrium point $x_e$ occurring in this region, $L_{f_{cl}(x_e)}h_i(x_e) = 0$, implying that $h_i(x_e) = 0$, $\forall i \in \mathscr{I}$. Therefore, $x_e$ must lie at the boundary intersection $\partial\mathscr{C}_{\mathscr{I}} = \bigcap_{i\in\mathscr{I}} \partial\mathscr{C}_i$ associated to these $r$ active CBFs. The conclusion is that boundary equilibrium points at $\partial\mathscr{C}_{\mathscr{I}}$ can only occur at the region where the CLF constraint is active and *only* the CBF constraints corresponding to the indexes $\mathscr{I}$ are simultaneously active, denoted here as $\Omega^{clf}_{\mathscr{I}} \subset \mathbb{R}^n$. Since in this case the remaining CBF constraints are all inactive, $\lambda_i = 0\ \forall i \notin \mathscr{I}$, and (5.19) reduces to

$$f_{\text{nom}}(x_e) = G(x_e)\left(p\alpha(V(x_e))\nabla V(x_e) - \sum_{i\in\mathscr{I}}\lambda_i \nabla h_i(x_e)\right) \tag{5.21}$$

Assume that there exist some indexes $i \in \mathscr{I}$ such that $L_g h_i(x_e) = 0$. In this case, in (5.21), $G(x_e)\nabla h_i(x_e) = 0$, and the corresponding KKT multiplier $\lambda_i$ does not appear in the equilibrium equation. Then, define the maximum subset $\mathscr{A} \subset \mathscr{I}$ containing $r_0 \leq r$ CBF indexes such that

$L_g h_i(x_e) \neq 0 \; \forall i \in \mathscr{A}$. Then, equation (5.21) becomes

$$f_{\text{nom}}(x_e) = G(x_e)\left(p\alpha(V(x_e))\nabla V(x_e) - \sum_{i\in\mathscr{A}} \lambda_i \nabla h_i(x_e)\right) \tag{5.22}$$

Notice that (5.22) is equivalent to $f_a(x,\lambda_a) = 0$ with $\lambda_a \in \mathbb{R}^{r_0}_{\geq 0}$ as defined in (5.3), which means that $x_e \in \mathbb{A}_a$. Thus, in this case, the equilibrium point is at the boundary intersection $\partial\mathscr{C}_{\mathscr{I}}$ and satisfies $f_a(x_e,\lambda_a) = 0$ for some $\lambda_a \in \mathbb{R}^{r_0}_{\geq 0}$. This shows that $x_e \in \partial\mathscr{C}_{\mathscr{I}} \cap \mathbb{A}_a$.

Since all possible cases were considered, the complete set of equilibrium points is described by the union $\mathscr{E} = \mathscr{E}_{\partial\mathscr{C}} \cup \mathscr{E}_{int(\mathscr{C})}$, where $\mathscr{E}_{\partial\mathscr{C}} = \bigcup_{\mathscr{I}}^{2^{\{1,\cdots,N\}}} \mathscr{E}_{\partial\mathscr{C}_{\mathscr{I}}}$ is the set of boundary equilibria occurring at all possible intersections of CBF boundaries, thus proving (5.9). □

As claimed by the end of the previous subsection, the vector field (5.3) is the central mathematical object for determining the existence of equilibrium points. Summarizing, all equilibrium points $x_e \in \mathscr{E}$ satisfy $f_a(x_e,\lambda) = 0$ for some vector $\lambda$ with non-negative components:

(i) for *boundary* equilibrium points $x_e \in \mathscr{E}_{\partial\mathscr{C}_{\mathscr{I}}}$ occurring at the intersection of $r$ CBF boundaries identified by the set of indexes $\mathscr{I} = \{a_1,\cdots,a_r\}$:

- if $L_g h_i(x_e) \neq 0 \; \forall i \in \mathscr{I}$, $f_a(x_e,\lambda) = 0$ with $\lambda \in \mathbb{R}^r_{\geq 0}$.
- if $L_g h_i(x_e) = 0$ for some (or all) of the CBF indexes in $\mathscr{I}$, define $\mathscr{A} \subset \mathscr{I}$ with $r_0 \leq r$ CBF indexes such that $L_g h_i(x_e) \neq 0 \; \forall i \in \mathscr{A}$. Then, $f_a(x_e,\lambda) = 0$ with $\lambda \in \mathbb{R}^{r_0}_{\geq 0}$. In particular, if $\mathscr{A} = \emptyset$, $L_g h_i(x_e) = 0 \; \forall i \in \mathscr{I}$ and $x_e$ satisfies $f_a(x_e,0) = 0$.

(ii) for *interior* equilibrium points $x_e \in \mathscr{E}_{int(\mathscr{C})}$, $f_a(x_e,0) = 0$.

A discussion on the regions where each type of equilibrium points occur can be found in [79], for the case of only one CBF. In particular, with the generalized controller (5.1), considering:

- the safety filter QP at (2.36) with multiple CBFs, $V(x) = 0$ and the equilibrium condition at (5.22) reduces to

$$f(x_e) + g(x_e)u_{\text{nom}}(x_e) + \sum_{i\in\mathscr{A}} \lambda_i G(x_e)\nabla h_i(x_e) = 0 \tag{5.23}$$

- the CLF-CBF QP at (2.37) with multiple CBFs, $u_{\text{nom}}(x) = 0$ and the equilibrium condition at (5.22) reduces to

$$f(x_e) - p\alpha(V(x_e))G(x_e)\nabla V(x_e) + \sum_{i\in\mathscr{A}} \lambda_i G(x_e)\nabla h_i(x_e) = 0 \tag{5.24}$$

Either way, safety-critical control literature shows that the both types of QP-based controllers introduce undesirable equilibrium points in the closed-loop system satisfying equations (5.23) or (5.24), for the safety filter QP or the CLF-CBF QP, respectively.

### 5.1.2 Stability of Equilibrium Points on the CBF QP Framework

In this section, we present expressions for the closed-loop system Jacobian computed at the different types of equilibrium points $x_e \in \mathscr{E}$ (boundary and interior). These derivations will be essential steps towards the main result of this section: a general result relating the stability of the closed-loop equilibrium points to the properties of the Jacobian defined at (5.4).

**Definition 5.1.4.** Define the matrix-valued transformation $J_a : \mathbb{R}^n \times \mathbb{R}^r_{\geq 0} \to \mathbb{R}^{n\times n}$ associated to the set $\mathscr{A}$ of $r$ CBF indexes as

$$J_a(x,\lambda_a) = \frac{\partial f_{\text{nom}}}{\partial x} - p\alpha(V)\frac{\partial G\nabla V}{\partial x} + \sum_{i\in\mathscr{A}} \lambda_i \frac{\partial G\nabla h_i}{\partial x} \tag{5.25}$$

with $\lambda_a = \begin{bmatrix} \lambda_{a_1} & \cdots & \lambda_{a_r} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^r_{\geq 0}$.

**Remark.** Notice that matrix (5.25) is *not the same* as the Jacobian (5.4) from Definition 5.1.2. The two matrices differ precisely by a factor of $-p\alpha'(V)G\nabla V\nabla V^\mathsf{T}$.

The matrix function $J_a$ in (5.25) will appear in the structure of the closed-loop Jacobian matrices at interior and boundary equilibrium points, as illustrated in the next two subsections.

#### 5.1.2.1 Closed-Loop Jacobian at Interior Equilibria

Next, an expression for the closed-loop Jacobian matrix at an interior equilibrium point $x_e \in \mathscr{E}_{int(\mathscr{C})}$ is derived.

**Proposition 5.1.1** (Interior Closed-Loop Jacobian)**.** At an *interior* equilibrium point $x_e \in \mathscr{E}_{int(\mathscr{C})}$ or at a *boundary* equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ occurring at the intersection of $r$ CBF boundaries $\partial\mathscr{C}_a = \cap_{i\in\mathscr{A}}\partial\mathscr{C}_i$ with set of indexes given by $\mathscr{A}$ and $L_g h_i(x_e) = 0\ \forall i \in \mathscr{A}$, the closed-loop Jacobian matrix of (2.14) with controller (5.1) is given by

$$J_{f_{cl}}(x_e) = \mathsf{P}_V J_a(x_e,0) - \frac{1}{c}\alpha'(V)G\nabla V\nabla V^\mathsf{T} \tag{5.26}$$

where $\mathsf{P}_V(x) = I_n - \frac{1}{c}G\nabla V\nabla V^\mathsf{T}$ with $c = p^{-1} + \|\nabla V\|^2_G > 0$.

*Proof.* This demonstration follows from the proof of Theorem 5.1.2: **Case 3** deals with the region where the CLF constraint is active and all CBF constraints are inactive (where *interior* equilibrium points lie), and **Case 4** deals with the region where the CLF constraint and $r$ CBF constraints are simultaneously active (where *boundary* equilibrium points lie). Either in **Case 3** or **Case 4** with $L_g h_i(x_e) = 0\ \forall i \in \mathscr{A}$, we have $\lambda_i = 0\,(i = 1,\cdots,N)$ in (5.15) and an expression for $\lambda_0 > 0$ can be found by combining (5.15) (with all $\lambda_i = 0$) and (5.16) with the complementary slackness condition (5.13), yielding $c\lambda_0 = F_V$, where $c = p^{-1} + \|\nabla V\|^2_G$. Taking its derivative with respect to $x_k$ yields

$$(\partial_k c)\lambda_0 + c\partial_k\lambda_0 = (\partial_k\nabla V)^\mathsf{T} f_{\text{nom}} + \nabla V^\mathsf{T}\partial_k f_{\text{nom}} + \alpha'(V)\partial_k V \tag{5.27}$$

Since $\partial_k c = (\partial_k \nabla V)^\mathsf{T} G \nabla V + \nabla V^\mathsf{T} \partial_k (G \nabla V)$, replacing it at (5.27) yields

$$\partial_k \lambda_0(x) = \frac{1}{c} \left( (\partial_k \nabla V)^\mathsf{T} (\underbrace{f_{\text{nom}} - \lambda_0 G \nabla V}_{f_{cl}(x)}) + \nabla V^\mathsf{T} (\partial_k f - \lambda_0 \partial_k (G \nabla V)) + \alpha'(V) \partial_k V \right) \tag{5.28}$$

At the equilibrium point $x_e$, $f_{cl}(x_e) = 0$ and $\lambda_0 = p\alpha(V(x_e))$. Then, using (5.25), (5.28) becomes

$$\partial_k \lambda_0(x_e) = \frac{1}{c} \left( \nabla V^\mathsf{T} [J_a(x_e, 0)]_k + \alpha'(V) [\nabla V]_k \right) \tag{5.29}$$

Equation (5.17) with $\lambda_i = 0, i = 1, \cdots, N$ gives the dynamics of the closed-loop system when the CLF is active and all CBFs are inactive: $f_{cl}(x) = f_{\text{nom}} - \lambda_0 G \nabla V$. Differentiating it with respect to $x_k$ yields

$$\partial_k f_{cl}(x) = \partial_k f_{\text{nom}} - \lambda_0 \partial_k (G \nabla V) - G \nabla V \partial_k \lambda_0 \tag{5.30}$$

At the equilibrium point $x_e$ considered here, $\lambda_0 = p\gamma(V(x_e))$. Then, substituting (5.29) in (5.30) yields

$$\begin{aligned} \partial_k f_{cl}(x_e) &= [J_a(x_e, 0)]_k - \frac{1}{c} G \nabla V \left( \nabla V^\mathsf{T} [J_a(x_e, 0)]_k + \alpha'(V) [\nabla V]_k \right) \\ &= \mathsf{P}_V [J_a(x_e, 0)]_k - \frac{1}{c} \alpha'(V) G \nabla V [\nabla V]_k \end{aligned} \tag{5.31}$$

which is simply the $k$-th column of the closed-loop Jacobian matrix $J_{f_{cl}}(x_e)$ in (5.26). □

#### 5.1.2.2 Closed-Loop Jacobian at Boundary Equilibria

Next, expressions for the closed-loop Jacobian matrix at a boundary equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}}$ are derived, under different assumptions.

**Definition 5.1.5** (Orthogonal Basis)**.** Let the set $\mathscr{A} = \{a_1, \cdots, a_r\} \subset 2^{\{1,\cdots,N\}}$ as in Definition 5.1.3 represent $1 \leq r \leq N$ CBF indexes such that the gradients $\{\nabla h_{a_1}, \cdots, \nabla h_{a_r}\}$ are all linearly independent and $L_g h_i \neq 0 \; \forall i \in \mathscr{A}$. Then, matrix $U_a = \begin{bmatrix} \nabla h_{a_1} & \cdots & \nabla h_{a_r} \end{bmatrix} \in \mathbb{R}^{n \times r}$ has full column rank. Define the sequence of $r$ vectors $z_1, \cdots, z_r$ as:

$$\begin{aligned}
z_1(x) &= \nabla h_{a_1}, & \hat{z}_1 &= \frac{z_1}{\|z_1\|_G}, \\
z_2(x) &= \nabla h_{a_2} - \langle \nabla h_{a_2}, \hat{z}_1 \rangle_G \hat{z}_1, & \hat{z}_2 &= \frac{z_2}{\|z_2\|_G}, \\
z_3(x) &= \nabla h_{a_3} - \langle \nabla h_{a_3}, \hat{z}_1 \rangle_G \hat{z}_1 - \langle \nabla h_{a_3}, \hat{z}_2 \rangle_G \hat{z}_2, & \hat{z}_3 &= \frac{z_3}{\|z_3\|_G}, \\
&\vdots & & \\
z_r(x) &= \nabla h_{a_r} - \sum_{i=1}^{r-1} \langle \nabla h_{a_r}, \hat{z}_i \rangle_G \hat{z}_i, & \hat{z}_r &= \frac{z_r}{\|z_r\|_G}
\end{aligned}$$

This sequence consists of an $r$-step Gram-Schmidt orthogonalization process for the CBF gradients $\{\nabla h_{a_1}, \cdots, \nabla h_{a_r}\}$, with the inner product $\langle \cdot, \cdot \rangle_G$. With respect to $\langle \cdot, \cdot \rangle_G$, $\{\hat{z}_1, \cdots, \hat{z}_r\}$ is an orthonormal set of vectors, that is, $\langle \hat{z}_i, \hat{z}_j \rangle_G = \delta_{ij}$, $i, j = 1, \cdots, r$, where $\delta_{ij}$ is the Dirac delta. Then, matrix $Q = \begin{bmatrix} \hat{z}_1 & \cdots & \hat{z}_r \end{bmatrix} \in \mathbb{R}^{n \times r}$ is unitary with respect to $\langle \cdot, \cdot \rangle_G$, that is, $Q^\mathsf{T} G Q = I_r$. Then, matrices $U_a$ and $Q$ are related by the QR decomposition $U_a = QR$:

$$\underbrace{\begin{bmatrix} \nabla h_{a_1} & \cdots & \nabla h_{a_r} \end{bmatrix}}_{U_a} = \underbrace{\begin{bmatrix} \hat{z}_1 & \cdots & \hat{z}_r \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \langle \nabla h_{a_1}, \hat{z}_1 \rangle_G & \langle \nabla h_{a_2}, \hat{z}_1 \rangle_G & \cdots & \langle \nabla h_{a_r}, \hat{z}_1 \rangle_G \\ 0 & \langle \nabla h_{a_2}, \hat{z}_2 \rangle_G & \cdots & \langle \nabla h_{a_r}, \hat{z}_2 \rangle_G \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \langle \nabla h_{a_r}, \hat{z}_r \rangle_G \end{bmatrix}}_{R}, \tag{5.32}$$

where the upper triangular matrix $R \in \mathbb{R}^{r \times r}$ is always invertible due to the fact that $U_a$ is full column rank. Define also the following vector orthogonal to $\{\hat{z}_1, \cdots, \hat{z}_r\}$ with respect to $\langle \cdot, \cdot \rangle_G$:

$$z_v(x) = \nabla V - \sum_{i=1}^{r} \langle \nabla V, \hat{z}_i \rangle_G \hat{z}_i = \nabla V - QQ^\mathsf{T} G \nabla V = \mathsf{P}_Q^\mathsf{T} \nabla V \tag{5.33}$$

where $\mathsf{P}_Q = I_n - GQQ^\mathsf{T}$ is a projection matrix, with the property $Q^\mathsf{T} G z_v = 0$. Furthermore, define the scalar function $\eta : \mathbb{R}^n \to \mathbb{R}_+$ as

$$\eta(x) = \frac{1}{1 + p\|z_v\|_G^2} \tag{5.34}$$

The vector function (5.33) and the scalar function (5.34) have the following properties:

i $0 < \eta(x) \leq 1 \; \forall x \in \mathbb{R}^n$.

ii $\eta(x) = 1$ if and only if $Gz_v = 0$ or $z_v = 0$, which occurs at boundary equilibrium points $x_e$ such that: (i) the gradients $\nabla V(x_e)$, $\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)$ are collinear, that is $\nabla V(x_e) = U_a(x_e)\lambda$ for some $\lambda \in \mathbb{R}^r$, or (ii) $\nabla V(x_e) = 0$.

iii $c = \eta^{-1} p^{-1} + \nabla V^\mathsf{T} GQQ^\mathsf{T} G \nabla V$.

The QR decomposition of matrix $U_a$ in (5.32) and (5.33)-(5.34) from Definition 5.1.5 will be useful for simplifying the derivations on the remainder of this section.

**Proposition 5.1.2** (Boundary Closed-Loop Jacobian)**.** Let $\mathscr{A} = \{a_1, \cdots, a_r\}$ denote the set of indexes of $1 \leq r \leq n$ CBFs whose boundary intersection is $\partial\mathscr{C}_a = \cap_{i \in \mathscr{A}} \partial\mathscr{C}_i$. At a *boundary* equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$, assume that $L_g h_i(x_e) \neq 0 \; \forall i \in \mathscr{A}$ and that all $r$ CBF gradients $\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)$ are linearly independent. Then, the closed-loop Jacobian matrix of (2.14) with controller (5.1) is given by

$$J_{f_{cl}}(x_e) = \left(I_n - GZN_1 Z^\mathsf{T}\right) J_a(x_e, \lambda_a) - GZN_1 \Psi Z^\mathsf{T} \tag{5.35}$$

where $J_a$ is given by (5.25) for the KKT multipliers $\lambda_a \in \mathbb{R}^r_{\geq 0}$ at $x_e$, $Z(x_e) = \begin{bmatrix} Q(x_e) & z_v(x_e) \end{bmatrix} \in \mathbb{R}^{n\times(r+1)}$ with $Q$ and $z_v$ as in Definition 5.1.5 and

$$N_1(x) = \begin{bmatrix} I_r & 0 \\ 0 & \eta p \end{bmatrix} > 0, \quad \Psi(x) = \begin{bmatrix} (R\Lambda' R^{-1})^\mathsf{T} & 0 \\ \Delta & \alpha'(V) \end{bmatrix} > 0 \tag{5.36}$$

with $\eta$ as defined in (5.34), $\Lambda' = \text{diag}\{\beta'_{a_1}(0), \cdots, \beta'_{a_r}(0)\} > 0$, $\Delta = \nabla V^\mathsf{T} GQ\left(\alpha'(V) I_r - R\Lambda' R^{-1}\right)^\mathsf{T}$ and $R$ being the upper triangular matrix from the QR decomposition of the full column rank matrix $U_a(x_e) \in \mathbb{R}^{n\times r}$, whose columns are the $r$ CBF gradients $\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)$.

*Proof.* This demonstration follows from the proof of Theorem 5.1.2. First, we seek to find a general expression for the KKT multipliers at **Case 3** of the proof of Theorem 5.1.2. Combining (5.15) with $\lambda_i = 0\ \forall i \notin \mathscr{A}$ and (5.16) with the complementary slackness conditions (5.13)-(5.14) yields the following system:

$$\underbrace{\begin{bmatrix} c & -\langle \nabla V, \nabla h_{a_1}\rangle_G & \cdots & -\langle \nabla V, \nabla h_{a_r}\rangle_G \\ -\langle \nabla V, \nabla h_{a_1}\rangle_G & \langle \nabla h_{a_1}, \nabla h_{a_1}\rangle_G & \cdots & \langle \nabla h_{a_1}, \nabla h_{a_r}\rangle_G \\ \vdots & \vdots & \ddots & \vdots \\ -\langle \nabla V, \nabla h_{a_r}\rangle_G & \langle \nabla h_{a_1}, \nabla h_{a_r}\rangle_G & \cdots & \langle \nabla h_{a_r}, \nabla h_{a_r}\rangle_G \end{bmatrix}}_{A_a(x) = A_a(x)^\mathsf{T}} \underbrace{\begin{bmatrix} \lambda_0 \\ \lambda_{a_1} \\ \vdots \\ \lambda_{a_r} \end{bmatrix}}_{\tilde{\lambda}_a} = \underbrace{\begin{bmatrix} F_V \\ -F_{h_{a_1}} \\ \vdots \\ -F_{h_{a_r}} \end{bmatrix}}_{b_a(x)} \tag{5.37}$$

where $c = p^{-1} + \|\nabla V\|_G^2 > 0$, $F_V(x) = \nabla V^\mathsf{T} f_{\text{nom}} + \alpha(V)$ and $F_{h_{a_i}}(x) = \nabla h_{a_i}^\mathsf{T} f_{\text{nom}} + \beta_{a_i}(h_{a_i})$. By defining matrix $U_a(x) = \begin{bmatrix} \nabla h_{a_1}(x) & \cdots & \nabla h_{a_r}(x) \end{bmatrix} \in \mathbb{R}^{n\times r}$, (5.37) can be rewritten compactly as

$$\underbrace{\begin{bmatrix} c & -\nabla V^\mathsf{T} G U_a \\ -U_a^\mathsf{T} G \nabla V & U_a^\mathsf{T} G U_a \end{bmatrix}}_{A_a(x)} \underbrace{\begin{bmatrix} \lambda_0 \\ \lambda_a \end{bmatrix}}_{\tilde{\lambda}} = \underbrace{\begin{bmatrix} F_V \\ -F_{h_a} \end{bmatrix}}_{b_a(x)}, \tag{5.38}$$

$$F_{h_a}(x) = U_a(x)^\mathsf{T} f_{\text{nom}}(x) + \bar{\beta}_a \in \mathbb{R}^r$$

where $\bar{\beta}_a(x) = \begin{bmatrix} \beta_{a_1}(h_{a_1}(x)) & \cdots & \beta_{a_r}(h_{a_r}(x)) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^r$. In particular, the set where the CLF constraint and *only* the $r$ CBF constraints corresponding to $\{h_{a_1}, \cdots, h_{a_r}\}$ are active - previously identified in (5.20) - is here rewritten as

$$\Omega_{\mathscr{A}}^{clf} = \{x \in \mathbb{R}^n \,|\, f_{cl}(x) = f_{\text{nom}}(x) - \lambda_0 G(x)\nabla V(x) + G(x)U_a(x)\lambda_a\} \tag{5.39}$$

with $\lambda_0 \geq 0, \lambda_a \in \mathbb{R}^r_{\geq 0}$ drawn from the solutions of (5.38). Using the known formula for the determinant of block matrices, the determinant of $A_a(x)$ in (5.38) is

$$\det A_a(x) = c \det S_{A_a} \geq 0 \tag{5.40}$$

where $S_{A_a}$ is the Schur complement of $A_a$, given by $S_{A_a} = U_a^\mathsf{T} \mathsf{P}_V G U_a \in \mathbb{R}^{r\times r}$, with

$$\mathsf{P}_V(x) = I_n - \frac{1}{c} G \nabla V \nabla V^\mathsf{T} \tag{5.41}$$

Besides the similarity, $\mathsf{P}_V$ at (5.41) is not a projection matrix, being positive definite. Since $c > 0$, $A_a(x_e)$ loses rank if and only if $S_{A_a}(x_e)$ loses rank. At a boundary equilibrium point $x_e$, $S_{A_a}(x_e)$ loses rank if and only if: (i) $L_g h_i(x_e) = 0$ for some $i \in \mathscr{A}$, (ii) if $U_a$ is not full column rank or (iii) $r > n$. Therefore, by the assumptions of the theorem, $\det A_a(x_e) \neq 0$ and thus $A_a(x_e)$ is invertible.

The derivative of (5.38) with respect to the $k$-th state $x_k$ yields

$$\partial_k A_a(x)\bar{\lambda}(x) + A_a(x)\partial_k \bar{\lambda}(x) = \partial_k b_a(x), \tag{5.42}$$

where $\bar{\lambda} = \begin{bmatrix} \lambda_0 & \lambda_a^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{r+1}_{\geq 0}$. Defining $\bar{U} = \begin{bmatrix} -\nabla V & U_a \end{bmatrix} \in \mathbb{R}^{n\times(r+1)}$, the partial derivatives of $A_a$ and $b_a$ are:

$$\partial_k A_a(x) = \partial_k(\bar{U}^\mathsf{T} G \bar{U}) \tag{5.43}$$

$$\partial_k b_a(x) = -\partial_k(\bar{U}^\mathsf{T} f_{\text{nom}}) - \bar{\Lambda}' \underbrace{\begin{bmatrix} -\partial_k V & \partial_k h_{a_1} & \cdots & \partial_k h_{a_r} \end{bmatrix}^\mathsf{T}}_{[\bar{U}^\mathsf{T}]_k} \tag{5.44}$$

where $\bar{\Lambda}'(x) = \text{diag}\{\alpha'(V), \beta'_{a_1}(h_{a_1}), \cdots, \beta'_{a_r}(h_{a_r})\} \in \mathbb{S}^{r+1}_{>0}$, since $\alpha$ and the $\beta_i$ are of class $\mathscr{K}$ and extended class $\mathscr{K}$, respectively. Under the assumption that $A_a$ is invertible, which holds at $x_e$ by the assumptions of the theorem, (5.42) can be solved for $\partial_k \bar{\lambda}(x)$, yielding

$$\partial_k \bar{\lambda}(x) = A_a(x)^{-1}\left(\partial_k b_a(x) - \partial_k A_a(x)\bar{\lambda}(x)\right) \tag{5.45}$$

The closed-loop system expression (5.17) with $\lambda_j = 0 \; \forall j \notin \mathscr{A}$ holds when the CLF and only the CBF constraints from $\mathscr{A}$ are active, and can be rewritten using $\bar{U}$ and $\bar{\lambda} = \begin{bmatrix} \lambda_0 & \lambda_a^\mathsf{T} \end{bmatrix}^\mathsf{T}$ as

$$f_{cl}(x) = f_{\text{nom}} + G\bar{U}\bar{\lambda} \tag{5.46}$$

Differentiating (5.46) with respect to the $k$-th state variable $x_k$ yields

$$\partial_k f_{cl}(x) = \partial_k f_{\text{nom}} + \partial_k(G\bar{U})\bar{\lambda} + G\bar{U}\partial_k\bar{\lambda} \tag{5.47}$$

To further derive (5.47), it is necessary to further derive $\partial_k \bar{\lambda}$ in (5.45). Substituting (5.43)-(5.44) into (5.45) yields

$$\begin{aligned} \partial_k \bar{\lambda}(x) &= -A_a^{-1}\left(\partial_k(\bar{U}^\mathsf{T} f_{\text{nom}}) + \partial_k(\bar{U}^\mathsf{T} G\bar{U})\bar{\lambda} + \bar{\Lambda}'[\bar{U}^\mathsf{T}]_k\right) \\ &= -A_a^{-1}\left((\partial_k \bar{U})^\mathsf{T} \underbrace{\left(f_{\text{nom}} + G\bar{U}\bar{\lambda}\right)}_{f_{cl}(x)} + \bar{U}^\mathsf{T}(\partial_k f_{\text{nom}} + \partial_k(G\bar{U})\bar{\lambda}) + \bar{\Lambda}'[\bar{U}^\mathsf{T}]_k\right) \end{aligned} \tag{5.48}$$

where the closed-loop expression (5.46) was used. Then, substituting (5.48) in (5.47) yields

$$\partial_k f_{cl}(x) = \left(I_n - G\bar{U}A_a^{-1}\bar{U}^\mathsf{T}\right)\left(\partial_k f_{\text{nom}} + \partial_k(G\bar{U})\bar{\lambda}\right) - G\bar{U}A_a^{-1}\left(\bar{\Lambda}'[\bar{U}^\mathsf{T}]_k - (\partial_k\bar{U})^\mathsf{T} f_{cl}(x)\right) \tag{5.49}$$

which is an expression for the $k$-th column of the closed-loop Jacobian $J_{f_{cl}}(x)$. At a boundary equilibrium point $x_e \in \partial\mathscr{C}_a$, $f_{cl}(x_e) = 0$ and $\lambda_0(x_e) = p\alpha(V(x_e))$. Therefore, the last term at (5.49) vanishes and $\partial_k f_{\text{nom}} + \partial_k(G\bar{U})\bar{\lambda}$ becomes the $k$-th column of matrix $J_a(x_e, \lambda_a(x_e))$ as defined at (5.25). Therefore, applying the equilibrium conditions and stacking the $n$ columns $\partial_k f_{cl}(x_e)$, $k = 1, \cdots, n$ yields:

$$J_{f_{cl}}(x_e) = (I - G\bar{U}A_a^{-1}\bar{U}^\mathsf{T})J_a(x_e, \lambda_a(x_e)) - G\bar{U}A_a^{-1}\bar{\Lambda}'\bar{U}^\mathsf{T} \tag{5.50}$$

which is already a valid expression for the closed-loop Jacobian at $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$. To further simplify (5.50), first notice that a formula for $A_a^{-1}$ can be obtained by performing Gauss elimination on the block matrix (5.38):

$$\begin{aligned} A_a^{-1}(x) &= \begin{bmatrix} c^{-1} + c^{-2}\nabla V^\mathsf{T} G U_a S_{A_a}^{-1} U_a^\mathsf{T} G \nabla V & c^{-1}\nabla V^\mathsf{T} G U_a S_{A_a}^{-1} \\ c^{-1} S_{A_a}^{-1} U_a^\mathsf{T} G \nabla V & S_{A_a}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} c^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{c^2}\begin{bmatrix} \nabla V^\mathsf{T} G U_a \\ c I_r \end{bmatrix} S_{A_a}^{-1} \begin{bmatrix} \nabla V^\mathsf{T} G U_a \\ c I_r \end{bmatrix}^\mathsf{T} \end{aligned} \tag{5.51}$$

Then, combining (5.51) and the definition of $\bar{U}$ yields

$$\begin{aligned} \bar{U}A_a^{-1} &= \begin{bmatrix} -\nabla V & U_a \end{bmatrix}\left(\begin{bmatrix} c^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{c^2}\begin{bmatrix} \nabla V^\mathsf{T} G U_a \\ c I_r \end{bmatrix} S_{A_a}^{-1} \begin{bmatrix} \nabla V^\mathsf{T} G U_a \\ c I_r \end{bmatrix}^\mathsf{T}\right) \\ &= \begin{bmatrix} -c^{-1}\nabla V & 0 \end{bmatrix} + \frac{1}{c^2}\underbrace{\left(c I_n - \nabla V \nabla V^\mathsf{T} G\right)}_{c\mathsf{P}_V^\mathsf{T}} U_a S_{A_a}^{-1} \begin{bmatrix} \nabla V^\mathsf{T} G U_a \\ c I_r \end{bmatrix}^\mathsf{T} \\ &= \begin{bmatrix} -c^{-1}\nabla V & 0 \end{bmatrix} + c^{-1}\mathsf{P}_V^\mathsf{T} U_a S_{A_a}^{-1}\begin{bmatrix} U_a^\mathsf{T} G \nabla V & c I_r \end{bmatrix} \end{aligned} \tag{5.52}$$

$$\begin{aligned} \bar{U}A_a^{-1}\bar{\Lambda}' &= \left(\begin{bmatrix} -c^{-1}\nabla V & 0 \end{bmatrix} + c^{-1}\mathsf{P}_V^\mathsf{T} U_a S_{A_a}^{-1}\begin{bmatrix} U_a^\mathsf{T} G \nabla V & c I_r \end{bmatrix}\right)\begin{bmatrix} \alpha'(V) & 0 \\ 0 & \Lambda' \end{bmatrix} \\ &= \begin{bmatrix} -c^{-1}\alpha'(V)\nabla V & 0 \end{bmatrix} + c^{-1}\mathsf{P}_V^\mathsf{T} U_a S_{A_a}^{-1}\begin{bmatrix} \alpha'(V) U_a^\mathsf{T} G \nabla V & c\Lambda' \end{bmatrix} \end{aligned} \tag{5.53}$$

Using (5.52)-(5.53), we can write:

$$\begin{aligned}
I_n - G\bar{U}A_a^{-1}\bar{U}^\mathsf{T} &= I_n - G\left(\begin{bmatrix} -c^{-1}\nabla V & 0 \end{bmatrix} + c^{-1}\mathsf{P}_V^\mathsf{T} U_a S_{A_a}^{-1}\begin{bmatrix} U_a^\mathsf{T} G\nabla V & cI_r \end{bmatrix}\right)\begin{bmatrix} -\nabla V^\mathsf{T} \\ U_a^\mathsf{T} \end{bmatrix} \\
&= I_n - c^{-1}G\nabla V\nabla V^\mathsf{T} - c^{-1}G\mathsf{P}_V^\mathsf{T} U_a S_{A_a}^{-1}U_a^\mathsf{T}\underbrace{\left(cI_r - G\nabla V\nabla V^\mathsf{T}\right)}_{c\mathsf{P}_V} \\
&= \mathsf{P}_V - \mathsf{P}_V G U_a S_{A_a}^{-1}U_a^\mathsf{T}\mathsf{P}_V = \underbrace{\left(I_n - \mathsf{P}_V G U_a S_{A_a}^{-1}U_a^\mathsf{T}\right)}_{\mathsf{P}_{U_a}}\mathsf{P}_V = \mathsf{P}_{U_a}\mathsf{P}_V
\end{aligned} \tag{5.54}$$

$$\begin{aligned}
G\bar{U}A_a^{-1}\bar{\Lambda}'\bar{U}^\mathsf{T} &= G\left(\begin{bmatrix} -c^{-1}\alpha'(V)\nabla V & 0 \end{bmatrix} + c^{-1}\mathsf{P}_V^\mathsf{T} U_a S_{A_a}^{-1}\begin{bmatrix} \alpha'(V)U_a^\mathsf{T} G\nabla V & c\Lambda' \end{bmatrix}\right)\begin{bmatrix} -\nabla V^\mathsf{T} \\ U_a^\mathsf{T} \end{bmatrix} \\
&= c^{-1}\alpha'(V)G\nabla V\nabla V^\mathsf{T} - c^{-1}\alpha'(V)\mathsf{P}_V G U_a S_{A_a}^{-1}U_a^\mathsf{T} G\nabla V\nabla V^\mathsf{T} \\
&\qquad + \mathsf{P}_V G U_a S_{A_a}^{-1}\Lambda' U_a^\mathsf{T} \\
&= c^{-1}\alpha'(V)\underbrace{\left(I_n - \mathsf{P}_V G U_a S_{A_a}^{-1}U_a^\mathsf{T}\right)}_{\mathsf{P}_{U_a}} G\nabla V\nabla V^\mathsf{T} + \mathsf{P}_V G U_a S_{A_a}^{-1}\Lambda' U_a^\mathsf{T} \\
&= c^{-1}\alpha'(V)\mathsf{P}_{U_a} G\nabla V\nabla V^\mathsf{T} + \mathsf{P}_V G U_a S_{A_a}^{-1}\Lambda' U_a^\mathsf{T}
\end{aligned} \tag{5.55}$$

where $G\mathsf{P}_V^\mathsf{T} = \mathsf{P}_V G$ was used in (5.54)-(5.55) and $\mathsf{P}_{U_a} = I_n - \mathsf{P}_V G U_a S_{A_a}^{-1}U_a^\mathsf{T}$ was defined. Notice that $U_a^\mathsf{T}\mathsf{P}_{U_a} = U_a^\mathsf{T} - S_{A_a}S_{A_a}^{-1}U_a^\mathsf{T} = 0$. Therefore, $\mathsf{P}_{U_a}$ is a generalized projection matrix for the orthogonal complement of the column space of $U_a$. Then, substituting (5.54)-(5.55) in (5.50) yields an expanded formula for the boundary closed-loop Jacobian:

$$J_{f_{cl}}(x_e) = \mathsf{P}_{U_a}\mathsf{P}_V J_a(x_e) - \frac{1}{c}\alpha'(V)\mathsf{P}_{U_a} G\nabla V\nabla V^\mathsf{T} - \mathsf{P}_V G U_a S_{A_a}^{-1}\Lambda' U_a^\mathsf{T} \tag{5.56}$$

Equation (5.56) is the formula for the boundary closed-loop Jacobian appearing in [69]. Next, Definition 5.1.5 will be used to reduce (5.56) to a simpler form. To achieve this, notice that its possible to obtain an algebraic expression for the inverse of $S_{A_a} = U_a^\mathsf{T}\mathsf{P}_V G U_a$ by using the matrix inversion lemma known as the Sherman–Morrison–Woodbury formula:

$$\begin{aligned}
S_{A_a}^{-1} = (U_a^\mathsf{T}\mathsf{P}_V G U_a)^{-1} &= \left(R^\mathsf{T}Q^\mathsf{T}\left(I_n - \frac{1}{c}G\nabla V\nabla V^\mathsf{T}\right)GQR\right)^{-1} \\
&= R^{-1}\left(I_r + Q^\mathsf{T}G\nabla V\left(-\frac{1}{c}\right)\nabla V^\mathsf{T}GQ\right)^{-1}R^{-\mathsf{T}} \\
&= R^{-1}\left(I_r - Q^\mathsf{T}G\nabla V\left(\underbrace{-c + \nabla V^\mathsf{T}GQQ^\mathsf{T}G\nabla V}_{-\eta^{-1}p^{-1}}\right)^{-1}\nabla V^\mathsf{T}GQ\right)R^{-\mathsf{T}} \\
&= R^{-1}\left(I_r + \eta p Q^\mathsf{T}G\nabla V\nabla V^\mathsf{T}GQ\right)R^{-\mathsf{T}}
\end{aligned} \tag{5.57}$$

where $U_a = QR$, $Q^\mathsf{T} GQ = I_r$ and property (iii) of (5.34) were used. Then, using (5.57):

$$\begin{aligned}
\mathsf{P}_V G U_a S_{A_a}^{-1} &= \left(I_n - \frac{1}{c} G\nabla V \nabla V^\mathsf{T}\right) GQRR^{-1} \left(I_r + \eta p Q^\mathsf{T} G \nabla V \nabla V^\mathsf{T} GQ\right) R^{-\mathsf{T}} \\
&= \left(I_n - \frac{1}{c} G\nabla V \nabla V^\mathsf{T}\right) \left(I_n + \eta p GQQ^\mathsf{T} G \nabla V \nabla V^\mathsf{T}\right) GQR^{-\mathsf{T}} \\
&= \left(I_n + \eta p GQQ^\mathsf{T} G\nabla V \nabla V^\mathsf{T} - \frac{1}{c} G\nabla V \nabla V^\mathsf{T} - \frac{\eta p}{c} G\nabla V \underbrace{\nabla V^\mathsf{T} GQQ^\mathsf{T} G \nabla V}_{c - \eta^{-1} p^{-1}} \nabla V^\mathsf{T}\right) GQR^{-\mathsf{T}} \\
&= \left(I_n + \eta p GQQ^\mathsf{T} G\nabla V \nabla V^\mathsf{T} \cancel{- \frac{1}{c} G\nabla V \nabla V^\mathsf{T}} - \eta p G \nabla V \nabla V^\mathsf{T} \cancel{+ \frac{1}{c} \nabla V \nabla V^\mathsf{T}}\right) GQR^{-\mathsf{T}} \\
&= \left(I_n - \eta p G \left(\underbrace{\nabla V - QQ^\mathsf{T} G \nabla V}_{z_v}\right) \nabla V^\mathsf{T}\right) GQR^{-\mathsf{T}} \\
&= \left(I_n - \eta p G z_v \nabla V^\mathsf{T}\right) GQR^{-\mathsf{T}}
\end{aligned} \tag{5.58}$$

where (5.33) and property (iii) of (5.34) were used. Using (5.58):

$$\mathsf{P}_V G U_a S_{A_a}^{-1} U_a^\mathsf{T} = \left(I_n - \eta p G z_v \nabla V^\mathsf{T}\right) GQR^{-\mathsf{T}} R^\mathsf{T} Q^\mathsf{T} = \left(I_n - \eta p G z_v \nabla V^\mathsf{T}\right) GQQ^\mathsf{T} \tag{5.59}$$

Using (5.59):

$$\begin{aligned}
\mathsf{P}_V G U_a S_{A_a}^{-1} U_a^\mathsf{T} \mathsf{P}_V &= \left(I_n - \eta p G z_v \nabla V^\mathsf{T}\right) GQQ^\mathsf{T} \left(I_n - \frac{1}{c} G \nabla V \nabla V^\mathsf{T}\right) \\
&= GQQ^\mathsf{T} - \eta p G z_v \nabla V^\mathsf{T} GQQ^\mathsf{T} - \frac{1}{c} GQQ^\mathsf{T} G \nabla V \nabla V^\mathsf{T} + \frac{\eta p}{c} G z_v \underbrace{\nabla V^\mathsf{T} GQQ^\mathsf{T} G \nabla V}_{c - \eta^{-1} p^{-1}} \nabla V^\mathsf{T} \\
&= GQQ^\mathsf{T} + \eta p G z_v \underbrace{(\nabla V - QQ^\mathsf{T} G \nabla V)}_{z_v}{}^\mathsf{T} - \frac{1}{c} G \underbrace{(z_v + QQ^\mathsf{T} G \nabla V)}_{\nabla V} \nabla V^\mathsf{T} \\
&= GQQ^\mathsf{T} + \eta p G z_v z_v^\mathsf{T} - \frac{1}{c} G \nabla V \nabla V^\mathsf{T}
\end{aligned} \tag{5.60}$$

where again (5.33) and property (iii) of (5.34) were used. Now we are properly equipped to simplify the terms appearing in (5.56). Using (5.60), notice that the product $\mathsf{P}_{U_a} \mathsf{P}_V$ can be rewritten:

$$\begin{aligned}
\mathsf{P}_{U_a} \mathsf{P}_V &= \left(I_n - \mathsf{P}_V G U_a S_{A_a}^{-1} U_a^\mathsf{T}\right) \mathsf{P}_V = I_n \cancel{- \frac{1}{c} G \nabla V \nabla V^\mathsf{T}} - GQQ^\mathsf{T} - \eta p G z_v z_v^\mathsf{T} \cancel{+ \frac{1}{c} G \nabla V \nabla V^\mathsf{T}} \\
&= I_n - GQQ^\mathsf{T} - \eta p G z_v z_v^\mathsf{T}
\end{aligned} \tag{5.61}$$

Using (5.58)-(5.59), the second term appearing at (5.56) can be further simplified by

$$\begin{aligned} c^{-1}\mathsf{P}_{U_a}G\nabla V &= c^{-1}G\nabla V - c^{-1}\mathsf{P}_V GU_a S_{A_a}^{-1}U_a^{\mathsf{T}}G\nabla V \\ &= c^{-1}G\nabla V - c^{-1}\left(I_n - \eta pGz_v\nabla V^{\mathsf{T}}\right)GQQ^{\mathsf{T}}G\nabla V \\ &= c^{-1}G(\underbrace{\nabla V - QQ^{\mathsf{T}}G\nabla V}_{z_v}) + \eta pc^{-1}Gz_v\underbrace{\nabla V^{\mathsf{T}}GQQ^{\mathsf{T}}G\nabla V}_{c-\eta^{-1}p^{-1}} = \eta pGz_v \end{aligned} \tag{5.62}$$

Then, substituting (5.61), (5.62) and (5.58) into the closed-loop Jacobian (5.56) yields

$$\begin{aligned} J_{f_{cl}}(x_e) = &\left(\underbrace{I_n - GQQ^{\mathsf{T}} - \eta pGz_vz_v^{\mathsf{T}}}_{\mathsf{P}_{U_a}\mathsf{P}_V}\right)J_a(x_e,\lambda_a(x_e)) - \alpha'(V)\ \underbrace{\eta pGz_v}_{c^{-1}\mathsf{P}_{U_a}G\nabla V}\ \underbrace{(z_v^{\mathsf{T}} + \nabla V^{\mathsf{T}}GQQ^{\mathsf{T}})}_{\nabla V^{\mathsf{T}}} \\ &- \underbrace{(I_n - \eta pGz_v\nabla V^{\mathsf{T}})GQR^{-\mathsf{T}}}_{\mathsf{P}_V GU_aS_{A_a}^{-1}}\Lambda'\underbrace{R^{\mathsf{T}}Q^{\mathsf{T}}}_{U_a^{\mathsf{T}}} \\ = &\left(I_n - GQQ^{\mathsf{T}} - \eta pGz_vz_v^{\mathsf{T}}\right)J_a(x_e,\lambda_a(x_e)) - \alpha'(V)\eta pGz_vz_v^{\mathsf{T}} \\ &- \eta pGz_v\underbrace{\nabla V^{\mathsf{T}}GQ\left(\alpha'(V)I_r - R\Lambda'R^{-1}\right)^{\mathsf{T}}}_{\Delta\in\mathbb{R}^{r\times r}}Q^{\mathsf{T}} - GQ(R\Lambda'R^{-1})^{\mathsf{T}}Q^{\mathsf{T}} \end{aligned} \tag{5.63}$$

Defining $\Delta = \nabla V^{\mathsf{T}}GQ\left(\alpha'(V)I_r - R\Lambda'R^{-1}\right)^{\mathsf{T}}$, the terms in (5.63) can be grouped and expressed as outer products between matrices, as

$$\begin{aligned} J_{f_{cl}}(x_e) = &\left(I_n - G\underbrace{\begin{bmatrix}Q & z_v\end{bmatrix}}_{Z}\underbrace{\begin{bmatrix}I_r & 0\\ 0 & \eta p\end{bmatrix}}_{N_1}\underbrace{\begin{bmatrix}Q^{\mathsf{T}}\\ z_v^{\mathsf{T}}\end{bmatrix}}_{Z^{\mathsf{T}}}\right)J_a(x_e,\lambda_a(x_e)) \\ &- G\underbrace{\begin{bmatrix}Q & z_v\end{bmatrix}}_{Z}\underbrace{\begin{bmatrix}I_r & 0\\ 0 & \eta p\end{bmatrix}}_{N_1}\underbrace{\begin{bmatrix}(R\Lambda'R^{-1})^{\mathsf{T}} & 0\\ \Delta & \alpha'(V)\end{bmatrix}}_{\Psi}\underbrace{\begin{bmatrix}Q^{\mathsf{T}}\\ z_v^{\mathsf{T}}\end{bmatrix}}_{Z^{\mathsf{T}}} \\ = &\left(I_n - GZN_1Z^{\mathsf{T}}\right)J_a(x_e,\lambda_a(x_e)) - GZN_1\Psi Z^{\mathsf{T}} \end{aligned}$$

which is the expression at (5.35), with matrices $N_1$ and $\Psi$ from (5.36). Lastly, notice that due to the fact that $\alpha$ and the $\beta_i\ \forall i \in \mathscr{A}$ are class $\kappa$ and extended class $\kappa$ functions, respectively, $\Psi > 0$. Furthermore, since $\eta > 0, N_1 > 0$. Matrix $I_n - GZN_1Z^{\mathsf{T}}$ is a matrix operator as defined in (2.1). □

**Assumption 9** (Disjoint Unsafe Sets)**.** The unsafe sets of the $N$ barriers are disjoint, that is:

$$\overline{\mathscr{C}}_i \cap \overline{\mathscr{C}}_j = \emptyset \qquad \forall i \neq j \tag{5.64}$$

**Remark.** The formula for the closed-loop Jacobian at (5.35) is valid for multiple CBFs with possibly intersecting unsafe sets and therefore does not require Assumption 9. The paper [65]

proposes a formula for the closed-loop Jacobian at a boundary equilibrium point akin to (5.35) considering the case of the CLF-CBF QP controller, that is, it does not consider the generalized controller (5.1), accounting for the safety-filter QP as well. The formula [65], Lemma 1, equation (29) is valid under the assumptions of: (i) only one CBF ($N = 1$) or (ii) Assumption 9. In this case, from the conditions of Theorem 5.1.2, there is only one KKT multiplier $\lambda_i$ associated to each boundary equilibrium point, with $i$ being the index of the $i$-th CBF boundary $\partial\mathscr{C}_i$ where the equilibrium point occurs, and the closed-loop Jacobian expression is given by (5.35) with $\hat{z}_i = \frac{\nabla h_i}{\|\nabla h_i\|_G}$, $Q = \begin{bmatrix} \hat{z}_i \end{bmatrix} \in \mathbb{R}^{n\times 1}$, $z_v = \nabla V - \langle \nabla V, \hat{z}_i \rangle_G \hat{z}_i$, $Z = \begin{bmatrix} \hat{z}_i & z_v \end{bmatrix} \in \mathbb{R}^{n\times 2}$, $N_1 = \text{diag}\{1, \eta p\} \in \mathbb{R}^{2\times 2}$, $R\Lambda' R^{-1} = \beta_i'(0)$, $\Delta = \langle \nabla V, \hat{z}_1 \rangle_G (\alpha'(V) - \beta_i(0))$, $\Psi \in \mathbb{R}^{2\times 2}$. The closed-loop Jacobian (5.35) is a generalization of the formula in [65], considering the safety-filter QP and the CLF-CBF QP, both with multiple CBFs with possibly intersecting boundaries.

One more thing can be said about Assumption 9. Assume there exist two barriers $h_i$ and $h_j$ with boundary intersections, that is, the set $\partial\mathscr{C}_i \cap \partial\mathscr{C}_j$ is not empty and (5.64) is not satisfied. Then, it should be possible to construct a new composite barrier $h_k$ such that its unsafe set contains the union of the unsafe sets from $h_i$ and $h_j$ (under mild assumptions on the regularities of $h_k$), thus representing (almost) the same safe region as $\mathscr{C}_i \cap \mathscr{C}_j$. A method for achieving this result is to use of the continuous composite barrier proposed by [52] at (2.42) for the intersection of safe sets.

**Corollary 5.1.2.1.** Under the same assumptions of Proposition 5.1.2, at a *boundary* equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ with $L_g V(x_e) = 0$, the closed-loop Jacobian expression simplifies to

$$J_{f_{cl}}(x_e) = \left(I_n - GQQ^\mathsf{T}\right) J_a(x_e, \lambda_a) - GQ(R\Lambda' R^{-1})^\mathsf{T} Q^\mathsf{T} \tag{5.65}$$

*Proof.* Since $L_g V(x_e) = 0$, $G(x_e)\nabla V(x_e) = 0$ and $z(x_e) = \nabla V(x_e)$. Therefore, $G(x_e) z_v(x_e) = 0$. Then, all terms with $G(x_e) z_v(x_e)$ at (5.63) vanish, and (5.65) contains all the remaining nonzero terms. Notice that matrix $I_n - GQQ^\mathsf{T}$ is an oblique projection matrix to the set $\{\nabla h_{a_1}, \cdots, \nabla h_{a_r}\}^\perp$, since $Q^\mathsf{T}(I_n - GQQ^\mathsf{T}) = Q^\mathsf{T} - Q^\mathsf{T} = 0$. □

### 5.1.2.3 Stability Properties of Equilibrium Points

Here, we use the previously obtained results to derive general stability properties for all types of equilibrium points occurring at the CBF QP framework. First, we present a known lemma that will allow us to demonstrate some of the results from this section.

**Lemma 5.1.3.** Let $X \in \mathbb{R}^{n\times n}$ be a symmetric positive semi-definite matrix, and let $\mathscr{Z} = \{z_1, \cdots, z_r\}$ be any set of $r$ linearly independent vectors such that $z_i \notin \mathscr{N}(X), i = 1, \cdots, r$. Additionally, let $\mathscr{V} = \{v_1, \cdots, v_{n-r}\}$ be a basis for the orthogonal complement of $\mathscr{Z}$ with the standard inner product $\langle \cdot, \cdot \rangle$, that is, $v_j^\mathsf{T} z_i = 0, \forall i, j$. Moreover, let $\mathscr{W} = \{w_1, \cdots, w_{n-r}\}$ be a basis for the orthogonal complement of $\mathscr{Z}$ with the inner product $\langle \cdot, \cdot \rangle_X$, that is, $w_j^\mathsf{T} X z_i = 0, \forall i, j$. Then, $\{Xz_1, \cdots, Xz_r\} \cup \mathscr{V}$ and $\mathscr{Z} \cup \mathscr{W}$ are bases for $\mathbb{R}^n$.

*Proof.* First, we demonstrate the linear independence of the vectors in $\{Xz_1,\cdots,Xz_r\}\cup\mathscr{V}$. Define

$$\sum_{i=1}^{r} a_i X z_i + \sum_{i=1}^{n-r} b_i v_i = 0\,. \tag{5.66}$$

Taking the standard inner product $\langle\cdot,\cdot\rangle$ of (5.66) with $z_j \in \mathscr{Z}$ yields $\sum_{i=1}^{r} a_i\langle z_j, z_i\rangle_X = 0$, $j = 1,\cdots,r$. Defining $Z = \begin{bmatrix} z_1 & \cdots & z_r \end{bmatrix} \in \mathbb{R}^{n\times r}$ and stacking all the $a_i$ constants in a column vector $a \in \mathbb{R}^r$ yields

$$Z^{\mathsf{T}} X Z a = 0 \tag{5.67}$$

Since $Z$ is full column rank and none of its columns lies in the nullspace of $X$, matrix $Z^{\mathsf{T}}XZ \in \mathbb{R}^{r\times r}$ is full rank, which means that the only solution to the homogeneous equation (5.67) is $a_i = 0$, $i = 1,\cdots,r$. Then, (5.66) reduces to $\sum_{i=1}^{n-r} b_i v_i = 0$. Since the vectors from $\mathscr{V}$ are linearly independent (since $\mathscr{V}$ is a basis), we have $b_i = 0$, $i = 1,\cdots,n-r$. Therefore, the set $\{Xz_1,\cdots,Xz_r\}\cup\mathscr{V}$ is composed of $n$ linearly independent vectors, constituting a basis for $\mathbb{R}^n$.

Next, we demonstrate the linear independence of the vectors in $\mathscr{Z}\cup\mathscr{W}$. Define

$$\sum_{i=1}^{r} a_i z_i + \sum_{i=1}^{n-r} b_i w_i = 0\,. \tag{5.68}$$

Taking the inner product $\langle\cdot,\cdot\rangle_X$ of (5.68) with $z_j \in \mathscr{Z}$ yields $\sum_{i=1}^{r} a_i\langle z_j, z_i\rangle_X = 0$, $j = 1,\cdots,r$, and by the same arguments as before, the unique solution is $a_i = 0$, $i = 1,\cdots,r$. Equation (5.68) reduces to $\sum_{i=1}^{n-r} b_i w_i = 0$, and since the vectors from $\mathscr{W}$ are linearly independent (since $\mathscr{W}$ is a basis), we have $b_i = 0$, $i = 1,\cdots,n-r$. Therefore, the set $\mathscr{Z}\cup\mathscr{W} = \{z_1,\cdots,z_r,w_1,\cdots,w_{n-r}\}$ is composed of $n$ linearly independent vectors, constituting a basis for $\mathbb{R}^n$. □

**Theorem 5.1.4** (Stability of Interior Equilibria)**.** An interior equilibrium point $x_e \in \mathscr{E}_{int(\mathscr{C})}$ on the conditions of Theorem 5.1.1 is locally asymptotically stable if the Jacobian $J_{f_a}(x_e,0)$ as defined in (5.4) is Hurwitz stable.

*Proof.* First, notice that an expression for the inverse of the positive definite matrix $\mathsf{P}_V$ can be found by using the matrix inversion lemma (Sherman–Morrison–Woodbury formula), as before:

$$\mathsf{P}_V^{-1} = I_n + pG\nabla V\nabla V^{\mathsf{T}}\,. \tag{5.69}$$

Then, left-multiplying the interior closed-loop Jacobian (5.26) by (5.69) yields

$$\begin{aligned}
\mathsf{P}_V^{-1}(x_e)J_{f_{cl}}(x_e) &= J_a(x_e,0) - \frac{\alpha'(V)}{c}\left(I_n + pG\nabla V\nabla V^{\mathsf{T}}\right)G\nabla V\nabla V^{\mathsf{T}} \\
&= J_a(x_e,0) - \frac{\alpha'(V)}{c}\underbrace{\left(1+p\|\nabla V\|_G^2\right)}_{pc}G\nabla V\nabla V^{\mathsf{T}} \\
&= J_a(x_e,0) - p\alpha'(V)G\nabla V\nabla V^{\mathsf{T}} = J_{f_a}(x_e,0)
\end{aligned} \tag{5.70}$$

By (5.70), notice that $J_{f_{cl}}(x_e) = \mathsf{P}_V(x_e)J_{f_a}(x_e,0)$. Then, the corresponding Lyapunov equation for $J_{f_{cl}}(x_e)$ is given by

$$J_{f_a}(x_e,0)^\mathsf{T}\mathsf{P}_V(x_e)^\mathsf{T}X + X\mathsf{P}_V(x_e)J_{f_a}(x_e,0) = -Y\,, \tag{5.71}$$

where $X > 0$ is given according to the following cases.

**Case 1.** $L_gV(x_e) \neq 0$: in this case, $X = \lambda_v \nabla V \nabla V^\mathsf{T} + W\Lambda_w W^\mathsf{T} > 0$ is an arbitrary positive definite matrix where $\lambda_v > 0$, $W = \begin{bmatrix} w_1 & \cdots & w_{n-1} \end{bmatrix} \in \mathbb{R}^{n\times(n-1)}$ is such that $w_i^\mathsf{T} G\nabla V = 0$, $i = 1,\cdots,n-1$, $\lambda_v > 0$ and $\Lambda_w \in \mathbb{R}^{n-1}$ is a positive definite diagonal matrix. Due to Lemma 5.1.3, the set $\{\nabla V, w_1, \cdots, w_{n-1}\}$ constitutes a basis for $\mathbb{R}^n$, and therefore $X > 0$. Notice that

$$\begin{aligned} X\mathsf{P}_V &= \left(\lambda_v \nabla V \nabla V^\mathsf{T} + W\Lambda_w W^\mathsf{T}\right)\left(I_n - \frac{1}{c}G\nabla V \nabla V^\mathsf{T}\right) \\ &= \underbrace{\lambda_v\left(1 - \frac{\|\nabla V\|_G^2}{c}\right)}_{p^{-1}c^{-1}>0}\nabla V\nabla V^\mathsf{T} + W\Lambda_w W^\mathsf{T} = \bar{X} \end{aligned} \tag{5.72}$$

Notice that $\bar{X} = \bar{X}^\mathsf{T} > 0$ is also arbitrary and positive definite. Therefore, using (5.72), (5.71) becomes $J_{f_a}(x_e,0)^\mathsf{T}\bar{X} + \bar{X}J_{f_a}(x_e,0) = -Y$, which by Lyapunov's theorem, always has an unique solution $\bar{X} > 0$ for all $Y > 0$ if and only if $J_{f_a}(x_e,0)$ is Hurwitz stable.

**Case 2.** $L_gV(x_e) = 0$: in this case, $X$ is just an arbitrary positive definite matrix, and $\mathsf{P}_V = I_n$. Then, trivially, (5.71) becomes $J_{f_a}(x_e,0)^\mathsf{T}X + XJ_{f_a}(x_e,0) = -Y$, which by Lyapunov's theorem, always has an unique solution $X > 0$ for all $Y > 0$ if and only if $J_{f_a}(x_e,0)$ is Hurwitz stable.

Therefore, in both cases, the interior equilibrium point $x_e$ is locally asymptotically stable if $J_{f_a}(x_e,0)$ is Hurwitz stable. □

Theorem 5.1.4 shows how the stability of *interior* equilibrium points is determined by $J_{f_a}(x_e,0)$ as defined in (5.4), including the origin, provided that $0 \in int(\mathscr{C})$. An immediate consequence of this fact is that $J_{f_{\text{nom}}}(0)$ must be Hurwitz stable for the origin to be an stable equilibrium point with the safety-filter QP, since $J_{f_a}(0,0) = J_{f_{\text{nom}}}(0)$, where $V(0) = 0$ and $\nabla V(0) = 0$ were used. In the case of the CLF-CBF QP, since $u_{\text{nom}}(x) = 0$, $J_{f_a}(0,0) = J_f(0)$, and we conclude that the Jacobian of the *drift* term $f$ in (2.13) must be Hurwitz stable for the origin to be an stable equilibrium point.

**Theorem 5.1.5** (Stable Eigenvalues of the Boundary Jacobian)**.** Let $J_{f_{cl}}(x_e)$ be the closed-loop Jacobian matrix at a *boundary* equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ at the intersection of $r \leq n$ CBF boundaries whose set of indexes is $\mathscr{A}$, under the same conditions of Proposition 5.1.2. Then, $J_{f_{cl}}(x_e)$ has at least $r$ strictly negative eigenvalues equal to $-\beta_i'(0)$, $\forall i \in \mathscr{A}$, with corresponding left eigenvectors equal to the $r$ CBF gradients $\nabla h_i(x_e)$, $\forall i \in \mathscr{A}$. In particular, if $x_e$ occurs at the intersection of exactly $r = n$ CBF boundaries, then $x_e$ is locally asymptotically stable.

*Proof.* For this proof, it is simpler to consider the alternate formula for the closed-loop boundary Jacobian $J_{f_{cl}}(x_e)$ shown in (5.56). Using the Schur complement $S_{A_a} = U_a^\mathsf{T}\mathsf{P}_V G U_a$ and the previous observation that $U_a^\mathsf{T}\mathsf{P}_{U_a} = 0$, (5.56) can be left-multiplied by $U_a^\mathsf{T}$, yielding $U_a^\mathsf{T}J_{f_{cl}} = -\Lambda' U_a^\mathsf{T}$,

thus showing that the $r$ columns of $U_a$ (that is, the active CBF gradients $\nabla h_i(x_e)$, $\forall i \in \mathscr{A}$) are left-eigenvectors of the closed-loop boundary Jacobian with corresponding strictly negative eigenvalues $-\beta_i'(0)$, $\forall i \in \mathscr{A}$. In particular, it follows that if the boundary equilibrium point $x_e$ occurs at the intersection of exactly $r = n$ CBF boundaries, all eigenvalues of $J_{f_{cl}}(x_e)$ are strictly negative, meaning that $x_e$ must be locally asymptotically stable. □

**Remark.** Theorem 5.1.5 has important implications. The first one is that there is always a stable manifold associated to any boundary equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ under the conditions of Theorem 5.1.2, with the tangent space to this manifold at $x_e$ being $\text{span}\{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}$. If $x_e$ is unstable, it must be so due to the remaining $n-r$ eigenvalues of $J_{f_{cl}}(x_e)$. The second implication is that if there exist a number of CBF boundary intersections equal to the dimension of the state space $\mathbb{R}^n$, boundary equilibrium points occurring at this intersection and under the conditions of Theorem 5.1.2 will always be stable. This is an important limitation of the safety-filter and CLF-CBF framework with multiple CBFs with intersecting unsafe sets: if the conditions for the *existence* of a certain boundary equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ occurring at $n$ CBF boundary intersections cannot be lifted (for example, by changing the nominal controller $u_{\text{nom}}$ or the CLF $V$), then it is also impossible to guarantee the global stability of the origin, since a non-zero measure attractor for $x_e$ will always exist.

The next key result of this section provides a necessary and sufficient condition for the stability of boundary equilibrium points arising in the CBF-based QP formulation for safety-critical control with controller (5.1).

**Theorem 5.1.6** (Stability of Boundary Equilibria)**.** Consider a boundary equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ of the closed-loop system (2.14) with controller (2.37), with $r = |\mathscr{A}| \leq n$ active CBF constraints, under the conditions of Proposition 5.1.2. If there exists $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^{\perp}$ such that

$$v^{\mathsf{T}} J_{f_a}(x_e, \lambda_a) v > 0\,, \tag{5.73}$$

then $x_e$ is unstable. Otherwise, it is locally asymptotically stable. In (5.73), $J_{f_a}$ is the Jacobian of $f_a$ with respect to $x$, as defined in (5.4), with the KKT multipliers $\lambda_a \in \mathbb{R}^r_{\geq 0}$ computed at $x_e$.

*Proof.* Consider any equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$, with closed-loop Jacobian matrix given by (5.35). An approximation for the closed-loop system dynamics on a neighborhood of $x_e$ is given by the Taylor series

$$\dot{x} = J_{f_{cl}}(x_e)\Delta x + \text{higher order terms} \tag{5.74}$$

where $\Delta x = x - x_e \in \mathbb{R}^n$ is a disturbance vector around the equilibrium point. By Lemma 5.1.3, let us write this vector using the basis $\{G(x_e)z_1(x_e), G(x_e)z_r(x_e), v_1(x_e), \cdots, v_{n-r}(x_e)\}$, with vectors $z_1, \cdots, z_r$ being the orthogonal basis (with respect to $\langle\cdot,\cdot\rangle_{G(x_e)}$) for the span of $\{\nabla h_{a_1}, \cdots, \nabla h_{a_r}\}$ at $x_e$, as in Definition 5.1.5, and with $v_1, \cdots, v_{n-r}$ being $n-r$ fixed basis vectors for the orthogonal

space of span$\{\nabla h_{a_1}(x_e),\cdots,\nabla h_{a_r}(x_e)\}$, denoted as $\{\nabla h_{a_1}(x_e),\cdots,\nabla h_{a_r}(x_e)\}^{\perp}$. Therefore,

$$\Delta x = \sum_{i=1}^{r} Gz_i(x_e)a_i + \sum_{i=1}^{n-r} b_i v_i(x_e) = G(x_e)Q(x_e)a + v \tag{5.75}$$

with $a = \begin{bmatrix} a_1 & \cdots & a_r \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^r$ and $v = \sum_{i=1}^{n-r} b_i v_i(x_e) \in \{\nabla h_{a_1}(x_e),\cdots,\nabla h_{a_r}(x_e)\}^{\perp}$. Additionally, $v^{\mathsf{T}}Q(x_e) = 0$ or $v^{\mathsf{T}}U_a(x_e) = 0$ by construction. Substituting (5.75) in (5.74) and left-multiplying the resulting expression by $U_a = QR$ (ignoring higher order terms) yields

$$\begin{aligned} R^{\mathsf{T}}Q^{\mathsf{T}}\dot{x} &= U_a^{\mathsf{T}} J_{f_{cl}}(x_e)\left(GQa+v\right) = -\Lambda' U_a^{\mathsf{T}}\left(GQa+v\right) \\ &= -\Lambda' R^{\mathsf{T}}Q^{\mathsf{T}}GQa - \Lambda' U_a^{\mathsf{T}} v = -\Lambda' R^{\mathsf{T}} a \end{aligned} \tag{5.76}$$

where $U_a^{\mathsf{T}} J_{f_{cl}} = -\Lambda' U_a^{\mathsf{T}}$ from Theorem 5.1.5 was used. Since $GQ$ is a fixed matrix at $x_e$, the time derivative of (5.75) is $\dot{x} = GQ\dot{a} + \dot{v}$, which can be substituted at the left-hand side of (5.76), yielding $R^{\mathsf{T}}Q^{\mathsf{T}}(GQ\dot{a}+\dot{v}) = R^{\mathsf{T}}\dot{a} = -\Lambda' R^{\mathsf{T}}a \Rightarrow \dot{a} = -R^{-\mathsf{T}}\Lambda' R^{\mathsf{T}}a$. Since the matrix $-R^{-\mathsf{T}}\Lambda' R^{\mathsf{T}}$ is similar to $-\Lambda'$, it is Hurwitz stable, which means that $\lim_{t\to\infty} a(t) = 0$. Replacing the dynamics of $a$ into the time derivative of (5.75) and solving for $\dot{v}$ yields the following dynamics for the transformed state:

$$\begin{aligned} \dot{a} &= -R^{-\mathsf{T}}\Lambda' R^{\mathsf{T}} a \\ \dot{v} &= \underbrace{\left(J_{f_{cl}}(x_e)GQ + GQR^{-\mathsf{T}}\Lambda' R^{\mathsf{T}}\right)}_{M(x_e)} a + J_{f_{cl}}(x_e)v \end{aligned} \tag{5.77}$$

From (5.77), since the dynamics of $a$ is determined by an asymptotically stable subsystem, it is evident that the stability of $x_e$ is determined by the properties of the closed-loop boundary Jacobian at the orthogonal space $\{\nabla h_{a_1}(x_e),\cdots,\nabla h_{a_r}(x_e)\}^{\perp}$. To see this, define the Lyapunov function candidate $V(a,v) = a^{\mathsf{T}}X_a a + v^{\mathsf{T}}X_v v > 0$, where $X_a = X_a^{\mathsf{T}} > 0$, $X_v = X_v^{\mathsf{T}} > 0$. Using (5.77), its time derivative is given by

$$\dot{V} = -\begin{bmatrix} a \\ v \end{bmatrix}^{\mathsf{T}} \underbrace{\begin{bmatrix} Y_a & -M^{\mathsf{T}}X_v \\ -X_v M & -Y_v \end{bmatrix}}_{Y} \begin{bmatrix} a \\ v \end{bmatrix} \tag{5.78}$$

$$Y_a = R\Lambda' R^{-1}X_a + X_a R^{-\mathsf{T}}\Lambda' R^{\mathsf{T}} \tag{5.79}$$

$$Y_v = J_{f_{cl}}(x_e)^{\mathsf{T}}X_v + X_v J_{f_{cl}}(x_e) \tag{5.80}$$

From (5.78), $x_e$ is asymptotically stable if $Y > 0$, and by Theorem 2.2.2 (Chetaev's instability), $x_e$ is unstable if at least one of the real eigenvalues of $Y$ is strictly negative. By Lyapunov's theorem, since $-R\Lambda' R^{-1}$ is Hurwitz stable, given any $Y_a > 0$, there exists an unique $X_a > 0$ solving (5.79). Choosing any $Y_a > 0$, matrix $Y$ is positive definite if and only if $-Y_v - X_v M Y_a^{-1} M^{\mathsf{T}} X_v > 0$ or, equivalently, $-Y_v > X_v M Y_a^{-1}(X_v M)^{\mathsf{T}} > 0$ using the Loewner ordering for symmetric positive

definite matrices. Provided that $Y_v < 0$, this matrix inequality can always be satisfied by choosing a sufficiently large $Y_a > 0$ , thus making the positive definite term $X_v M Y_a^{-1} M^\mathsf{T} X_v$ arbitrarily small when compared to the positive definite term $-Y_v$. Therefore, expanding the expression for $\dot{V}$, the stability of the boundary equilibrium point $x_e$ is conditioned to the definiteness of the quadratic term on $v$ in (5.78), namely

$$v^\mathsf{T} Y_v v = v^\mathsf{T} \left( J_{f_{cl}}(x_e)^\mathsf{T} X_v + X_v J_{f_{cl}}(x_e) \right) v \tag{5.81}$$

with $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$. In summary, $x_e \in \mathscr{E}_{\partial \mathscr{C}_a}$ is *asymptotically stable* if the term (5.81) is negative definite $\forall v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$. On the contrary, $x_e \in \mathscr{E}_{\partial \mathscr{C}_a}$ is *unstable* if $\exists v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$ such that (5.81) is positive.

To demonstrate condition (5.73), (5.81) will be reduced to a simpler form considering two distinct cases: (i) $L_g V(x_e) \neq 0$ and (ii) $L_g V(x_e) = 0$.

**Case 1.** $L_g V(x_e) \neq 0$: in this case, the following arbitrary parametrization for $X_v$ is proposed:

$$X_v = Z \Lambda_z Z^\mathsf{T} + W \Lambda_w W^\mathsf{T} > 0 \tag{5.82}$$

where $\Lambda_z = \text{diag}\{\Lambda_Q, \sigma_{z_v}\}$, $\Lambda_Q \in \mathbb{R}^{r \times r}$ are diagonal and positive definite, $Z = \begin{bmatrix} Q & z_v \end{bmatrix} \in \mathbb{R}^{n \times (r+1)}$ is the matrix defined at (5.35), $\Lambda_w \in \mathbb{R}^{(n-r-1)\times(n-r-1)}$ is also diagonal and positive definite, and $W \in \mathbb{R}^{n \times (n-r-1)}$ is a full column rank matrix whose columns are orthogonal to the columns of $Z$ with respect to the inner product $\langle \cdot, \cdot \rangle_G$, that is, $W^\mathsf{T} G Z = 0$. According to Lemma 5.1.3, the set containing the $n$ column vectors of $Z$ and $W$ constitutes a basis for $\mathbb{R}^n$, and therefore $X_v$ at (5.82) is an arbitrary positive definite matrix. Then, using (5.82) and (5.35) yields:

$$\begin{aligned} X_v J_{f_{cl}}(x_e) &= \left( Z \Lambda_z Z^\mathsf{T} + W \Lambda_w W^\mathsf{T} \right) \left( I_n - G Z N_1 Z^\mathsf{T} \right) J_a(x_e, \lambda_a) - \left( Z \Lambda_z Z^\mathsf{T} + W \Lambda_w W^\mathsf{T} \right) G Z N_1 \Psi Z^\mathsf{T} \\ &= Z \Lambda_z \underbrace{(I_{r+1} - \bar{N}_1)}_{N_0} Z^\mathsf{T} J_a(x_e, \lambda_a) + W \Lambda_w W^\mathsf{T} J_a(x_e, \lambda_a) - Z \Lambda_z \bar{N}_1 \Psi Z^\mathsf{T} \\ &= \underbrace{\left( \sigma_v \eta z_v z_v^\mathsf{T} + W \Lambda_w W^\mathsf{T} \right)}_{\bar{X}_v} J_a(x_e, \lambda_a) - Z \Lambda_z \bar{N}_1 \Psi Z^\mathsf{T} \end{aligned} \tag{5.83}$$

where $\bar{N}_1 = Z^\mathsf{T} G Z N_1 = \text{diag}\{I_r, 1 - \eta\}$, $N_0 = I_{r+1} - \bar{N}_1 = \text{diag}\{0, \eta\}$ and therefore, $Z \Lambda_z N_0 Z^\mathsf{T} = \sigma_v \eta z_v z_v^\mathsf{T}$. Using (5.83), equation (5.80) becomes

$$Y_v = J_a(x_e, \lambda_a)^\mathsf{T} \bar{X}_v + \bar{X}_v J_a(x_e, \lambda_a) - Z \left( \Psi^\mathsf{T} \Lambda_z \bar{N}_1 + \Lambda_z \bar{N}_1 \Psi \right) Z^\mathsf{T} \tag{5.84}$$

Then, using (5.84) to construct the quadratic form $v^\mathsf{T} Y_v v$ at (5.81) yields

$$v^\mathsf{T} Y_v v = 2 v^\mathsf{T} \bar{X}_v J_a(x_e, \lambda_a) v - 2 v^\mathsf{T} Z \Lambda_z \bar{N}_1 \Psi Z^\mathsf{T} v \tag{5.85}$$

Since $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$, $v^\mathsf{T} Q = 0$, and the second term on the right-hand side of

(5.85) becomes

$$\begin{aligned} v^\mathsf{T} Z \Lambda_z \bar{N}_1 \Psi Z^\mathsf{T} v &= \alpha'(V)\sigma_v(1-\eta)(v^\mathsf{T} z_v)^2 \\ &= \alpha'(V)\sigma_v(1-\eta)(v^\mathsf{T}\nabla V)^2 \end{aligned} \tag{5.86}$$

However, notice that:

$$\begin{aligned} p\alpha'(V)\bar{X}_v G z_v z_v^\mathsf{T} &= p\alpha'(V)\left(\sigma_v \eta z_v z_v^\mathsf{T} + W\Lambda_w W^\mathsf{T}\right) G z_v z_v^\mathsf{T} \\ &= p\alpha'(V)\sigma_v \eta z_v \underbrace{p^{-1}\left(\eta^{-1}-1\right) z_v^\mathsf{T}}_{z_v^\mathsf{T} G z_v} = \alpha'(V)\sigma_v(1-\eta) z_v z_v^\mathsf{T} \end{aligned} \tag{5.87}$$

where $W^\mathsf{T} G z_v = 0$ and $z_v^\mathsf{T} G z_v = p^{-1}(\eta^{-1}-1)$ were used. Next, writing the quadratic form of the rank-one matrix (5.87) with $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$ yields precisely (5.86), again due to the fact that $v^\mathsf{T} Q = 0$. Then, (5.85) can be rewritten as

$$\begin{aligned} v^\mathsf{T} Y_v v &= 2 v^\mathsf{T} \bar{X}_v \underbrace{\left(J_a(x_e,\lambda_a) - p\alpha'(V) G \nabla V \nabla V^\mathsf{T}\right)}_{J_{f_a}(x_e,\lambda_a)} v \\ &= v^\mathsf{T}\left(J_{f_a}(x_e,\lambda_a)^\mathsf{T}\bar{X}_v + \bar{X}_v J_{f_a}(x_e,\lambda_a)\right) v \end{aligned} \tag{5.88}$$

Notice that $\bar{X}_v = \sigma_v \eta z_v z_v^\mathsf{T} + W\Lambda_w W^\mathsf{T}$ is not full rank, since each one of the $r$ columns of matrix $GQ$ is an element of its nullspace (since $\bar{X}_v GQ = 0$). However, the columns of $\bar{X}_v$ span the set $\{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$, being always possible to choose $\sigma_v > 0$ and $\Lambda_v > 0$ in such a way that $\bar{X}_v v = v$. Then, with this choice for $\bar{X}_v$, (5.88) yields

$$v^\mathsf{T} Y_v v = 2 v^\mathsf{T} J_{f_a}(x_e,\lambda_a) v \tag{5.89}$$

**Case 2.** $L_g V(x_e) = 0$: in this case, the closed-loop Jacobian matrix at the equilibrium point is given by (5.65) and $z_v(x_e) = 0$, resulting in the last column of matrix $Z$ from (5.82) vanishing, thus causing $Z$ to lose rank by 1. Thereby, a different arbitrary parametrization for $X_v$ is proposed:

$$X_v = Q\Lambda_Q Q^\mathsf{T} + W\Lambda_w W^\mathsf{T} > 0 \tag{5.90}$$

where $\Lambda_Q \in \mathbb{R}^{r\times r}$ is diagonal and positive definite, $Q \in \mathbb{R}^{n\times r}$ is the matrix whose $r$ columns are the orthonormal vectors $\hat{z}_i$ from the QR decomposition of $U_a$ at (5.32), $\Lambda_w \in \mathbb{R}^{(n-r-1)\times(n-r-1)}$ is also diagonal and positive definite, and $W \in \mathbb{R}^{n\times(n-r)}$ is a matrix whose columns are orthogonal to the columns of $Q$ with respect to the inner product $\langle\cdot,\cdot\rangle_G$, that is, $W^\mathsf{T} GQ = 0$. According to Lemma 5.1.3, the set containing the $n$ column vectors of $Q$ and $W$ constitutes a basis for $\mathbb{R}^n$, and therefore $X_v$ at (5.90) is an arbitrary positive definite matrix. Substituting the closed-system Jacobian matrix (5.65) and matrix (5.90) at (5.80) and using the properties $Q^\mathsf{T} GQ = I_r$, $W^\mathsf{T} GQ = 0$

to simplify the resulting expression yields

$$\begin{aligned} X_v J_{f_{cl}}(x_e) &= \left(Q\Lambda_Q Q^\mathsf{T} + W\Lambda_w W^\mathsf{T}\right)\left(I_n - GQQ^\mathsf{T}\right) J_a(x_e,\lambda_a) \\ &\quad - \left(Q\Lambda_Q Q^\mathsf{T} + W\Lambda_w W^\mathsf{T}\right) GQ(R\Lambda' R^{-1})^\mathsf{T} Q^\mathsf{T} \\ &= \underbrace{\left(Q\Lambda_Q Q^\mathsf{T} + W\Lambda_w W^\mathsf{T}\right)}_{X_v} J_a(x_e,\lambda_a) - Q\Lambda_Q Q^\mathsf{T} J_a(x_e,\lambda_a) - Q\Lambda_Q (R\Lambda' R^{-1})^\mathsf{T} Q^\mathsf{T} \\ &= X_v J_a(x_e,\lambda_a) - Q\Lambda_Q \left(Q^\mathsf{T} J_a(x_e,\lambda_a) + (R\Lambda' R^{-1})^\mathsf{T} Q^\mathsf{T}\right) \end{aligned} \tag{5.91}$$

Writing the quadratic form of $v^\mathsf{T} Y_v v$ in (5.91) with $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$ yields

$$v^\mathsf{T} Y_v v = 2v^\mathsf{T} X_v J_a(x_e,\lambda_a) v = 2v^\mathsf{T} X_v \left(J_{f_a}(x_e,\lambda_a)\Big|_{G\nabla V = 0}\right) v \tag{5.92}$$

Then, since it is always possible to choose $\Lambda_Q > 0$ and $\Lambda_v > 0$ in such a way that $X_v v = v$, (5.92) also yields equation (5.89) but with $G(x_e)\nabla V(x_e) = 0$.

Ultimately, both cases (i) $L_g V(x_e) \neq 0$ and (ii) $L_g V(x_e) = 0$ lead to the conclusion that the equilibrium point is unstable if $\exists v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$ such that (5.89) is strictly positive, that is, condition (5.73) holds. Otherwise, that is, if the right-hand side of (5.89) is non-positive $\forall v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$, then the equilibrium point must be locally asymptotically stable due to the negative definiteness of the remaining terms at the expression for $\dot{V}$ at (5.78). □

Theorems 5.1.4, 5.1.5 and 5.1.6 reveal the stability properties of all equilibrium points formed by the closed-loop system resulting from the application of the generalized QP controller (5.1) to the affine nonlinear system (2.13). In particular, Theorems 5.1.4 and 5.1.6 show that the stability of all equilibrium points is fully determined by the Jacobian of the vector field $f_a$, as defined in (5.4). Regarding the affine nonlinear plant, it comprises the first order partial derivatives of the drift term $f(x)$ and of the input matrix $g(x)$. Regarding the controller parameters, it depends on the first order partial derivatives of the safety-filter nominal controller $u_{\text{nom}}(x)$ and the second order partial derivatives of the CLF $V$ and CBFs $h_1, \cdots, h_N$, that it, their Hessian matrices.

This result is not restricted by any extra assumptions regarding the drift term $f(x)$ or the input matrix $g(x)$. However, it is interesting to recover the results obtained in previous works, namely the ones published in [64]. In the next Corollary, it is shown that, for driftless affine nonlinear systems with full rank $g(x)$ controlled by the CLF-CBF QP controller, the stability of equilibrium points is *completely* determined by the geometry of the CLF-CBF pair alone, being independent on the specificities of the system model $g(x)$. A simplified version of this result was demonstrated for an integrator system, in [64].

**Corollary 5.1.6.1** (Driftless Full rank Systems)**.** For the CLF-CBF QP controller, consider any equilibrium point $x_e \in \mathscr{E}$ that is not a stationary point of $V$, that is $\nabla V(x_e) \neq 0$. Additionally, consider that the system dynamics (2.13) is driftless, that is, $f(x) = 0$ everywhere, and that $g(x)$ is full rank for all values of $x \in \mathscr{C}$. Then, the following statements hold:

(i) $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ for some boundary intersection $\partial\mathscr{C}_a$ with $L_g h_i(x_e) \neq 0$ at least some $i \in \mathscr{A}$.

(ii) $\nabla V(x_e)$ is a conical combination of $\{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}$, that is, $\nabla V(x_e) = U_a(x_e)\lambda$, $\lambda \in \mathbb{R}^r_{\geq 0}$.

(iii) $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ with $L_g h_i(x_e) \neq 0 \; \forall i \in \mathscr{A}$ is unstable if $\exists v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$ such that

$$\frac{1}{\|\sum_{i\in\mathscr{A}}^r \lambda_i \nabla h_i(x_e)\|} \sum_{i\in\mathscr{A}} \lambda_i \|\nabla h_i(x_e)\| \, \kappa_{h_i}(v, x_e) > \kappa_V(v, x_e) \tag{5.93}$$

where $\kappa_V(v, x_e)$ and $\kappa_{h_i}(v, x_e)$ are the *curvatures* of $V$ and $h_i$, respectively, computed at the equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}_a}$ and towards a direction $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$:

$$\kappa_V(v, x_e) = \frac{v^\mathsf{T} H_V(x_e) v}{\|\nabla V(x_e)\|}, \quad \kappa_{h_i}(v, x_e) = \frac{v^\mathsf{T} H_{h_i}(x_e) v}{\|\nabla h_i(x_e)\|}$$

In particular, if $x_e$ does not occur at any boundary intersection, occurring instead at one of the isolated boundaries $\partial\mathscr{C}_i$, (5.93) reduces to $\kappa_{h_i}(v, x_e) > \kappa_V(v, x_e)$.

*Proof.* For a driftless nonlinear system $\dot{x} = g(x)u$ with full rank $g(x)$ and controlled by the CLF-CBF QP controller (2.37) with $N$ CBFs, $u_{\text{nom}}(x) = 0 \Rightarrow f_{\text{nom}}(x) = 0$ and $G(x) > 0 \; \forall x \in \mathbb{R}^n$.

Let $x_e \in \mathscr{E}$ be an *interior* equilibrium point or a *boundary* equilibrium point occurring at the boundary intersection $\partial\mathscr{C}_a$ with $L_g h_i(x_e) = 0 \; \forall i \in \mathscr{A}$, where $\mathscr{A}$ is the set of active CBFs. For both cases, all KKT multipliers vanish at the equilibrium condition. Since $f_{\text{nom}}(x) = 0$, the equilibrium condition reduces to $p\alpha(V(x_e))\nabla V(x_e) = 0$, which only holds if $V(x_e) = 0$ or $\nabla V(x_e) = 0$. In both cases, $x_e$ would be an stationary point of $V$, contradicting the assumption of the Theorem. Therefore, $x_e \in \mathscr{E}$ must be a boundary equilibrium point lying at some boundary intersection with $L_g h_i(x_e) \neq 0$ for at least some $i \in \mathscr{A}$, thereby proving (i).

The equilibrium condition $f_a(x_e, \lambda_a) = 0$ can also be written as

$$f_{\text{nom}}(x_e) + G(x_e)\omega_a(x_e, \lambda_a) = 0, \tag{5.94}$$

$$\omega_a(x, \lambda_a) = -p\alpha(V)\nabla V + \sum_{i\in\mathscr{A}}^r \lambda_i \nabla h_i \tag{5.95}$$

From the assumptions of the theorem, $f_{\text{nom}}(x_e) = 0$ and $G(x_e)$ is full rank. Therefore, (5.94) yields $\omega_a(x_e, \lambda_a) = 0$. Since $\lambda_i \geq 0 \; \forall i \in \mathscr{A}$ in (5.95), the CLF gradient $\nabla V(x_e)$ is a conical combination of the CBF gradients at $x_e$, proving (ii).

Since the CLF and CBF gradients are co-directed at $x_e$, they share the same orthogonal space, that is, $\{\nabla V(x_e)\}^\perp = \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$. Therefore, any $v \in \{\nabla h_{a_1}(x_e), \cdots, \nabla h_{a_r}(x_e)\}^\perp$ also satisfies $v^\mathsf{T}\nabla V(x_e) = 0$. Since $G(x_e) > 0$ is full rank and $\nabla V(x_e) \neq 0$, **Case 2** from the demonstration of Theorem 5.1.6 never applies here. Therefore, using $\bar{X}_v = \sigma_v \eta z_v z_v^\mathsf{T} + W\Lambda_w W^\mathsf{T}$

from **Case 1** of the demonstration of Theorem 5.1.6, (5.88) yields

$$v^\mathsf{T}\bar{X}_v J_{f_a}(x_e,\lambda_a)v = 2v^\mathsf{T}\bar{X}_v\left(\sum_{i\in\mathscr{A}}\lambda_i J_{G\nabla h_i} - p\alpha(V)J_{G\nabla V} - p\alpha'(V)G\nabla V\nabla V^\mathsf{T}\right)v$$
$$= 2v^\mathsf{T}W\Lambda_w W^\mathsf{T}\left(G\left(\sum_{i\in\mathscr{A}}\lambda_i J_{\nabla h_i} - p\alpha(V)J_{\nabla V}\right) + \left[\frac{\partial G}{\partial x_1}\omega_a \cdots \frac{\partial G}{\partial x_n}\omega_a\right]\right)v \quad (5.96)$$

where $v^\mathsf{T}\bar{X}_v = v^\mathsf{T}W\Lambda_w W^\mathsf{T}$ was used, since $v^\mathsf{T}z_v(x_e) = 0$ due to the fact that $v^\mathsf{T}\nabla V(x_e) = 0$. The last term at (5.96) vanishes since $\omega_a(x_e,\lambda_a) = 0$, and $J_{\nabla h_i} = H_{h_i}$ and $J_{\nabla V} = H_V$ are the Hessian matrices of the $i$-th CBF and CLF, respectively. Equation (5.96) then becomes

$$v^\mathsf{T}\bar{X}_v J_{f_a}(x_e,\lambda_a)v = 2v^\mathsf{T}W\Lambda_w W^\mathsf{T}G\left(\sum_{i\in\mathscr{A}}\lambda_i H_{h_i} - p\alpha(V)H_V\right)v \quad (5.97)$$

Notice that the right nullspace of the rank-deficient matrix $GW\Lambda_w W^\mathsf{T}$ is given by the columns of matrix $GZ$ (since $W^\mathsf{T}GZ = 0$). Since $v \in \{\nabla h_{a_1}(x_e),\cdots,\nabla h_{a_r}(x_e)\}^\perp = \{\nabla V(x_e)\}^\perp$ is not in the column space of $GZ$, from the rank-nullity theorem, $v$ must be in the column space of $GW\Lambda_w W^\mathsf{T}$, which means that it's always possible to choose $\Lambda_w > 0$ such that $v^\mathsf{T}W\Lambda_w W^\mathsf{T}G = v^\mathsf{T}$. With this choice for $\bar{X}_v$ in (5.97), the instability condition for $x_e$ becomes

$$v^\mathsf{T}\bar{X}_v J_{f_a}(x_e,\lambda_a)v = 2v^\mathsf{T}\left(\sum_{i\in\mathscr{A}}\lambda_i H_{h_i} - p\alpha(V)H_V\right)v$$
$$= 2\sum_{i\in\mathscr{A}}\lambda_i \underbrace{\frac{v^\mathsf{T}H_{h_i}v}{\|\nabla h_i\|\,\kappa_{h_i}(v,x_e)}} - 2p\alpha(V)\underbrace{\frac{v^\mathsf{T}H_V v}{\|\nabla V\|\,\kappa_V(v,x_e)}} > 0 \quad (5.98)$$

From $\omega_a(x_e,\lambda_a) = 0$, $p\alpha(V)\|\nabla V(x_e)\| = \|\sum_{i\in\mathscr{A}}^r \lambda_i\nabla h_i(x_e)\|$ and (5.98) becomes

$$\sum_{i\in\mathscr{A}}\lambda_i\|\nabla h_i(x_e)\|\,\kappa_{h_i}(v,x_e) > \|\sum_{i\in\mathscr{A}}^r \lambda_i\nabla h_i(x_e)\|\,\kappa_V(v,x_e)\,,$$

which becomes (5.93) when both sides are divided by $\|\sum_{i\in\mathscr{A}}^r \lambda_i\nabla h_i(x_e)\|$. In particular, if $x_e$ occurs at an isolated boundary $\partial\mathscr{C}_i$, the weighted sum of CBF gradients at (5.93) spams only the $i$-th active CBF index, and therefore $p\alpha(V)\|\nabla V(x_e)\| = \lambda_i\|\nabla h_i(x_e)\|$, which means that the sufficient condition for instability of $x_e$ at (5.93) becomes a strictly positive difference between CBF and CLF curvatures $\kappa_{h_i}(v,x_e) - \kappa_V(v,x_e) > 0$ at some direction $v \in \{\nabla h_i(x_e)\}$. □

The result presented in [64], Theorem 2 is a special case of Corollary 5.1.6.1 considering only one CBF. Theorem 5.1.6 fully extends this result by considering: (i) general control-affine nonlinear systems with possibly rank-deficient input matrix $g(x)$, (ii) two CBF-based QP controllers, namely the safety-filter QP and the CLF-CBF QP and (iii) considering multiple CBFs with possibly intersecting boundaries.

**Remark.** Consider the technique of *feedback linearization* for control-affine nonlinear systems with full column rank $g(x)$: in this case, a feedback controller

$$u(x) = g^{\mathsf{T}}(x)\left(g(x)g(x)^{\mathsf{T}}\right)^{-1}u'(x) - f(x)$$

can be applied to (2.13), effectively transforming the system dynamics into an $n$-dimensional integrator $\dot{x} = u'(x)$, where Corollary 5.1.6.1 holds. In that case, if the CLF-CBF QP controller is applied at $u'(x)$, all undesired equilibrium points lie at the CBF boundaries and, under Assumption 9, their stability is completely determined by the difference of curvatures between the active CBF and the CLF.

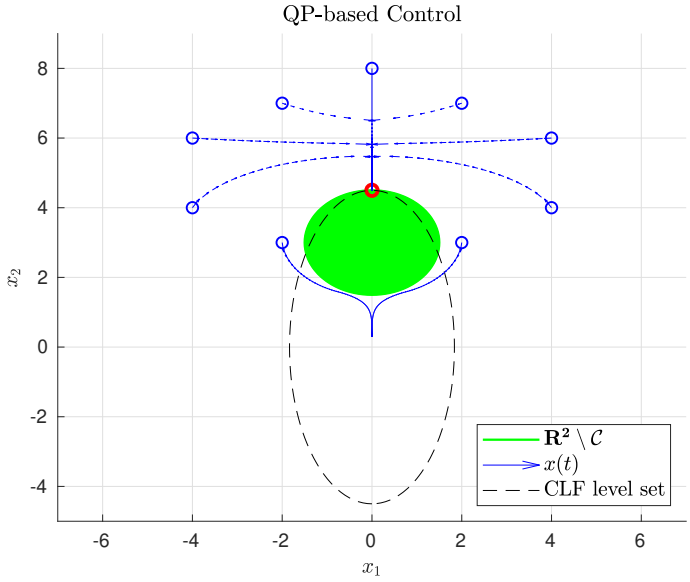



Figure 5.1: Numerical example of trajectories (blue curves) for integrator $\dot{x} = u$ with the QP-based controller (2.37) at different initial conditions, using a quadratic CLF-CBF pair and an integrator system. Some trajectories converge asymptotically to the origin $0 \in \mathbb{R}^2$, but others converge to an stable equilibrium point (red circle) at the boundary of the unsafe set (the green ball).

**Numerical example 1.** In [64], we have demonstrated through simulations that undesirable asymptotically stable equilibria can occur even for a very simple system (an integrator), considering a very simple CLF and CBF (quadratic polynomials). In Fig. 5.1, we illustrate some trajectories of the two-dimensional integrator $\dot{x} = u$ with the CLF-CBF QP-based controller (2.37): many trajectories above the circular green obstacle (the unsafe set for a quadratic CBF) converge to an stable boundary equilibrium point $x_e \in \partial\mathscr{C}$ on the top (red circle). At $x_e$, the CLF and CBF gradients are co-directed and, from Corollary 5.1.6.1, the stability of $x_e$ is due to the fact that the difference between CBF and CLF curvatures at the boundary tangent space at $x_e$ is not strictly positive, as condition (5.93) requires, considering only one CBF.

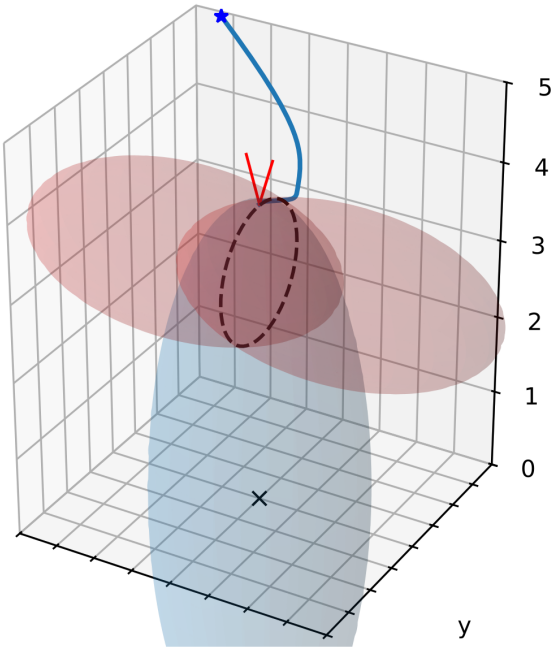



Figure 5.2: Example of a asymptotically stable equilibrium point occurring at the intersection of two quadratic CBF boundaries.

**Numerical example 2.** Figure 5.2 illustrates the simulation of a safety-critical control task for the following linear system $\dot{x} = Bu$, given by

$$\dot{x}_1 = u_1 - 2u_3\,, \quad \dot{x}_2 = u_2\,, \quad \dot{x}_3 = -2u_1 + u_3\,,$$

with the CLF-CBF QP-based controller[1]. Here, $u_{\text{nom}}(x) = 0$ and $H(x) = I_3$ in (5.1). The CBFs are two quadratic functions $h_1$ and $h_2$ centered at the points $(-1,0,3)$ and $(1,0,3)$ with constant positive definite Hessian matrices $H_{h_1}, H_{h_2} \in \mathbb{R}^3$, respectively. Their boundaries are the red ellipsoids in Fig. 5.2, with the union of their interiors constituting the unsafe set which the system trajectories should avoid. The CLF $V$ is also a quadratic centered at the origin with positive definite Hessian $H_V$, whose level set is shown in blue, at an equilibrium point $x_e$ on top of the boundary intersection $\partial\mathscr{C}_a = \partial\mathscr{C}_1 \cap \partial\mathscr{C}_2$, shown in Fig. 5.2 by the dashed black circle. As seen from the state trajectory converging towards $x_e$, the equilibrium point is asymptotically stable. The class $\kappa$ functions used for the CLF and CBFs are simple linear functions $\alpha(V) = \alpha V$ and $\beta_i(V) = \beta_i V$, $i = 1,2$, respectively. For this system, the conditions of Corollary 5.1.6.1 apply. At $x_e$, since the system is driftless and since $g(x) = B$ is full rank,

$$p\alpha(V(x_e))\nabla V(x_e) = \lambda_1 \nabla h_1(x_e) + \lambda_2 \nabla h_2(x_e)\,,$$

meaning that the gradient of the CLF is a conical combination of the gradients of the active CBFs

[1]The code repository used for producing these results is publicly available at `https://github.com/CaipirUltron/CompatibleCLFCBF/tree/mydevel`.

at $x_e$. That is precisely the case at Fig. 5.2, where the CLF gradient (not shown, but pointing at the vertical direction at $x_e$) lies at the cone defined by the two normalized gradients $\nabla h_1(x_e)$, $\nabla h_2(x_e)$, shown as the two red vectors pointing up at the equilibrium point. Furthermore, due to the fact that the system is driftless with a full rank $G(x_e) = BB^\mathsf{T}$, from Corollary 5.1.6.1, there should not exist any $v \in \{\nabla h_1(x_e), \nabla h_2(x_e)\}^\perp$ such that

$$\lambda_1 v^\mathsf{T} H_{h_1} v + \lambda_2 v^\mathsf{T} H_{h_2} v - p\alpha(V(x_e)) v^\mathsf{T} H_V v > 0\,.$$

This is indeed the case from matrices $H_V, H_{h_1}, H_{h_2}$ used in the simulation, which demonstrates theoretically that $x_e$ is indeed asymptotically stable.

By the next definition, we provide a more concise way of writing the conditions for existence and stability of the closed-loop equilibrium points in the CLF-CBF formulation, by stating these conditions in terms of an integral transformation of the CLF $V$.

**Definition 5.1.6.** Given a CLF $V : \mathbb{R} \to \mathbb{R}_+$ centered at the origin with the class $\mathscr{K}$ function $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$, define the function $\overline{V} : \mathbb{R} \to \mathbb{R}_+$ as

$$\overline{V}(x) = \int_0^{V(x)} \alpha(\tau) d\tau \tag{5.99}$$

$$\nabla \overline{V}(x) = \alpha(V) \nabla V \tag{5.100}$$

$$H_{\overline{V}}(x) = \alpha(V) H_V + \alpha'(V) \nabla V \nabla V^\mathsf{T} \tag{5.101}$$

with (5.100) and (5.101) being its gradient and Hessian matrix, respectively.

**Proposition 5.1.3.** The transformed CLF (5.99) has the following properties:

(i) $\overline{V}(x) > 0 \; \forall x \neq 0$ and $\overline{V}(0) = 0$.

(ii) The integral transformation on the right side of (5.99) is bijective.

(iii) $V(x)$ and $\overline{V}(x)$ have the same level sets.

*Proof.* Property (i) can be seem from the fact that $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class $\mathscr{K}$ function, and therefore its integral is positive and strictly increasing. Furthermore, at $x = 0$, since $V(0) = 0$, the limits of integration on the right side become both zero, which means that $\overline{V}(0) = 0$. Property (ii) can also be inferred from the fact that $\alpha$ is of class $\mathscr{K}$: since its integral is a positive and strictly increasing function, its inverse exists. Property (iii) can be demonstrated by noticing that the gradients of $V$ and $\overline{V}$ are co-directed, by (5.100). Since level sets are always orthogonal to the gradient, two functions with collinear gradients everywhere must share the same level sets. □

Particularly, property (ii) means that if the transformed CLF $\overline{V}$ is a known function, then the original $V$ can always be computed from it by inverting the integral transformation (5.99). While a closed formula for the inverse transformation might be difficult or even impossible to derive, it is always possible to compute it efficiently using numerical algorithms.

**Example 1**: Let $\alpha(\tau) = \alpha\tau$ be a class $\mathscr{K}$ function, where $\alpha > 0$ is a positive constant. Then, the transformed CLF is given by

$$\overline{V}(x) = \int_0^{V(x)} \alpha\tau d\tau = \alpha \int_0^{V(x)} \tau d\tau = \alpha \frac{\tau^2}{2}\Big|_0^{V(x)} = \alpha \frac{1}{2} V(x)^2 \tag{5.102}$$

In that case, the inverse transformation of (5.102) can be easily computed: $V(x) = \left(\frac{2}{\alpha}\overline{V}(x)\right)^{\frac{1}{2}}$.

**Example 2**: Let $\alpha(\tau) = \sum_{i=1}^{d} \alpha_i \tau^i$ be a polynomial, where $\alpha_i$, $i = 1, \cdots, d$ are constants such that function $\alpha$ is of class $\mathscr{K}$. Therefore, $\alpha'(\tau) = \sum_{i=1}^{d} i\alpha_i \tau^{i-1} > 0, \forall \tau$. Then, the transformed CLF is

$$\begin{aligned}\overline{V}(x) &= \int_0^{V(x)} \sum_{i=1}^{d} \alpha_i \tau^i d\tau = \sum_{i=1}^{d} \alpha_i \int_0^{V(x)} \tau^i d\tau = \sum_{i=1}^{d} \alpha_i \frac{\tau^{i+1}}{i+1}\Big|_0^{V(x)} = \sum_{i=1}^{d} \alpha_i \frac{1}{i+1} V(x)^{i+1} \\ &= \alpha_1 \frac{1}{2} V(x)^2 + \alpha_2 \frac{1}{3} V(x)^3 + \cdots + \alpha_d \frac{1}{d+1} V(x)^{d+1}\end{aligned} \tag{5.103}$$

Given a known CLF $\overline{V}(x)$, finding a closed formula for the inverse transformation of (5.103) is equivalent to finding a closed-loop formula for the roots of a $d$-degree polynomial in variable $V(x)$, which by the Abel–Ruffini theorem is impossible in general for $d \geq 5$ [93]. However, by property (ii) of Proposition 5.1.3 it is still possible to compute $V(x)$ from a known $\overline{V}(x)$ using numerical methods.

The transformed CLF defined in (5.99) can be used to rewrite (5.3) and (5.4) as

$$f_a(x, \lambda_a) = f_{\text{nom}}(x) - pG(x)\nabla\overline{V}(x) + \sum_{i \in \mathscr{A}} \lambda_i G(x) \nabla h_i(x) \tag{5.104}$$

$$J_{f_a}(x, \lambda_a) = \frac{\partial f_{\text{nom}}}{\partial x} - p\frac{\partial (G\nabla\overline{V})}{\partial x} + \sum_{i \in \mathscr{A}} \lambda_i \frac{\partial (G\nabla h_i)}{\partial x} \tag{5.105}$$

Equations (5.104) and (5.105) provide an alternative interpretation for condition $f_a(x_e, \lambda_a) = 0$ defining the equilibrium manifold at (5.5). Since $\lambda_a \in \mathbb{R}^r_{\geq 0}$, any equilibrium point $x_e \in \mathscr{E}$ must occur inside of a manifold of $\mathbb{R}^n$ where $pG\nabla\overline{V} - f_{\text{nom}}$ is a conical combination of vectors $G\nabla h_i$, $\forall i \in \mathscr{A}$. Furthermore, the stability of $x_e \in \mathscr{E}$ is fully determined by the linear combination of the Jacobian matrices at (5.105).

### 5.1.3 CLF-Compatibility

Our theoretical developments so far show that undesirable *stable* equilibrium points can occur both on the interior or on the boundary of the safe sets, leading to a potential problem of *compatibility* between the safety and stabilization objectives, which was already explored in Section 4.3 of Chapter 4. Although safety is achieved in the form of forward invariance of the system trajectories with respect to the safe set $\mathscr{C} = \cap_{i=1}^{N} \mathscr{C}_i$, the stabilization of the origin might not be achieved for some initial conditions due to the presence of undesirable, possibly stable equilibrium points. As

will be shown in the next sections, methods for dealing with undesirable *boundary* equilibrium points of this type will be developed.

**Assumption 10** (CLF Condition)**.** Given a nonlinear control affine dynamical system (2.13), the CLF $V(x)$ satisfies $L_f V(x) < 0, \forall x \neq x_0 \in \mathbb{R}^n$, where $V(x_0) = 0$.

In [79], a method for dealing with equilibrium points whose corresponding KKT multiplier is $\lambda = 0$ was presented, considering the CLF-CBF QP controller with a single CBF. There, a transformation of the original affine system was proposed, by rewriting (2.13) as

$$\dot{x} = f'(x) + g(x)u'(x) \tag{5.106}$$

where $f'(x) = f(x) + g(x)u_{\text{fb}}(x)$ is a transformed drift term and $u'(x) := u(x) - u_{\text{fb}}(x)$ is a transformed control input. Here, $u_{\text{fb}}(x)$ is a fixed feedback controller, designed in such a way that it satisfies the CLF condition, that is, $L_{f'}V(x) = L_f V(x) + L_g V(x)u_{\text{fb}}(x) < 0$, with $f'(x_0) = 0$ ($x_0$ is the CLF global minimum). In this formulation, the control that is effectively applied to the affine nonlinear system is given by $u(x) = u_{\text{fb}}(x) + u'(x)$, where $u'(x)$ is computed from a transformed QP:

$$\min_{(u',\delta)\in\mathbb{R}^{m+1}} \|u'\|^2 + p\delta^2 \tag{5.107}$$

$$s.t.\ L_{f'}V(x) + \langle L_g V(x), u'\rangle + \alpha(V(x)) \leq \delta \tag{CLF}$$

$$L_{f'}h(x) + \langle L_g h(x), u'\rangle + \beta(h(x)) \geq 0 \tag{CBFs}$$

Then, [79] demonstrates that the feedback system resulting from (2.13) with control given by $u(x) = u_{\text{nom}}(x) + u'(x)$, with $u'(x)$ given by (5.107) is such that:

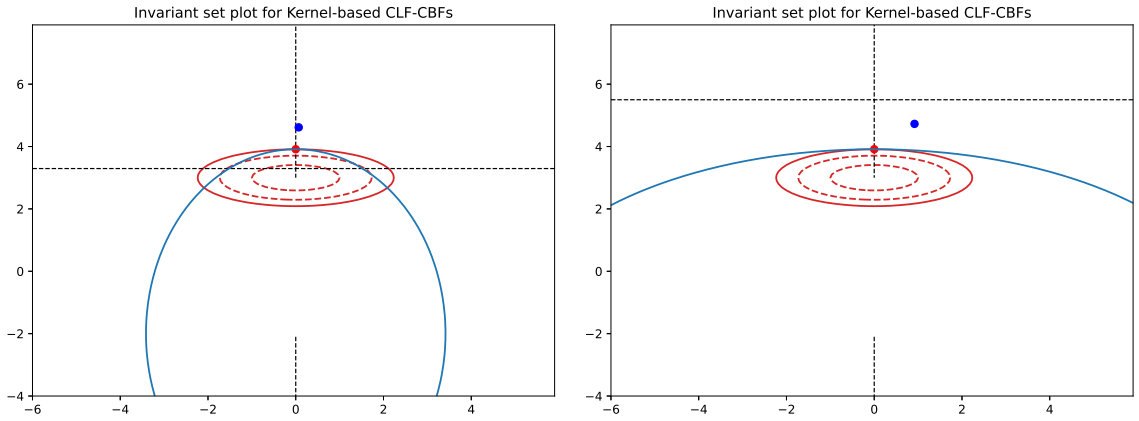
i the safe set $\mathscr{C}$ is forward invariant;

ii the CLF global minimum $x_0$ is locally asymptotically stable;

iii it has no equilibrium points with $\lambda = 0$ other than the origin, including undesired boundary equilibrium points with $L_g h(x_e) = 0$ or undesired interior equilibrium points.

While this technique effectively *removes* some of the undesirable equilibrium points from the closed-loop system, *boundary* equilibrium points (with $\lambda > 0$), could still be present and could potentially be stable. In fact, as already hinted at by some works in literature, the existence of certain classes of undesirable equilibrium points is a direct result of the topology of the problem and cannot be avoided. We will actually demonstrate this claim in Section 5.3. Furthermore, notice that the QP (5.107) is exactly the same as (2.37), with the drift term replaced by $f'(x)$ instead of $f(x)$. Therefore, all of the theoretical developments so far are still valid for the transformed system (5.106) instead of the original one, considering only a single CBF (since the analysis is exactly the same). Here, due to the similarity of analysis, we conjecture that the same result from [79] still holds in the multiple CBF case and even with the safety filter QP. Therefore, in the remainder

of this thesis, we focus on searching for solutions to the problem of undesirable stable boundary equilibrium points with $L_g h_i(x_e) \neq 0$ for the CLF-CBF QP controller.




(a) Non-compatible CLF-CBF pair.

(b) Compatible CLF-CBF pair.

Figure 5.3: Intersection between $\mathbb{A}_1$ and boundary $\partial\mathscr{C}_1$ (quadratic case).

Consider now Fig. 5.3, illustrating yet another case of boundary equilibrium formation in a driftless system with full-rank input matrix $g(x)$, where the conditions of Corollary 5.1.6.1 apply. Let $\{V, h_1\}$ be the quadratic CLF-CBF pair whose level sets are shown in Fig. 5.3. The set $\mathbb{A}_1$ is the set where the gradients $\nabla V$ and $\nabla h_1$ are co-directed, and is shown in dashed black lines. In the quadratic case, this set is expressed by quadric surfaces in the state space $\mathbb{R}^n$. Two different configurations are shown in Fig. 5.3, depending on the CLF geometry (the CBF geometry was kept constant):

(i) Fig.5.3a (left) illustrates three boundary equilibrium points (left, top and right), all in the intersection between $\mathbb{A}_1$ and the boundary $\partial\mathscr{C}_1$. The one at the top is stable, while the left and right ones are unstable.

(ii) Fig.5.3b (right), one of the eigenvalues of the CLF Hessian $H_V$ was reduced, resulting in a positive curvature difference between the CBF and CLF at the top equilibrium point (which is now unstable). Additionally, this change on the CLF geometry caused the horizontal branch of $\mathbb{A}_1$ to go up, eventually removing the extra unstable equilibrium points on the left and right. Therefore, only one equilibrium point remains at the top, and by Corollary 5.1.6.1, it is unstable.

From Theorem 5.1.6, the stability of boundary equilibrium points depends on a relation between the system dynamics, and CLF-CBF geometries. In the CLF-CBF QP-based formulation, the system to be controlled is given and since the CBFs are used to model hard safety requirements that must be fulfilled, the immediate degree of freedom for controlling the stability of the equilibrium points is the choice of the CLF geometry. This is evident from the previously shown example in Fig. 5.3, where we were able to induce a qualitative change in the number of equilibrium points of the system (a bifurcation, removing two equilibrium points) and a quantitative change in the

stability of one of the equilibrium points (changing it from stable to unstable) by only modifying the geometry of the CLF.

Therefore, considering the possibility that not all of the undesirable equilibrium points (other than $x_0$) can to be *removed*, we propose the definition of *CLF compatibility*, regarding the concept of a CLF that stabilizes the system around the origin and is completely compatible with the safety requirements (CBFs), thereby not generating any attractive equilibrium points other than the origin.

**Definition 5.1.7** (CLF Compatibility)**.** A CLF with global minimum at $0 \in \mathbb{R}^n$ is said to be *i-th compatible* if the origin is the only stable equilibrium point of the closed-loop system (2.14) with only the $i$-th CBF implemented in the QP controller (2.41).

Under Assumptions 8 and 10, it follows that the CLF minimum is quasi-globally asymptotically stable for the closed-loop system (2.14) with the QP controller (2.41) implemented with only the $i$-th CBF constraint and an $i$-th compatible CLF $V$, provided that no other types of attractors exist in the closed-loop system, such as limit cycles. Even when some boundary equilibrium points exist, since they are all unstable, the trajectories fail to converge to the origin only at a set of measure zero, which actually corresponds to the stable manifold associated to these boundary equilibrium points. Furthermore, if a CLF is compatible with all $N$ CBFs in QP (2.41), if the QP is feasible and if no other types of attractors exist in the closed-loop system, then the origin is quasi-globally asymptotically stable with the the CLF-CBF QP controller (2.41).

In the next sections, we propose two different strategies for modifying the original CLF used for stabilization on the safety-critical problem in such a way that the effective CLF used by the QP (2.41) with a single or with multiple barriers is completely compatible with all the CBFs.

## 5.2 CLF-Compatibility with Non-Radial CLFs

This section explores the possibility of modifying the CLF geometry (shape) in order to avoiding the *existence* of undesired boundary equilibrium points. We aim at the possibility of designing a stabilizing controller $u^\star(x)$ such that the resulting closed-loop system (2.14) does not contain any boundary equilibria, by modifying the original CLF in such a way that the formation of boundary equilibria is avoided.

This section will make use of some new notation conventions for the *projection* operator matrix, previously defined at Section 2.1. For a vector $v \in \mathbb{R}^n$, the *scaled orthogonal projection matrix* is defined as $\mathsf{P}_v^s = \|v\|^2 \mathsf{P}_v = \|v\|^2 I_n - vv^\mathsf{T} \in \mathbb{R}^{n\times n}$, which is a scaled version of the matrix represention for the orthogonal projection operator $\mathsf{P}_v$ defined at Section 2.1, but with usual inner product (that is, $G = I_n$) and $N_1 = I_n$. It projects any vector into the orthogonal complement of $v \in \mathbb{R}^n$, $\{v\}^\perp$. It has the following properties:

i $\mathsf{P}_v^s = (\mathsf{P}_v^s)^\mathsf{T}$ (symmetry).

ii $(\mathsf{P}_v^s)^2 = \|v\|^2 \mathsf{P}_v$, where $\mathsf{P}_v = I_n - \frac{vv^\mathsf{T}}{\|v\|^2}$.

iii The spectrum of $\mathsf{P}_v^s$ is composed of 0 and $\|v\|^2$ with algebraic multiplicity 1 and $n-1$, respectively. Therefore, $\mathsf{P}_v^s$ is positive semi-definite.

iv $\mathsf{P}_v^s z = \|v\|^2 z$ for all $z \in \{v\}^\perp$.

v $\mathsf{P}_v^s w = 0$ for all $w \in \mathbb{R}^n$ such that $v \parallel w$.

vi Considering a time-varying $v(t) \in \mathbb{R}^n$, the following holds for $\mathsf{P}_v^s(t) = \|v(t)\|^2 I_n - v(t)v(t)^\mathsf{T}$:

$$\frac{1}{2} w^\mathsf{T} \frac{d\mathsf{P}_v^s(t)}{dt} w = v(t)^\mathsf{T} \mathsf{P}_w^s \frac{dv}{dt}$$

The operator $O_n : \mathbb{R}^n \to \mathbb{R}^{n \times \frac{n}{2}(n-1)}$ is defined by $S(\omega)x = O_n(x)\omega$. For $n = 3$, using the anti-commutativity of the cross product, we have $S(\omega)x = -S(x)\omega$, and therefore $O_3(x) = -S(x)$. In general, $O_n : \mathbb{R}^n \to \mathbb{R}^{\frac{1}{2}n(n-1)\times n}$, with its components ibeing linear functions of the components of $x$. Given a the input function $g(x) = [\, g_1(x) \;\cdots\; g_m(x) \,] \in \mathbb{R}^{n\times m}$ from the nonlinear affine system (2.13) and a vector $v \in \mathbb{R}^n$, the matrix-valued function $\Gamma_{g,v} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is defined as

$$\Gamma_{g,v}(x) = \sum_{i=1}^{m} \left( g_i(x)^\mathsf{T} v I_n + g_i(x) v^\mathsf{T} \right) \nabla g_i(x)$$

**Definition 5.2.1** (Radial Function)**.** A radial function $r : \mathbb{R}^n \to \mathbb{R}$ is defined by the property $r(Qx) = r(x)$ for all $Q \in \mathbb{SO}(n)$. That is, $r$ is invariant under coordinate rotations around the origin.

### 5.2.1 Safety-Critical QP Controller with Non-radial CLFs

Here, we consider the CLF-CBF QP-based controller (2.37) with a single CBF $h : \mathbb{R}^n \to \mathbb{R}$, with safe set defined by $\mathscr{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$, as in (2.30). The theory developed in Section 5.1 can be readly applied for the single CBF case. In this section, we discuss the possibility of modifying the original CLF in such a way that the formation of boundary equilibria is avoided. Consider that a positive-definite, *non-radial* reference CLF $V_r : \mathbb{R}^n \to \mathbb{R}$ is given. Next, define an extended CLF $V : \mathbb{R}^n \times \mathbb{SO}(n) \to \mathbb{R}$ as

$$V(x, Q) = V_r(Qx) \tag{5.108}$$

where $Q \in \mathbb{SO}(n)$ represents a rotation state for the CLF. The time derivative of $Q$ is given by

$$\dot{Q} = Q S(\omega) \tag{5.109}$$

where $\omega \in \mathbb{R}^{\frac{1}{2}n(n-1)}$ is a virtual control signal with the same dimension as $\mathfrak{so}(n)$, acting as a generalized angular velocity controlling the rotation of the CLF level sets around the origin. Using (5.109), we have $\dot{V} = \nabla V^\mathsf{T} \dot{x} + \nabla_Q V^\mathsf{T} \omega$, with gradients $\nabla V(x, Q) = Q^\mathsf{T} \nabla V_r(Qx) \in \mathbb{R}^n$ and $\nabla_Q V(x, Q) = O_n(x)^\mathsf{T} \nabla V(x, Q) \in \mathbb{R}^{\frac{1}{2}n(n-1)}$.

As presented in Theorem 5.1.2, a necessary condition for the existence of boundary equilibria is collinearity among vectors $f(x)$, $G(x)\nabla V(x,Q)$ and $G(x)\nabla h(x)$. Therefore, we design a positive semi-definite function that measures the proximity of the system state from this condition:

$$D(x,Q) = \frac{1}{2}\nabla V(x,Q)^{\mathsf{T}} G\left(\mathsf{P}_f + \mathsf{P}_{G\nabla h}\right) G\nabla V(x,Q) \tag{5.110}$$

where $\mathsf{P}^s_f = \|f(x)\|^2 I_n - f(x)f(x)^{\mathsf{T}}$ and $\mathsf{P}^s_{G\nabla h} = \|G(x)\nabla h(x)\|^2 I_n - G(x)\nabla h(x)\nabla h(x)^{\mathsf{T}} G(x)$ are scaled orthogonal projection matrices for $f(x)$ and $G(x)\nabla h(x)$. Due to property (iii) of the scaled orthogonal projection, matrix $\mathsf{P}_f + \mathsf{P}_{G\nabla h}$ in (5.110) is positive semi-definite, meaning that $D(x,Q) \geq 0$.

**Assumption 11.** In the nonlinear affine system (2.13), the input matrix $g(x)$ is full rank for all $x \in \mathbb{R}^n$.

Under Assumption 11, $D(x,Q) = 0$ if and only if $f(x)$, $G(x)\nabla h(x)$ and $G(x)\nabla V(x)$ are simultaneously collinear. Using (5.109) with property (vi) of the scaled projection matrix, the time derivative of (5.110) is given by

$$\dot{D} = \nabla D(x,Q)^{\mathsf{T}}\dot{x} + \nabla_Q D(x,Q)^{\mathsf{T}}\omega \tag{5.111}$$

where the gradients $\nabla D \in \mathbb{R}^n$, $\nabla_Q D \in \mathbb{R}^{\frac{1}{2}n(n-1)}$ are given by

$$\begin{aligned}
\nabla D(x,Q) &= \left(H_V G + \Gamma^{\mathsf{T}}_{g,\nabla V}\right)\left(\mathsf{P}^s_f + \mathsf{P}^s_{G\nabla h}\right) G\nabla V + \left(H_h G + \Gamma^{\mathsf{T}}_{g,\nabla h}\right)\mathsf{P}^s_{G\nabla V} G\nabla h + \nabla f^{\mathsf{T}}\mathsf{P}^s_{G\nabla V} f \\
\nabla_Q D(x,Q) &= \left(H_V O_n - O_n(\nabla V)\right)^{\mathsf{T}} G\left(\mathsf{P}^s_f + \mathsf{P}^s_{G\nabla h}\right) G\nabla V
\end{aligned}$$

Using (5.110), define the set

$$\mathscr{C}_D = \{(x,Q) \in \mathbb{R}^n \times \mathbb{SO}(n) \mid D(x,Q) \geq \varepsilon\}$$

At a boundary equilibrium point $x_e \in \mathscr{E}_{\partial\mathscr{C}}$, we have $D(x_e,Q) = 0$. Therefore, $\mathscr{E}_{\partial\mathscr{C}} \cap \mathscr{C}_D = \emptyset$ for $\varepsilon > 0$. Aiming at avoiding the existence of equilibrium conditions imposed by $D(x,Q) = 0$ when close to the boundary $\partial\mathscr{C}$, we define the following barrier function candidate:

$$h_D(x,Q) = \sigma(h(x))\left(D(x,Q) - \varepsilon\right) \tag{5.112}$$

where $\varepsilon > 0$ is a small positive constant and $\sigma : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a smooth, strictly decreasing function with the following properties: (i) $\sigma(h) > 0$, (ii) $\sigma'(0) = 0$ and (iii) $\lim_{h\to\infty}\sigma(h) = 0$. Using the previously defined gradients of (5.110), we have $\dot{h}_D = L_f h_D + L_g h_D u + (\nabla_Q h_D)^{\mathsf{T}}\omega$ with $L_f h_D = (\nabla h_D)^{\mathsf{T}} f(x)$ and $L_g h_D = (\nabla h_D)^{\mathsf{T}} g(x)$, where the gradients with respect to $x$ and to the rotational

state $Q$ are:

$$\nabla h_D(x,Q) = \sigma(h)\nabla D + \sigma'(h)(D-\varepsilon)\nabla h$$
$$\nabla_Q h_D(x,Q) = \sigma(h)\nabla_Q D$$

Next, we propose a modification on the QP-based approach by [9] to achieve stabilization and safety for (2.14) without boundary equilibria, under certain feasibility conditions.

**Theorem 5.2.1.** Consider the nonlinear system with dynamics given by (2.13) under Assumption 11. Let $V_r(x)$ be a reference, positive definite, non-radial CLF, and $V(x,Q)$ be the CLF given by (5.108) satisfying Assumption 8, where $Q \in \mathbb{SO}(n)$ is a virtual state with dynamics given by (5.109). Additionally, consider a CBF $h(x)$ defining the safe set $\mathscr{C}$ and an auxiliary CBF $h_D(x,Q)$ given by (5.112). For any initial condition $(x_0,Q_0) \in \mathbb{R}^n \times \mathbb{SO}(n)$ with $D(x_0,Q_0) \neq 0$, the solutions $u^\star(x) \in \mathbb{R}^m$, $\omega^\star(x) \in \mathbb{R}^{\frac{1}{2}n(n-1)}$ of the QP

$$u^\star(x),\omega^\star(x),\delta^\star(x) = \underset{(u,\omega,\delta)}{\operatorname{argmin}} \|u\|^2 + p_1\|\omega\|^2 + p_2\delta^2 \tag{5.113}$$
$$s.t.\ L_fV + L_gV\,u + \nabla_Q V^\mathsf{T}\omega + \alpha(V) \leq \delta \tag{CLF}$$
$$L_fh + L_gh\,u + \beta_1(h) \geq 0 \tag{CBF1}$$
$$L_fh_D + L_gh_D u + (\nabla_Q h_D)^\mathsf{T}\omega + \beta_2(h_D) \geq 0 \tag{CBF2}$$

with $p_1,p_2 > 0$ and functions $\alpha$ of class $\mathscr{K}$ and $\beta_1,\beta_2$ of extended class $\mathscr{K}$ renders the sets $\mathscr{C}$ and $\mathscr{C}_D$ forward invariant. This guaranteees that the trajectory does not converge to boundary equilibrium points.

*Proof.* Following the proof of Proposition 1 in [7], note that for any $x \in \partial\mathscr{C}$, $\dot{h}(x) \geq -\beta_1(h(x)) = 0$. By Nagumo's theorem [11], the set $\mathscr{C}$ is forward invariant. Similarly, note that for any $(x,Q) \in \partial\mathscr{C}_D$, $\dot{h}_D(x,Q) \geq -\beta_2(h_D(x,Q)) = 0$ on the third constraint. Then, using the same argument, the set $\mathscr{C}_D$ is also forward invariant. To show that the trajectory does not converge to boundary equilibrium points, first note that the equilibrium conditions are now given by $(f_{cl}(x),\dot{Q}) = (0,0) \in \mathbb{R}^n \times \mathbb{SO}(n)$. Using (5.109), this condition is equivalent to $f_{cl}(x) = 0$ and $\omega = 0$. Next, we have to consider the possible solutions of the QP (5.113) depending on the activation of the constraints.

**CBF2 is inactive.** If the second CBF constraint (CBF2) is inactive, equilibrium happens under the same conditions of Theorem 5.1.2. Assume that the closed-loop system trajectory converges to some boundary equilibrium point $(x_e,Q_e) \in \partial\mathscr{C} \times \mathbb{SO}(n)$. Then, because of the collinearity conditions between vectors $f$, $G\nabla V$, $G\nabla h$ at $(x_e,Q_e)$, we have $D(x_e,Q_e) = 0$, which is a contradiction with the fact that $\mathscr{C}_D$ is forward invariant.

**CBF2 is active.** Consider now that the second CBF constraint (CBF2) is active, i.e., $\dot{h}_D + \beta(h_D) = 0$. Assume that the closed-loop system trajectory converges to some boundary equilibrium point $(x_e,Q_e) \in \partial\mathscr{C} \times \mathbb{SO}(n)$. Since $\dot{V} = L_{f_{cl}}V(x_e,Q_e) + \nabla_Q V(x_e,Q_e)\omega = 0$ at the equilibrium point $(x_e,Q_e)$, from the KKT conditions for (5.113), we have $\delta = p^{-1}\lambda_0$. Using the CLF constraint $\dot{V} + \gamma(V) \leq \delta$, we get $\lambda_0 \geq p\gamma(V(x_e,Q_e))$ for the first KKT multiplier $\lambda_0$. Then, since $0 \notin \partial\mathscr{C}$

and $x_e \in \partial\mathscr{C}$, we have $\lambda_0 > 0$ at the equilibrium. Therefore, the CLF constraint must be active. Additionally, we know that the first CBF constraint (CBF1) must also be active, since $\dot{h}+\alpha(h) = L_{f_{cl}}h+\alpha(h) = 0$ for $x_e \in \partial\mathscr{C}$. Using the gradient $\nabla_Q h_D$, $\dot{h}_D = L_{f_{cl}}h_D + \sigma(0)\nabla_Q D^\mathsf{T}\,\omega = 0$. Therefore, $h_D(x_e,Q_e) = \sigma(0)(D(x_e,Q_e)-\varepsilon) = 0$, and $D(x_e,Q_e) = \varepsilon$, which means that $f(x_e)$, $G(x_e)\nabla V(x_e,Q_e)$ and $G(x_e)\nabla h(x_e)$ are not simultaneously collinear. Using similar arguments than the ones used in Theorem 5.1.2, the KKT conditions for (5.113) when all constraints are active yield the following equilibrium condition:

$$f_{cl}(x_e) = f - \lambda_1 G\nabla V + \lambda_2 G\nabla h + \lambda_3\sigma(0)G\nabla D = 0\,. \tag{5.114}$$

Since $\lambda_1 > 0$ and $g(x)$ is full rank, the second term on the right-hand side of (5.114) never vanishes, which means that it must be cancelled out by at least one of the remaining terms. Since $D(x_e,Q_e) = \varepsilon > 0$, we know that $f$, $G\nabla V$ and $G\nabla h$ cannot be simultaneously collinear at $(x_e,Q_e)$. Every other possible solution of (5.114) require at least one of these three conditions: (i) $f \parallel G\nabla h$, $\nabla V \parallel \nabla D$, (ii) $f \parallel G\nabla V$, $\nabla h \parallel \nabla D$ or (iii) $\nabla h \parallel \nabla V$, $f \parallel G\nabla D$. These cases can be rewritten as: (i) $f = \kappa G\nabla h$, $\mathsf{P}^s_{\nabla V}\nabla D = 0$, (ii) $f = \kappa G\nabla V$, $\mathsf{P}^s_{\nabla h}\nabla D = 0$ or (iii) $\nabla h = \kappa\nabla V$, $\mathsf{P}^s_f G\nabla D = 0$, respectivelly, for some $\kappa \in \mathbb{R}$. Using the expressions for the gradients of $D$ and the definition of the scaled projection matrix, we conclude that in all of these cases, $f$, $G\nabla V$ and $G\nabla h$ must be simultaneously collinear, which is a contradiction with $D(x_e,Q_e) = \varepsilon > 0$. Therefore, precisely because $\mathscr{C}_D$ is forward invariant, the closed-loop system trajectory never converges to boundary equilibrium points. □

**Remark.** Notice that Theorem 5.2.1 does not imply that boundary equilibrium points do not exist in the closed-loop system; in fact, we will demonstrate in the next section that it is impossible to remove all equilibrium points for some classes of systems. It merelly guarantees that the trajectories never converge to (possibly existing) boundary equilibrium points, due to the forward invariance of set $\mathscr{C}_D$.

**Remark.** In Theorem 5.2.1, notice that if $D(x(0),Q(0)) = 0$, at an initial condition $x(0),Q(0)$, we have $\nabla_Q h_D(x(0),Q(0)) = \sigma(h)\nabla_Q D = 0$, and the QP (5.113) could possibly be unfeasible. A practical way to deal with this problem is to set $u^\star(t) = 0$ and $\omega^\star(t) = \omega_\varepsilon \approx 0$ ($\omega_\varepsilon$ is a small constant) in case $D(x(t),Q(t)) \approx 0$ for $0 < t$, in order to induce a brief initial rotation with constant velocity on the CLF level sets. Once $D(x(t),Q(t)) > 0$, the control switches to (5.113), and $D(x(t),Q(t)) > 0, \forall t > 0$ due to the forward invariance of $\mathscr{C}_D$. Notice that the influence of the third constraint on the solution becomes negligible as the trajectories stray from the boundary $\partial\mathscr{C}$, since $h_D \to 0$, $L_f h_D \to 0$, $L_g h_D \to 0$ and $\nabla_Q h_D \to 0$ as $h(x) \to \infty$.

### 5.2.2 Simulation Results

In this section, we present numerical examples of our approach for the integrator and different nonlinear systems. We use the same CLF used before as the reference CLF $V_r(x) = \frac{1}{2}\sigma_1 x_1^2 + \frac{1}{2}\sigma_2 x_2^2$, $\sigma_1 = 6$, $\sigma_2 = 1$ and CBF $h(x) = \frac{1}{2}\|x-x_c\|^2 - \frac{1}{2}r^2$, with $x_c = [\,0 \quad 3\,]^\mathsf{T}$ and $r = 1.5$. For the proposed controller on Theorem 5.2.1, we have used $p_1 = p_2 = 5$, $\alpha(V) = \alpha V$, $\beta_1(h) = \beta_2(h) = \beta h$ with

$\alpha = \beta = 1$ and $\varepsilon = 0.1$. In the integrator case, $f(x) = 0$ and $g(x) = I_n$, and (5.110) simplifies to

$$D(x,Q) = \frac{1}{2}\nabla V(x,Q)^{\mathsf{T}} \mathsf{P}^{s}_{\nabla h}(x) \nabla V(x,Q)$$

Figure 5.4 shows different system trajectories for the same initial conditions shown in Fig. 5.1. The red equilibrium point on top of the obstacle does not exist for the closed-loop system with the proposed controller, and all trajectories are attracted towards the stable origin instead. We also show the level set of the CLF for a particular state on a particular trajectory, illustrating that $V(x,Q)$ is actually a rotated version of the reference CLF $V_r(x)$. Our proposed control rotates the CLF level sets in order to avoid the trajectory to approach the set defined by $D(x,Q) = 0$.

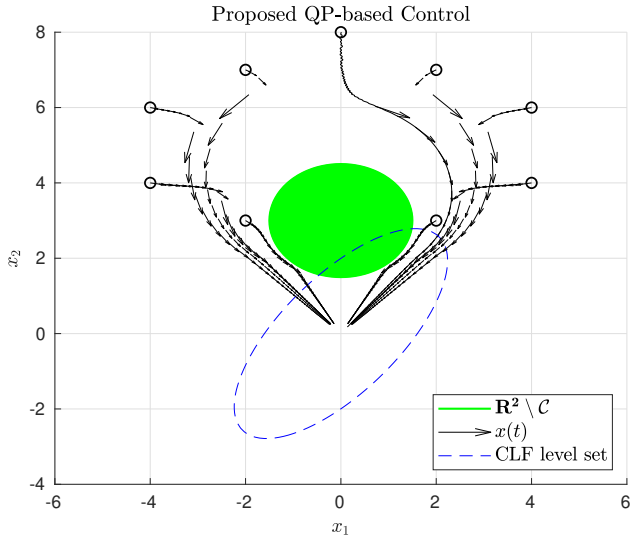



Figure 5.4: System trajectories for the closed-loop system with the proposed QP-based control for the integrator.

In Figs. 5.5 and 5.6, we show the results for two different nonlinear systems with $f(x)$ given by $f_1(x) = 0.1\|x\|[\,1 \quad 1\,]^{\mathsf{T}}$ and $f_2(x) = 0.1(\|x\| - x^{\mathsf{T}}x)[\,1 \quad 1\,]^{\mathsf{T}}$, respectivelly, both with $g(x) = I_n$. For the same initial conditions, the trajectories show the evolution of the closed-loop system state for both controllers. As before, the nominal QP-based controller proposed by [9] introduces the same undesired equilibrium point in the closed-loop system, while the trajectories for our proposed controller are attracted towards the origin. This behavior is obtained by the rotation of the CLF level sets around the origin induced by the QP-based controller (5.113).

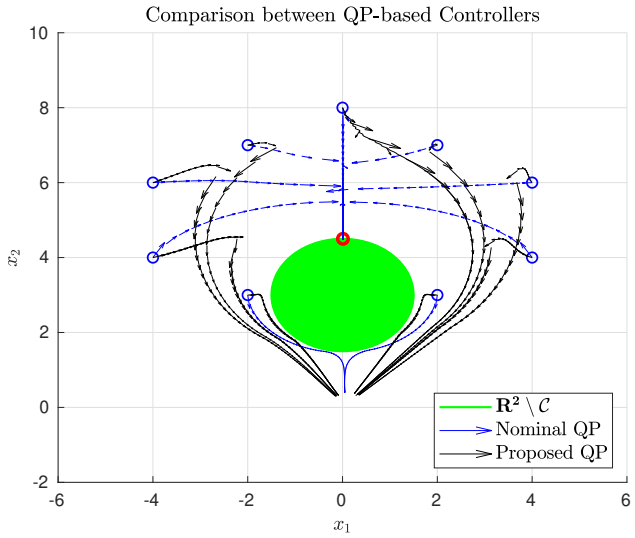



Figure 5.5: System trajectories for the closed-loop system with nominal and proposed QP-based controller for $f(x) = f_1(x)$ and $g(x) = I_n$ with a circular obstacle.

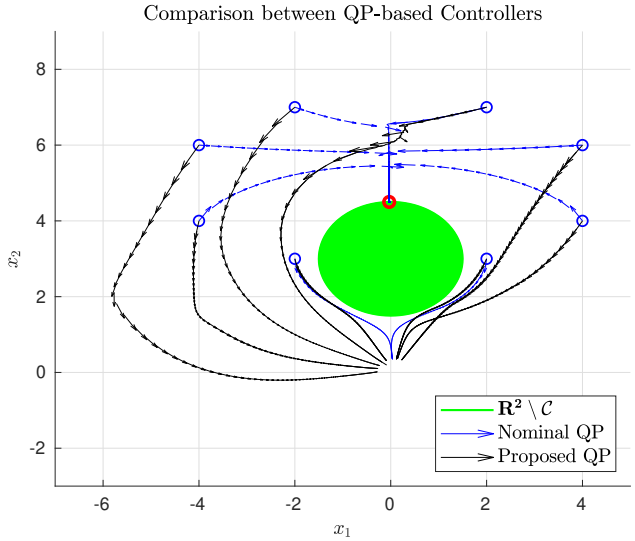



Figure 5.6: System trajectories for the closed-loop system with nominal and proposed QP-based controller $f(x) = f_2(x)$ and $g(x) = I_n$ with a circular obstacle.

## 5.3 CLF-Compatibility with Linear Matrix Polynomials

In Section 5.2, we have used a rotational degree-of-freedom on the CLF to avoid the system trajectories from approaching boundary equilibrium points. That is, we have demonstrated that it is indeed possible to modify the CLF in such a way that the equilibrium points are avoided.

However, the strategy proposed at Section 5.2 suffers from some important limitations:

i It can only be applied for nonlinear systems with a full rank input matrix $g(x)$.

ii Controller (5.113) is highly complex (due to the required computation of partial derivatives of (5.112)) and could be unfeasible even under the feasibility condition of Theorem 5.1.1, due to the extra CBF constrant for barrier $h_D$.

iii It relies on the use of a non-radial CLF to avoid the convergence of the state towards boundary equilibrium points; however, as seem from Lemma 5.3.1**(iii)**, it cannot completely *remove* all equilibrium points for every system.

iv It does not take into account the *stability* properties of the equilibrium points: if the state starts initially close to an *unstable* equilibrium point, the barrier $h_D$ rotates the CLF anyway, seeking to destroy the equilibrium collinearity conditions among vectors $f$, $G\nabla V$ and $G\nabla h$. This is highly unnecessary for CLF-compatibility: since the equilibrium point is not stable, the system trajectory would not converge towards it anyway.

Therefore, we consider the problem of CLF compatibility in a different way, considering a particular class of systems and CLF-CBFs where the existence and stabilization conditions for equilibrium can be written in a simple form, using the theory of Linear Matrix Polynomials [33]. We will demonstrate that it is in fact *impossible* to remove all undesired boundary equilibrium points for some types of systems. However, it is possible to modify the phase portrait of the closed-loop system to avoid convergence towards undesired states. We explore the possibility of modifying the CLF geometry (shape) in order to reduce the number of undesired boundary equilibrium points to a minimum in such a way that only the *unstable* ones remain (if any).

In this section, we consider the class of *quadratic* CLFs and CBFs. Let the transformed CLF $\overline{V}$ in (5.99) and the $N$ CBFs be parametrized by irreducible quadratic polynomials on $\mathbb{R}^n$, as

$$\overline{V}(x) = \frac{1}{2}\Delta x^{\mathsf{T}} H_{\overline{V}} \Delta x \geq 0\,, \qquad \Delta x = x - x_0 \tag{5.115}$$

$$h_i(x) = \frac{1}{2}\Delta x_i^{\mathsf{T}} H_{h_i} \Delta x_i - \frac{1}{2}\,, \qquad \Delta x_i = x - x_i \tag{5.116}$$

where the parameters of the CLF and of each CBF are, respectively, $\{x_0, H_{\overline{V}}\}$ and $\{x_i, H_{h_i}\}$, where (i) $x_0 \in \mathbb{R}^n$ and $x_i \in \mathbb{R}^n$, $i = 1, \cdots, N$ are the CLF and CBF centers and (ii) $H_{\overline{V}}, H_{h_i} \in \mathbb{S}^n_+$, $i = 1, \cdots, N$ are symmetric, positive definite Hessian matrices for the CLF and CBFs. Notice that the level sets of irreducible quadratic polynomials with positive definite Hessians are $n$-dimensional ellipses. Therefore, a quadratic CLF defined by (5.115) is convex. From the gradient of (5.116), the CBF boundaries $\partial\mathscr{C}_i$ are given by $\partial\mathscr{C}_i = \{x \in \mathbb{R}^n \,|\, \Delta x_i^{\mathsf{T}} H_{h_i} \Delta x_i = 1\}$, $i = 1, \cdots, N$. Given a

class $\kappa$ function $\alpha$, the original CLF $V$ can always be computed from $\overline{V}$ due to property (ii) of Proposition 5.1.3. Here, we demonstrate yet another result regarding the CLF condition and the transformed CLF $\overline{V}$.

**Proposition 5.3.1.** If Assumption 10 holds for a CLF $V(x)$, it also holds for the transformed CLF $\overline{V}$ in Definition 5.1.6.

*Proof.* If Assumption 10 holds for $V(x)$, then $L_f V = \langle \nabla V, f \rangle < 0 \ \forall x \neq 0$. Using (5.100), for all $x \neq 0$ we have $\nabla V = \alpha(V)^{-1} \nabla \overline{V}$ with $\alpha(V)^{-1} > 0$. Then, $L_f V = \alpha(V)^{-1} \langle \nabla \overline{V}, f \rangle < 0$ implies $L_f \overline{V} < 0$ for all $x \neq 0$. Then, Assumption 10 holds for $\overline{V}(x)$ as well. □

In the next two subsections, namely Sections 5.3.1 and 5.3.2, we introduce the two classes of systems that will be considered in the remainder of this section. The methods to be developed in Section 5.3.3 apply to these two classes of systems.

### 5.3.1 Linear Systems

Consider the class of linear, time invariant systems of the type

$$\dot{x} = A\Delta x + Bu\,, \qquad \Delta x = x - x_0 \tag{5.117}$$

These systems are control affine with $f(x) = A\Delta x$, $A \in \mathbb{R}^{n\times n}$ and $g(x) = B \in \mathbb{R}^{n\times m}$. Without loss of generality, we assume that $x_0$ is the CLF minimum such that $V(x_0) = 0$. Since $\overline{V}$ is quadratic and given by (5.115), for Assumption 10 to hold, it is necessary that

$$\nabla \overline{V}^{\mathsf{T}} f(x) = \Delta x^{\mathsf{T}} H_{\overline{V}} A \Delta x = \frac{1}{2} \Delta x^{\mathsf{T}} (H_{\overline{V}} A + A^{\mathsf{T}} H_{\overline{V}}) \Delta x < 0\,, \quad \forall x \neq x_0$$

which means that $H_{\overline{V}} A + A^{\mathsf{T}} H_{\overline{V}}$ must be negative definite. In case Assumption 10 does not hold, one can apply the transformation proposed by [79] if the linear system $(A,B)$ is controllable. Then, $u(x) = u_{\text{fb}}(x) + u'(x)$, where $u'(x)$ is given by (5.107) and $f'(x) = A\Delta x + Bu_{\text{fb}}(x)$. Here, since $u_{\text{fb}}(x)$ must be chosen such that $L_{f'} V(x) < 0$, it is sufficient to design a linear feedback controller $u_{\text{fb}}(x) = -K\Delta x$. Then, $f'(x) = (A - BK)\Delta x$, and therefore $K$ must be chosen such that matrix $A - BK$ is Hurwitz stable, which is always possible if the pair $(A,B)$ is controllable. Thus, the transformed system yields

$$\dot{x} = A'\Delta x + Bu'(x)\,, \quad A' = A - BK \qquad \text{(Hurwitz stable)}$$

with $H_{\overline{V}} A' + (A')^{\mathsf{T}} H_{\overline{V}} < 0$. Note that, is this case, the feedback system with $u'(x)$ given by (5.107) has no interior equilibrium points other than $x_0$ or boundary equilibrium points with the corresponding KKT multiplier $\lambda = 0$ [79]. However, general boundary equilibrium points with corresponding KKT multiplier $\lambda > 0$ could still occur, and will be addressed in the following.

In the remaining of the section, we assume that the CLF $\overline{V}$ satisfies the CLF condition (Assumption 10) for the given LTI system (that is, $A$ is a Hurwitz stable matrix) and that Assumption 9

holds. Combining the LTI system model with the CLF and CBF gradients of (5.115) and (5.116), respectively, the boundary equilibrium condition at (5.104) becomes

$$A\Delta x + \lambda BB^{\mathsf{T}} H_{h_i} \Delta x_i - pBB^{\mathsf{T}} H_{\overline{V}} \Delta x = 0\,, \qquad \lambda > 0$$

for an equilibrium point $x_e \in \partial\mathscr{C}_i$. Next, applying the transformation $\nu = \Delta x_i$ at this equation and writting the equation for the boundary $\partial\mathscr{C}_i$ in terms of this transformed state $\nu$ (as $\nu^{\mathsf{T}} H_{h_i} \nu = 1$) yields the following system of equations:

$$\begin{aligned} P(\lambda)\nu &= w\,, \quad \lambda > 0 \\ \nu^{\mathsf{T}} H_{h_i} \nu &= 1 \end{aligned}$$

where $P(\lambda) = \lambda M - N$ is a *linear matrix pencil*[2] with constant matrices $M = BB^{\mathsf{T}} H_{h_i}$ and $N = pBB^{\mathsf{T}} H_{\overline{V}} - A$ and constant vector $w = N(x_i - x_0) \in \mathbb{R}^n$.

### 5.3.2 Driftless Full rank Systems

Consider the class of driftless, affine nonlinear systems satisfying Assumption 11:

$$\dot{x} = g(x)u \tag{5.118}$$

where $g(x)$ is full-rank for all $x \in \mathbb{R}^n$. For this type of system, the transformation proposed by [79] is not required, since by Corollary 5.1.6.1, no interior equilibrium points other than the CLF minimum $x_0 \in \mathbb{R}^n$ exist. Furthermore, since $g(x)$ is full-rank, $L_g h(x) \neq 0\ \forall x \in \partial\mathscr{C}_i$, and therefore no boundary equilibrium points with $L_g h(x_e) = 0$ exist.

From (5.104), the boundary equilibrium condition is $f_i(x,\lambda) = 0,\ \lambda > 0$. Combining the driftless system dynamics with the gradients of (5.115) and (5.116), this condition becomes:

$$G(x_e)\left(\lambda H_{h_i} \Delta x_i - pH_{\overline{V}} \Delta x\right) = 0\,, \qquad \lambda > 0$$

Since $G(x_e)$ is full-rank, this equation is equivalent to $\lambda H_{h_i} \Delta x_i - pH_{\overline{V}} \Delta x = 0$. Applying the same state transformation $\nu = \Delta x_i$ and rewriting the equation for the boundary $\partial\mathscr{C}_i$ in terms of the transformed state $\nu$ yields the following system of equations:

$$\begin{aligned} P(\lambda)\nu &= w\,, \quad \lambda > 0 \\ \nu^{\mathsf{T}} H_{h_i} \nu &= 1 \end{aligned}$$

where $P(\lambda) = \lambda M - N$ is a symmetric linear matrix pencil with constant matrices $M = H_{h_i}$ and $N = pH_{\overline{V}}$ and constant vector $w = N(x_i - x_0) \in \mathbb{R}^n$.

[2] A real linear matrix pencil is a matrix operator $P : \mathbb{R} \to \mathbb{R}^{n\times n}$, where $P(\lambda) = \lambda M - N$, with $M, N \in \mathbb{R}^{n\times n}$ [46].

### 5.3.3 The Q-Function for Regular Matrix Pencils

From Sections 5.3.1 and 5.3.2, we conclude that, considering the classes of LTI and driftless full rank systems, the stability of boundary equilibrium points formed by the CLF-CBF QP controller under Assumption 9 and with quadratic CLF and CBFs can be determined by studying the solutions of the following system of equations:

$$P(\lambda)v = w, \quad \lambda > 0 \tag{5.119}$$

$$v^{\mathsf{T}} H_{h_i} v = 1 \tag{5.120}$$

where $P(\lambda) = \lambda M - N$ is a regular linear matrix pencil. Additionally, $w \in \mathbb{R}^n$ is a constant vector, $\lambda > 0$ is the corresponding KKT multiplier and $v = x - x_i$ is the transformed state vector. The solutions of (5.119) are closely tied to the theory of Linear Matrix Pencils [46] and, more generally, Linear Matrix Polynomials [33]. Let $\sigma_P = \{\lambda \in \mathbb{C} : \det P(\lambda) = 0\}$ be the spectrum of the pencil $P$. Since $P(\lambda)$ is regular, its inverse matrix $P(\lambda)^{-1}$ exists $\forall \lambda \notin \sigma_P$. Then, equation (5.119) can be solved for $v$, yielding

$$v(\lambda) = P(\lambda)^{-1} w \tag{5.121}$$

Equation (5.121) describes the equilibrium manifold $f_i(v, \lambda) = 0$ where boundary equilibrium points with $\lambda > 0$ occur in terms of the translated state $v$. Substituting it into $q(v) = v^{\mathsf{T}} H_{h_i} v$ yields

$$q(\lambda) = w^{\mathsf{T}} P(\lambda)^{-\mathsf{T}} H_{h_i} P(\lambda)^{-1} w = \frac{n(\lambda)}{|P(\lambda)|^2}, \tag{5.122}$$

$$n(\lambda) = w^{\mathsf{T}} \operatorname{adj} P(\lambda)^{\mathsf{T}} H_{h_i} \operatorname{adj} P(\lambda) w \tag{5.123}$$

where $n(\lambda)$ and $|P(\lambda)|^2$ are non-negative polynomials with coefficients in $\mathbb{R}$. Equation (5.122) is defined for the $i$-th barrier $h_i$. For the classes of systems described in Sections 5.3.1 and 5.3.2, it encodes all the necessary information for computing the equilibrium points on the $i$-th boundary. Note that, due to (5.120), every $\lambda_e \geq 0 \notin \sigma_P$ satisfying $q(\lambda_e) = 1$ corresponds to a *non-degenerate* equilibrium solution, where the corresponding boundary equilibrium point is $x_e = v(\lambda_e) + x_i = P(\lambda_e)^{-1} w + x_i \in \partial \mathscr{C}_i$.

**Lemma 5.3.1** (Q-Function Properties)**.** Under Assumptions 8-10, consider the safety-critical control problem with the two classes of systems described in Section 5.3.1 or Section 5.3.2 and its corresponding Q-function $q(\lambda)$, associated to the $i$-th CBF. Then, the following conditions hold:

i If Assumption 8 holds, then $q(0) \geq 1$.

ii If $M$ is invertible, then $\lim_{\lambda \to \pm\infty} q(\lambda) = 0$. Otherwise, if there is exactly one eigenvalue at infinity, the limit $\lim_{\lambda \to \pm\infty} q(\lambda)$ is a non-negative constant. If there are more than one eigenvalue at infinity, the limit diverges.

iii If $q(\lambda)$ is proper, the closed-loop system (2.14) has at least one boundary equilibrium point.

iv Let $R(\lambda) \in \mathbb{R}^{n\times(n-1)}$ be a matrix polynomial on the orthogonal space of $H_{h_i}\operatorname{adj}P(\lambda)w \in \mathbb{R}^n$, that is, satisfying $R(\lambda)^{\mathsf{T}}H_{h_i}\operatorname{adj}P(\lambda)w = 0\ \forall\lambda \in \mathbb{R}$. Define the *stability* matrix as

$$S(\lambda) = R(\lambda)^{\mathsf{T}}\left(P(\lambda)+P(\lambda)^{\mathsf{T}}\right)R(\lambda) \tag{5.124}$$

Then, an equilibrium point $x_e = \nu(\lambda_e)+x_i \in \partial\mathscr{C}_i$ is stable if and only if $S(\lambda_e) \leq 0$. Otherwise, it is unstable.

v There exists a scalar $\lambda_- \in \mathbb{R}$ such that $S(\lambda) \leq 0$ for all $\lambda \leq \lambda_-$. Similarly, there exists a scalar $\lambda_+ \in \mathbb{R}$ such that $S(\lambda) \geq 0$ for all $\lambda \geq \lambda_+$.

vi The maximum number of negative semi-definite intervals of $S(\lambda)$ is equal to $n$.

*Proof.* These demonstrations assume a pencil $P(\lambda) = \lambda M - N$, with $M$ and $N$ as defined in Section 5.3.1 or Section 5.3.2.

**(i)** Using the $M$ and $N$ matrices defined in Section 5.3.1 or Section 5.3.2 for the pencil $P(\lambda) = \lambda M - N$, evaluating the equilibrium condition (5.119) at $\lambda = 0$ yields $P(0)\nu = w$, yielding $\nu(0) = x_0 - x_i$. Therefore, $q(0) = (x_0-x_i)^{\mathsf{T}}H_{h_i}(x_0-x_i)$. Then, by (5.116), $q(0) > 1$ is equivalent to $h_i(x_0) > 0$, which means that $x_0 \in \mathscr{C}_i$, that is, Assumption 8 is satisfied. This proves (i).

Next, some facts about the Q-function (5.122) are presented. In (5.122), notice that the denominator $|P(\lambda)|^2$ is a positive semidefinite polynomial of max. degree $2n$. Additionally, since the entries of the adjoint matrix $\operatorname{adj}(\lambda M - N)$ are polynomials of max. degree $(n-1)$, the numerator $n(\lambda)$ in (5.123) is a sum of squares of polynomials of max. degree $(n-1)$ each, being therefore a polynomial of max. degree $(2n-2)$. Therefore, in general, $q(\lambda)$ is the ratio of a polynomial of max. degree $(2n-2)$ by a polynomial of max. degree $2n$. This proves **(i)**. □

**(ii)** If $M$ is invertible, there are no generalized eigenvalues at infinity, meaning that all the $2n$ poles of $q(\lambda)$ are finite, and therefore $q(\lambda)$ is a proper rational function. As such, as $\lambda \to \pm\infty$, the denominator $|P(\lambda)|^2$ grows faster than the numerator $n(\lambda)$, and $q(\lambda) \to 0$. An alternate way of demonstrating this fact is by noting that, as $\lambda \to \pm\infty$, $P(\lambda)^{-1} \to \lambda^{-1}M^{-1} \to 0$, since $M^{-1}$ exists. Therefore, $\nu = P(\lambda)^{-1}w \to 0$ and $q(\lambda) \to 0$. Therefore, $\lim_{\lambda\to\pm\infty} q(\lambda) = 0$. Next, assume that $M$ is singular. If there exists only one generalized eigenvalue at infinity, then two repeated poles of $q(\lambda)$ vanish and $d(\lambda)$ becomes a polynomial of degree exactly $2n-2$. In this case, $q(\lambda)$ is not proper and the degrees of $n(\lambda)$ and $|P(\lambda)|^2$ are the same. Long polynomial division implies that $q(\lambda) = C + r(\lambda)$, where $C$ is a constant and $r(\lambda)$ is a proper rational function. Then, $\lim_{\lambda\to\pm\infty} q(\lambda) = C$. Since $q(\lambda)$ is a positive semidefinite function, $C \geq 0$. Finally, if there exists two or more generalized eigenvalues at infinity, then the denominator $|P(\lambda)|^2$ becomes a polynomial of degree strictly less than $2n-2$, meaning that $q(\lambda)$ is not proper and diverges to $\infty$ as $\lambda \to \pm\infty$. This proves **(ii)**. □

**(iii)** Consider an arbitrary closed interval $\mathscr{I} \subset \mathbb{R}_+$. If $q(\lambda) > 1$ for all $\lambda \in \mathscr{I}$, then $\mathscr{I}$ does not contain equilibrium point solutions. Using (5.122), $q(\lambda) > 1 \Rightarrow n(\lambda) - |P(\lambda)|^2 > 0$ over $\mathscr{I}$. If it

was possible to guarantee this condition for the entire positive real line $\mathscr{I} = \mathbb{R}_+$, then no boundary equilibrium points would exist. However, this is impossible in general, since $q(\lambda) \geq 0$ and $\lim_{\lambda \to +\infty} q(\lambda) = 0$ in case $q(\lambda)$ is proper (which happens if the pencil $P$ has no generalized eigenvalues at infinity). For the considered problem, this demonstrates the impossibility of removing all undesirable equilibrium points for certain types of systems. This proves **(iii)**. □

**(iv)** Next, consider the translated state $\nu = x - x_i$ and the LTI and driftless systems in Sections 5.3.1 or 5.3.2. The boundary equilibrium point is given by $\nu_e = \nu(\lambda_e) \in \partial\mathscr{C}_i$ for some $\lambda_e \geq 0$ such that $q(\lambda_e) = 1$, and $\nu$ as defined in (5.121).

**Case (i).** For LTI systems, the columns of $J_{f_a}(\nu, \lambda)$ are $\partial_k f_a(\nu, \lambda) = [P(\lambda)]_k$. Using (5.119), $J_{f_a}(\nu_e, \lambda) = P(\lambda_e)$.

**Case (ii).** For driftless systems, the columns of $J_{f_a}(\nu, \lambda)$ are $\partial_k f_a(\nu, \lambda_e) = \partial_k G(\nu)(P(\lambda)\nu - w) + G(\nu)[P(\lambda)]_k$. Using (5.119), $J_{f_a}(\nu_e, \lambda_e) = G(\nu_e)P(\lambda_e)$.

Since $\nabla h_i(\nu) = H_{h_i}\nu$, using (5.121), the vector rational function $\nabla h_i(\lambda) = H_{h_i}P(\lambda)^{-1}w$ describes the barrier gradient in the equilibrium manifold consisting of states $\nu$ such that $f_a(\nu, \lambda) = 0$. At the equilibrium point $\nu_e$, we have $\nabla h_i(\nu_e) = \nabla h_i(\lambda_e)$. From Theorem 5.1.6, for any of the two cases, if there exists $v \in \{\nabla h_i(\lambda_e)\}^\perp$ such that $v^\mathsf{T} P(\lambda_e) v > 0$, then $\nu_e$ is unstable. Let $v = \mathsf{P}_{\nabla h_i}(\lambda_e) z$ with $z \in \mathbb{R}^n$, where $\mathsf{P}^s_{\nabla h_i}(\lambda) = \|\nabla h_i(\lambda)\|^2 I_n - \nabla h_i(\lambda)\nabla h_i(\lambda)^\mathsf{T}$ is a scaled orthogonal projection. Then, $v$ is a projection into $\{\nabla h_i(\nu_e)\}^\perp$. Substituting $v$ into (5.73), $\nu_e$ is unstable if $\exists z \in \mathbb{R}^n$ such that

$$\begin{aligned} z^\mathsf{T} S_{\text{null}}(\lambda_e) z > 0\,, \\ S_{\text{null}}(\lambda) = \mathsf{P}_{\nabla h_i}(\lambda)^\mathsf{T}(P(\lambda) + P(\lambda)^\mathsf{T})\mathsf{P}_{\nabla h_i}(\lambda) \end{aligned} \tag{5.125}$$

By construction, $\nabla h_i(\lambda)$ is in the nullspace of $S_{\text{null}}(\lambda)$ for all $\lambda \in \mathbb{R}$. The null-space of $\nabla h_i(\lambda)$ is the same as that of the vector polynomial $H_{h_i} \operatorname{adj} P(\lambda) w$, since they differ only by a scalar factor of $|P(\lambda)|^{-1}$. Since $|P(\lambda)|$ is of maximum degree $n$ and $P(\lambda)\operatorname{adj} P(\lambda) = |P(\lambda)| I_n$, $\operatorname{adj} P(\lambda)$ is a polynomial matrix of maximum degree $n-1$. Since $H_{h_i} \operatorname{adj} P(\lambda) w$ is a polynomial vector, it can be written in terms of its vector coefficients as

$$H_{h_i} \operatorname{adj} P(\lambda) w = v_0 + \lambda v_1 + \cdots + \lambda^l v_l\,, \qquad l \leq n-1 \tag{5.126}$$

with the $v_i$ being constant vector coefficients. Let

$$r(\lambda) = r_0 + \lambda r_1 + \cdots + \lambda^d r_d \in \mathbb{R}^n$$

be a vector polynomial of degree $d \in \mathbb{N}$ in the null-space of (5.126). Then:

$$r(\lambda)^\mathsf{T}(v_0 + \lambda v_1 + \cdots + \lambda^l v_l) = 0\,, \qquad \forall \lambda \in \mathbb{R}$$

which means that the coefficients of $r(\lambda)$ must satisfy $\sum_{i=0}^k v_i^\mathsf{T} r_{i-k} = 0$ for $k = \{0, l+d\}$. These $l+d+1$ equations can be stacked in matrix form $V\bar{r} = 0$, with $\bar{r}^\mathsf{T} = [\,r_0^\mathsf{T}\ r_1^\mathsf{T}\ \cdots\ r_d^\mathsf{T}\,] \in \mathbb{R}^{(d+1)n}$ ,

$V \in \mathbb{R}^{(l+d+1)\times(d+1)n}$ [26][Chapter XII, Section 3]. Since

$$\mathbb{R}^n = \text{span}\{H_{h_i} \text{adj}\, P(\lambda) w\} \bigoplus \{H_{h_i} \text{adj}\, P(\lambda) w\}^{\perp}$$

$\forall \lambda \in \mathbb{R}$ (that is, $\mathbb{R}^n$ is a direct sum of the two orthogonal subspaces), one can always obtain $n-1$ linearly independent basis vectors for $\{H_{h_i} \text{adj}\, P(\lambda) w\}^{\perp}$ from the vector coefficients drawn from a basis of $\mathscr{N}(V)$. Then, the arbitrary vector $z$ from (5.125) can be decomposed using the basis

$$\{\nabla h_i(\lambda_e), r_1(\lambda_e), \cdots, r_{n-1}(\lambda_e)\}$$

where $r_i(\lambda)$, $i \in \{1, \cdots, n-1\}$ are basis polynomials for $\{H_{h_i} \text{adj}\, P(\lambda) w\}^{\perp}$. Then, $z = \beta \nabla h_i(\lambda_e) + R(\lambda_e)\alpha$, where $R(\lambda) = [r_1(\lambda) \ \cdots \ r_{n-1}(\lambda)] \in \mathbb{R}^{n\times(n-1)}$ is a matrix polynomial of degree $d$ whose columns are in $\{H_{h_i} \text{adj}\, P(\lambda) w\}^{\perp}$, and $\beta \in \mathbb{R}$, $\alpha \in \mathbb{R}^{n-1}$ are the coordinates of $z$ in this basis. Then, we have $\mathsf{P}_{\nabla h_i}(\lambda) z = \|\nabla h(\lambda)\|^2 R(\lambda)\alpha$, and condition (5.125) becomes

$$z^{\mathsf{T}} S_{\text{null}}(\lambda_e) z = \|\nabla h(\lambda_e)\|^4 \alpha^{\mathsf{T}} S(\lambda_e)\alpha > 0 \tag{5.127}$$

with $S(\lambda)$ as defined in (5.124). Since $\alpha$ is now arbitrary, from Theorem 5.1.6, we conclude that the equilibrium point $v_e$ is stable if and only if $S(\lambda)$ is negative semi-definite at $\lambda_e$. Otherwise, $\exists \alpha \in \mathbb{R}^{n-1}$ such that (5.127) is strictly positive, and $v_e$ is unstable. This proves **(iv)**. □

**(v)** The polynomial matrix $S(\lambda) \in \mathbb{R}^{n-1\times n-1}$ has maximum degree 3 (odd) and its leading coefficient is positive semi-definite. That means that there must exist threshold values $\lambda_+$, $\lambda_-$ such that $S(\lambda) \geq 0$ for all $\lambda \geq \lambda_+$ and $S(\lambda) \leq 0$ for all $\lambda \leq \lambda_-$, respectively. This proves **(v)**. □

**(vi)** Furthermore, the determinant $|S(\lambda)|$ has a maximum of $3(n-1)$ real roots, which are exactly the values of $\lambda$ where the eigenvalue curves of $S(\lambda)$ change sign. That means that all $n-1$ eigenvalue curves of $S(\lambda)$ must go from negative to positive as $\lambda$ increases. Then, excluding these $n-1$ roots, a total of $2(n-1)$ roots remain, which can result in a maximum of $n-1$ negative semi-definite intervals for $S(\lambda)$ (in case all roots of $|S(\lambda)|$ are real), plus the negative semi-definite interval of infinite length $\mathscr{I}_1 = (-\infty, \sigma_-]$. Therefore, in the worst case, $n$ negative semi-definite intervals for $S(\lambda)$ exist. This proves **(iv)**. □

Lemma 5.3.1 illustrates some of the most important properties of the Q-function. The next result states the condition for CLF-compatibility in terms of the spectral properties of the stability matrix $S(\lambda)$ defined at (5.125).

**Corollary 5.3.1.1.** Under Assumptions 8-10 and considering LTI or drift-less full-rank systems, a quadratic CLF (5.115) is $i$-th compatible if and only if $S(\lambda_e)$ is not negative semi-definite at the positive real roots $\lambda_e \geq 0 \in \mathbb{R}$ of the polynomial $z(\lambda) = n(\lambda) - |P(\lambda)|^2$.

*Proof.* This result follows directly from Lemma 5.3.1. Notice that the positive roots of $z(\lambda) = n(\lambda) - |P(\lambda)|^2$ correspond to the equilibrium solutions $q(\lambda) = \frac{n(\lambda)}{|P(\lambda)|^2} = 1$. If all of them occur at the regions where $S(\lambda)$ is not negative semi-definite, then these roots correspond to unstable

boundary equilibrium points. By Assumption 10, no interior equilibrium points other then the CLF minimum exist. Then, we conclude that the CLF (5.115) is $i$-th compatible. □

Next, some numerical examples of Q-functions are provided for clarification of the analysis, considering (i) a generic driftless system and (ii) LTI systems.

#### 5.3.3.1 Driftless System: Numerical Example

Consider the following CLF and CBF in $\mathbb{R}^3$:

$$V(x) = \frac{1}{2}\sum_{j=1}^{3} \sigma_j x_j^2, \qquad h_1(x) = \frac{1}{2}\left(\|x - x_1\|^2 - 1\right)$$

The CLF is a quadratic form with parameters $\sigma_1$, $\sigma_2$ and $\sigma_3$. Its level sets are paraboloids in three dimensions aligned with the main axis of $\mathbb{R}^n$ and centered on the origin, while the unsafe set defined by the CBF defines an open ball of radius $r > 0$ centered on $x_1^\mathsf{T} = [\, x_{1_1} \; x_{1_2} \; x_{1_3} \,]^\mathsf{T}$. In this particular example, we use the values $\sigma_1 = 1.0$, $\sigma_2 = 3.0$, $\sigma_3 = 5.0$, $r = 1$, $x_1^\mathsf{T} = [\, 1.0 \; 1.0 \; 0.5 \,]^\mathsf{T}$. Notice that due to the circular shape of the obstacle, the CBF Hessian is simply $H_{h_1} = I_3$.

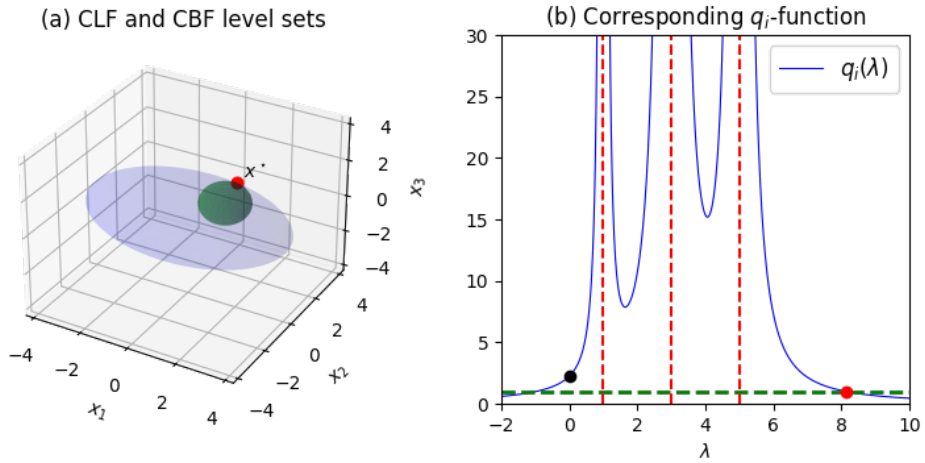



Figure 5.7: (i) Level sets of a quadratic CLF-CBF pair on $\mathbb{R}^3$ and (ii) corresponding Q-function, with 3 pencil eigenvalues of positive type (vertical dashed lines). An equilibrium point of the closed-loop system is shown as the red point in (i), corresponding to a solution of (5.119)-(5.120), that is, $q(\lambda) = 1$. This solution is also shown in red in (ii).

Figure 5.7(a) shows a CBF boundary (in green) modeling a circular obstacle and the geometry of the CLF level set (in blue) on a particular equilibrium point (in red) of the closed-loop system formed by a general driftless nonlinear system with controller (2.41). Notice that the gradients $\nabla V(x_e)$ and $\nabla h(x_e)$ are positively collinear on the boundary equilibrium point $x_e \in \mathscr{C}_1$, defining the orthogonal complement $\{\nabla V(x_e)\}^\perp \equiv \{\nabla h(x_e)\}^\perp$. This set is also a tangent plane to the CLF and CBF level sets $V(x) = V(x_e)$ and $h_1(x) = h_1(x_e)$, respectively. Geometrically, the differences of sectional curvatures $\Delta\kappa(v, x_e) = \kappa_{h_i}(v, x_e) - \kappa_V(v, x_e)$ on every direction $v \in \{\nabla V(x_e)\}^\perp \equiv \{\nabla h(x_e)\}^\perp$ are positive, which by Corollary 5.1.6.1 implies that $x_e$ is unstable.

For the considered driftless system, (5.119) yields

$$\underbrace{\begin{bmatrix} \lambda - r^2\sigma_1 & 0 & 0 \\ 0 & \lambda - r^2\sigma_2 & 0 \\ 0 & 0 & \lambda - r^2\sigma_3 \end{bmatrix}}_{P(\lambda)} \underbrace{\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}}_{v} = r^2 \underbrace{\begin{bmatrix} \sigma_1 x_{1_1} \\ \sigma_2 x_{1_2} \\ \sigma_3 x_{1_3} \end{bmatrix}}_{w}, \tag{5.128}$$

from which one can compute all equilibrium points. The pencil spectra of generalized eigenvalues is $\sigma_P = \{r^2\sigma_1, r^2\sigma_2, r^2\sigma_3\} = \{1,3,5\}$. Furthermore, the corresponding Q-function can be explicitly computed as

$$q(\lambda) = \frac{\|v(\lambda)\|^2}{r^2} = \sum_{j=1}^{3} \frac{(r\sigma_j x_{1_j})^2}{(\lambda - r^2\sigma_j)^2}$$

Figure 5.7(b) illustrates the graph of this Q-function for the given parameters of this example. The three pencil eigenvalues are represented by the vertical red dashed lines. These eigenvalues create a partition of the domain of $q_1(\lambda)$, dividing it into $n+1$ open intervals: $\mathscr{I}_1 = (-\infty, 1)$, $\mathscr{I}_2 = (1,3)$, $\mathscr{I}_3 = (3,5)$ and $\mathscr{I}_4 = (5,+\infty)$.

Notice that $q(0) = \frac{1}{r^2}\|x_1\|^2 > 1$. By Lemma 5.3.1**(i)**, this condition implies that the origin is contained in the safe set $\mathscr{C}_1$. Using the Q-function expression, this condition is equivalent to $\|x_1\| > r$, which geometrically means that the origin is not contained inside the ball of radius $r$ defining the obstacle boundary. From Fig. 5.7(b), notice that $q(0) > 1$ (black point), implying that in fact the obstacle does not enclose the origin for the given parameters of this example.

In general, the boundary equilibrium points are computed from $x_e = v(\lambda_e) + x_1$, where $\lambda_e \geq 0$ are the positive solutions of $q(\lambda_e) = 1$. Since $n = 2$ in this example, these solutions can be computed by solving the roots of the $6n$-degree polynomial given by $z(\lambda) = n(\lambda) - |P(\lambda)|^2 = \|v(\lambda)\|^2 - r^2$, where $v(\lambda) = P(\lambda)^{-1}w$. Notice that, since $q(\lambda) > 1\ \forall \lambda \in \mathbb{R}_{\geq 0} \cap (\mathscr{I}_1 \cup \mathscr{I}_2 \cup \mathscr{I}_3)$, no equilibrium solutions $\lambda_e$ exist with $0 \leq \lambda_e \leq 5$. In Fig. 5.7(b), only one solution exists (red point) at the interval $\mathscr{I}_4$.

In this example, $S(\lambda)$ is a $2 \times 2$ matrix polynomial, with $S(\lambda) > 0\ \forall \lambda \in \mathscr{I}_4$ and $S(\lambda) \leq 0\ \forall \lambda \in \mathbb{R}_{\geq 0} \setminus \mathscr{I}_4$, which means, by Lemma 5.3.1**(iv)**, that the corresponding equilibrium point $x_e = v(\lambda_e) + x_i$ is unstable. This is precisely the equilibrium point showed in Fig. 5.7(a), in red. Therefore, according to Corollary 5.3.1.1, the origin is the only stable equilibrium point of the closed-loop system resulting from the QP controller (2.41) the quadratic CLF-CBF pair given in this example. Therefore, $V$ is *compatible* with barrier $h_1$.

#### 5.3.3.2 LTI Systems: Numerical Examples

**Example 1.** Consider the two-dimensional LTI system $\dot{x} = -2x + u$. Figure 5.8 shows the graphs of the corresponding Q-function $q(\lambda)$ and zero polynomial $z(\lambda) = n(\lambda) - |P(\lambda)|^2$ for the given LTI system, with quadratic CLF $V$ and CBF $h_1$ given by (5.115)-(5.116), respectively, with Hessian

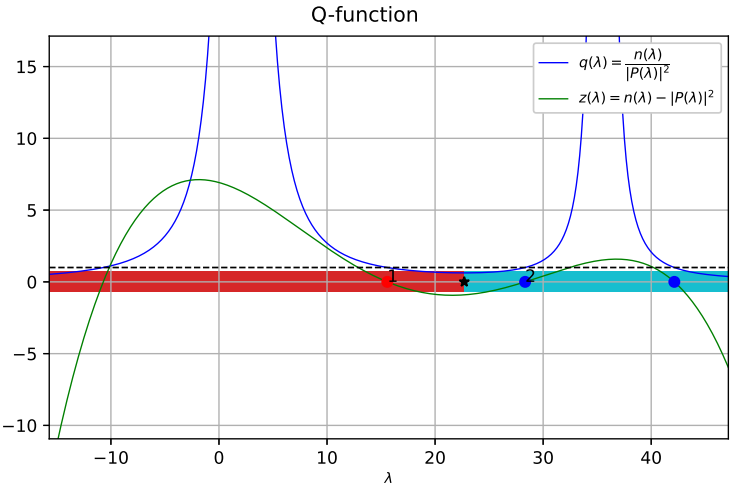



Figure 5.8: Example of a Q-function for the LTI system $\dot{x} = -2x + u$ in two dimensions.

matrices given by

$$H_{\overline{V}} = \begin{bmatrix} 1.95 & 0.26 \\ 0.26 & 0.54 \end{bmatrix}, \quad H_{h_1} = \begin{bmatrix} 0.14 & -0.16 \\ -0.16 & 1.99 \end{bmatrix}.$$

The CLF is centered at the origin $x_0 = [\,0 \;\; 0\,]^\mathsf{T}$, while the CBF is centered at $x_1 = [\,0 \;\; 5\,]^\mathsf{T}$. The asymptotes of $q(\lambda)$ occur at the three generalized eigenvalues of the pencil $P$, which are all real in this example. In this example, $q(0) \geq 1$, or equivalently $z(0) \geq 0$. From Lemma 5.3.1**(i)**, this implies Assumption 8, that is, $x_0 \in \mathscr{C}$, which is indeed true from the CBF expression, since $h(0) \geq 0$. The stability matrix $S(\lambda)$ is here a $1 \times 1$ matrix (that is, a scalar polynomial) of degree 3 with real coefficients. Therefore, $|S(\lambda)|$ has three roots in $\mathbb{C}$. Lemma 5.3.1**(v)** implies that a maximum of $n = 2$ negative semi-definite intervals can occur for $S(\lambda)$. In this example, $S(\lambda) \leq 0$ in the interval $\approx (-\infty, 23)$ (red strip) and $S(\lambda) \geq 0$ in the interval $\approx (23, +\infty)$ (blue strip). These intervals are separated by the real root of $|S(\lambda)|$ at $\lambda \approx 23$ (star-shaped dot); the remaining two roots of $|S(\lambda)|$ correspond to a complex-conjugate pair. There is one solution $z(\lambda_e) = 0$ around $\lambda_e \approx 16$ corresponding to a stable equilibrium point (red dot) and other two around $\lambda_e \approx 28, 42$ corresponding to unstable equilibrium points (blue dots). From Corollary 5.3.1.1, the CLF is not compatible with $h_1$.

**Example 2.** Consider the three-dimensional LTI system $\dot{x} = -x + u$. Figure 5.9 shows the graphs of the corresponding Q-function $q(\lambda)$ and zero polynomial $z(\lambda) = n(\lambda) - |P(\lambda)|^2$ for this example LTI system, with quadratic CLF $V$ and CBF $h_1$ given by (5.115)-(5.116), respectively, with Hessians given by

$$H_{\overline{V}} = \begin{bmatrix} 2.0 & 0 & 0 \\ 0 & 0.125 & 0 \\ 0 & 0 & 0.50 \end{bmatrix}, \quad H_{h_1} = \begin{bmatrix} 0.131 & -0.047 & 0.008 \\ -0.047 & 0.517 & -0.183 \\ 0.008 & -0.183 & 1.977 \end{bmatrix}.$$

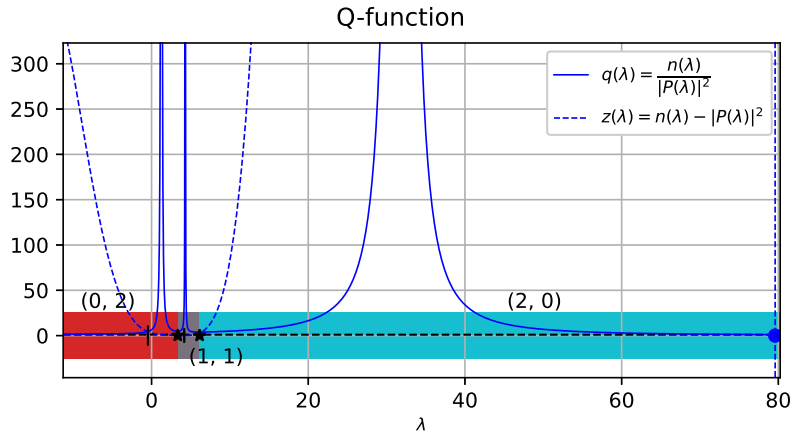



Figure 5.9: Example of a Q-function for the LTI system $\dot{x} = -x + u$ in three dimensions.

The CLF is centered at the origin $x_0 = [\,0\ 0\ 0\,]^\mathsf{T}$, while the CBF is centered at $x_1 = [\,6\ 0\ 2\,]^\mathsf{T}$. The asymptotes of $q(\lambda)$ occur at the three generalized eigenvalues of the pencil $P$, which are also real in this example. In this example, again $q(0) \geq 1$, or equivalently $z(0) \geq 0$. From Lemma 5.3.1**(i)**, this implies Assumption 8, that is, $x_0 \in \mathscr{C}$, which is indeed true from the CBF expression, since $h(0) \geq 0$. In three dimensions, the stability matrix polynomial $S(\lambda)$ is $2 \times 2$ of degree 3 with real coefficients. Therefore, $|S(\lambda)|$ has six roots in $\mathbb{C}$. Lemma 5.3.1**(v)** implies that a maximum of $n = 3$ negative semi-definite intervals can occur for $S(\lambda)$. In this example, $S(\lambda) \leq 0$ in the interval $(-\infty, 3.33)$ (red strip) and $S(\lambda) \geq 0$ in the interval $(6.13, +\infty)$ (blue strip), being *indefinite* in the interval $[3.33, 6.13]$. These intervals are separated by the real roots of $|S(\lambda)|$ at $\lambda \approx 3.33$ and $\lambda \approx 6.13$ (star-shaped dots); the remaining four roots of $|S(\lambda)|$ correspond to two complex-conjugate pairs. There is only one equilibrium solution $z(\lambda_e) = 0$ around $\lambda_e \approx 79.6$ corresponding to an unstable equilibrium point (blue dot). From Corollary 5.3.1.1, the CLF is compatible with $h_1$.

### 5.3.4 Safety-Critical QP Controller with Compatible CLF-CBFs

From the two previous examples and from Corollary 5.3.1.1, it is clear that CLF-compatibility can be achieved for quadratic CLF-CBF pairs and driftless / LTI systems in case it is possible to control some degree of freedom of the system in such a way that no equilibrium solutions $q(\lambda_e) = 1$ (or, equivalently, $z(\lambda_e) = 0$) exist in the regions of $\mathbb{R}_{\geq 0}$ where the stability matrix $S(\lambda)$ is negative definite. By the end of Section 5.1.3, we hinted at the possibility of using the CLF shape as this extra degree-of-freedom. We have shown that it is indeed possible to modify the CLF in such a way that the equilibrium points are avoided, by introducing *rotating* CLFs in Section 5.2. In this section we develop yet another strategy to modify the CLF in order to achieve compatibility. Backed by Corollary 5.3.1.1, our objective is twofold:

i We aim to design a "compatibilization" algorithm for computing a compatible CLF from a non-compatible one.

ii We aim to develop a control strategy to smoothly transform the CLF used in the QP-controller (2.37) towards the compatible CLF computed from the compatibilization algorithm, in the regions of the state space where boundary equilibrium points occur.

We start by showing how CLF-compatibility can be tested for a corresponding Q-function by solving an optimization problem.

**Lemma 5.3.2** (Compatibility Barrier)**.** Under Assumptions 8-10, consider the Q-function $q(\lambda) = \frac{n(\lambda)}{|P(\lambda)|^2}$ associated to the $i$-th CBF, where $n(\lambda)$ is given by (5.123) and $P(\lambda) = \lambda M - N$ is the linear matrix pencil with matrices $M, N$ defined in Sections 5.3.1 or 5.3.2. Let $\mathscr{I} = (-\infty, \sigma]$ be the first negative semi-definite interval of the linear matrix polynomial $S(\lambda)$ defined in (5.124), that is, $S(\lambda) \leq 0 \; \forall \lambda \leq \sigma$, and define the functional

$$B(q) = \min_{\lambda \in \mathscr{I} \cap \mathbb{R}_{\geq 0}} n(\lambda) - \varepsilon |P(\lambda)|^2 \tag{5.129}$$

where $\varepsilon > 1$. Then, the CLF (5.115) is $i$-th compatible if $B(q) \geq 0$ and $S'(\lambda) \geq 0$.

*Proof.* Under Assumption 8-9, if $B(q)$ is non-negative in (5.129), there are positive roots of $z_\varepsilon(\lambda) = n(\lambda) - \varepsilon |P(\lambda)|^2$ in $\mathscr{I}$. Then, $q(\lambda) = \frac{n(\lambda)}{|P(\lambda)|^2} \geq \varepsilon > 1$ in $\mathscr{I} \cap \mathbb{R}_{\geq 0}$, which means that no boundary equilibrium solutions of (5.119)-(5.120) exist in $\mathscr{I}$. The monotonicity constraint $S'(\lambda) \geq 0$ ensures that the eigenvalues of $S(\lambda)$ are *monotonically increasing*, which is a sufficient condition to ensure that no negative semi-definite interval of $S(\lambda)$ other than $\mathscr{I}$ exists, and therefore any possible equilibrium solution of (5.119)-(5.120) occurring at $\mathbb{R}_{\geq 0} \setminus \mathscr{I}$ must correspond only to *unstable* equilibrium points. Furthermore, under Assumption 10, no interior equilibrium points other than $x_0$ exist. This shows that if $B(q) \geq 0$ and $S'(\lambda) \geq 0$, then no stable equilibrium other than $x_0$ exists. Thus, the CLF is $i$-th compatible. □

As a functional of the Q-function associated to the $i$-th CBF, the barrier function $B(q)$ in (5.129) is dependent on the system dynamics, CLF and $i$-th CBF geometry. Lemma 5.3.2 presents a sufficient, although not necessary condition for CLF $i$-th compatibility. It will be useful for the fulfillment of the first objective of this section: the derivation of a "compatibilization" algorithm for computing a compatible CLF from a non-compatible one.

**Definition 5.3.1.** Under Assumption 8, Hessian $H$ is $i$-th compatible if its corresponding CLF $V(x) = \frac{1}{2}\Delta x^\top H \Delta x$ is $i$-th compatible.

Under Assumption 8, let $\overline{V}_r(x) = \frac{1}{2}\Delta x^\top H_{\overline{V}_r} \Delta x$ be a reference quadratic CLF centered on $x_0 \in \mathscr{C}$ and define the following optimization problem related to the $i$-th CBF:

$$\begin{aligned} H_{\overline{V}_i} = \underset{H \in \mathbb{S}^n_+}{\operatorname{argmin}} \; & \|H - H_{\overline{V}_r}\|^2 && (5.130)\\ \text{s.t. } & HA + A^\top H \leq 0 && \text{(CLF condition)}\\ & B(q) \geq 0 && \text{(compatibility)}\\ & S'(\lambda) \geq 0 && \text{(monotonicity)} \end{aligned}$$

The result of optimization (5.130) is the Hessian $H_{\overline{V}_i}$ of the closest quadratic $\overline{V}_i(x) = \frac{1}{2}\Delta x^\top H_{\overline{V}_i}\Delta x$ to the reference CLF $\overline{V}_r$ satisfying the following conditions:

i $\overline{V}_i$ is a valid CLF since it satisfies the CLF condition $L_f\overline{V}_i \leq 0$ for the given LTI system: $L_f\overline{V}_i \leq 0 \implies H_{\overline{V}_i}A + A^\top H_{\overline{V}_i} \leq 0$. For driftless systems, the CLF condition is always satisfied since $f(x) = 0$.

ii $\overline{V}_i$ is $i$-th compatible (due to Theorem 5.3.2). Likewise, $H_{\overline{V}_i}$ is $i$-th compatible.

In a way, optimization (5.130) acts as a *compatibility* filter for the CLF Hessian: it tries to compute the closest compatible Hessian to the reference $H_{\overline{V}_r}$. Thus, if the reference $H_{\overline{V}_r}$ is already $i$-th compatible, the result of (5.130) is simply $H_{\overline{V}_i} = H_{\overline{V}_r}$. Furthermore, in (5.130), the barrier $B(q)$ depends on the polynomials $n(\lambda)$, $|P(\lambda)|^2$ (the Q-function numerator and denominator) and the stability matrix polynomial $S(\lambda)$, which can be efficiently computed using methods for polynomial computation. These polynomials depend on the system and on the CLF-CBF parameters. Therefore, the functional $B(q)$ is dependent on the optimization variable $H$. That means that the solver for optimization (5.130) must recompute the required polynomials at each iteration until convergence is achieved. Any solver supporting non-convex constrained optimization could be used, such as Sequential Least Squares Programming (SLSQP) [12].

**Theorem 5.3.3.** Consider the LTI system (5.117) or the driftless nonlinear system (5.118), with controller (2.41) with only one quadratic CBF $h_i$ given by (5.116) and a CLF $V_i$ given by the inverse transformation of a quadratic CLF $\overline{V}_i$ given by (5.115), with Hessian $H_{\overline{V}_i}$ obtained from the optimization (5.130). Then, the origin is the only stable equilibrium point of the closed-loop system (2.14).

*Proof.* The proof follows from Lemma 5.3.2 and the properties of the optimization (5.130). For linear systems, due to the fact that $H_{\overline{V}_i}A + A^\top H_{\overline{V}_i} \leq 0$ holds for the solution $H_{\overline{V}_i}$ of (5.130), $\overline{V}_i$ satisfies the CLF condition. For driftless systems, $\overline{V}_i$ also trivially satisfies the CLF condition since $f(x) = 0$. Due to the fact that the solution $H_{\overline{V}_i}$ defines a CLF $\overline{V}_i$ satisfying the constraints $B(q) \geq 0$ and $S'(\lambda) \geq 0$, using Lemma 5.3.2, $\overline{V}_i$ is $i$-th compatible. Finally, due to property (iii) of Proposition 5.1.3, since the CLF $V_i$ obtained from the compatible CLF $\overline{V}_i$ using the inverse transformation (5.99) has the same level sets as $\overline{V}_i$, it must be $i$-th compatible as well. Then, it follows that the origin is the only stable equilibrium point of the closed-loop system (2.14). □

Theorem 5.3.3 means that the optimization (5.130) consists of a process of producing CLFs that are compatible with a given CBF $h_i$, which can then be used in the CLF-CBF QP-based controller (2.41), effectively guiding the system trajectories towards the CLF minimum (goal) without converging to undesirable attractors. Next, we propose an algorithm for dynamically modifying the Hessian of the stabilizing CLF in such a way that it is rendered compatible with a given CBF, at particular regions of the state-space. Consider the $N$ quadratic CBFs $\{h_1, \cdots, h_N\}$, each given by (5.116), and let $\{H_{\overline{V}_1}, \cdots, H_{\overline{V}_N}\}$ be a set of $N$ corresponding compatible Hessians (according to Definition 5.3.1) computed using the optimization (5.130), where $H_{\overline{V}_i}$ is the closest

$i$-th compatible Hessian to the reference $H_{\overline{V}_r}$. Define a parametric CLF $\overline{V}$ with a parametrized Hessian as

$$\overline{V}(x,\pi) = \frac{1}{2}\Delta x^\top \underbrace{L(\pi)^\top L(\pi)}_{H_{\overline{V}}(\pi)} \Delta x = \frac{1}{2}\|L(\pi)\Delta x\|^2 \tag{5.131}$$

where $\pi \in \mathbb{R}^{\dim \mathbb{S}^n}$ is a $\frac{1}{2}n(n+1)$-dimensional state vector defining the geometry of the level sets of $\overline{V}$ (also known as the CLF shape state) and $L : \mathbb{R}^{\frac{1}{2}n(n+1)} \to \mathbb{S}^n$ is a map to the vector space of symmetric matrices. For convenience, we also define the inverse map $L^{-1} : \mathbb{S}^n \to \mathbb{R}^{\frac{1}{2}n(n+1)}$, converting a symmetric matrix to its corresponding vector of flattened components.

We seek to design a controller for the CLF state $\pi$, so that the level sets of $\overline{V}$ are dynamically changed. Let the dynamics of $\pi$ be given by an integrator $\dot{\pi} = u_\pi$. Then, a simple proportional feedback controller can be used to stabilize $\pi$ around a time-varying reference $\pi_{\text{ref}}(t)$:

$$u_\pi = -\kappa_\pi(\pi - \pi_{\text{ref}}(t)), \quad \kappa_\pi > 0$$

The reference $\pi_{\text{ref}}(t)$ corresponds to the components of a reference Hessian matrix toward which the CLF Hessian is intended to converge.

Now, consider the QP controller (2.37) with the CLF $V$ computed from the inverse CLF transformation of $\overline{V}(x,\pi)$ (Definition 5.1.6). This transformation is always guaranteed to exist by Proposition 5.1.3(ii). By Theorem 5.1.2, under Assumption 9, all of the $i$-th boundary equilibrium points are contained in the set

$$\Omega_i^{clf} = \{x \in \mathbb{R}^n \,|\, f_{cl}(x) = f_{\text{nom}}(x) - \lambda_0 G(x)\nabla V(x) + G(x)\nabla h_i(x)\lambda_i\} \tag{5.132}$$

with $\lambda_0, \lambda_i > 0$ given by the corresponding KKT multipliers. That is, (5.132) is the set where the CLF and only the $i$-th CBF constraint are active; a particular version of the (5.39) but with only one active CBF constraint. If the trajectory of the closed-loop system (2.14) approaches an asymptotically stable equilibrium point on the $i$-th boundary, then the state eventually enters $\Omega_i^{clf}$. Then, the following strategy is considered:

i If the state is inside $\Omega_i^{clf}$, $\overline{V}$ must converge to $\overline{V}_i$, the closest $i$-th compatible CLF to $\overline{V}_r$, since this will induce a bifurcation on the closed-loop system state space, either removing or rendering the $i$-th boundary equilibrium points unstable. This can be achieved by setting $\pi_{\text{ref}} = L^{-1}(\sqrt{H_{\overline{V}_i}})$, where $\sqrt{H_{\overline{V}_i}}$ is the matrix square root of the $i$-th compatible CLF Hessian $H_{\overline{V}_i}$.

ii If the state is outside $\cup_i^N \Omega_i^{clf}$ (a deadlock free region of the state space, since no boundary equilibrium points can exist there), $\overline{V}$ must converge to the reference CLF $\overline{V}_r$. Therefore, if the state is in a deadlock-free region, its evolution is governed by the reference CLF. This can be achieved by setting $\pi_{\text{ref}} = L^{-1}(\sqrt{H_{\overline{V}_r}})$, where $\sqrt{H_{\overline{V}_r}}$ is the matrix square root of the reference CLF Hessian $H_{\overline{V}_r}$.

With this strategy, the CLF shape changes when the state enters a new region $\Omega_i^{clf}$ (for some $i \in 1,\cdots,N$) or leaves $\cup_i^N \Omega_i^{clf}$, and $\overline{V}$ is rendered compatible with the CBF of the corresponding region where it is active. This desired effect can be achieved by the following controller for the CLF shape state $\pi \in \mathbb{R}^{\dim \mathbb{S}^n}$:

$$u_\pi = -\kappa_\pi(\pi - \pi_{\text{ref}}(t)), \quad \kappa_\pi > 0 \tag{5.133}$$
$$\pi_{\text{ref}}(t) = \begin{cases} L^{-1}(\sqrt{H_{\overline{V}_i}}), & \text{if } x(t) \in \mathbb{S}_i \\ L^{-1}(\sqrt{H_{\overline{V}_r}}), & \text{otherwise} \end{cases}$$

with set $\Omega_i^{clf}$ as in (5.132) and reference CLF Hessians drawn from $\{H_{\overline{V}_r}, H_{\overline{V}_1}, \cdots, H_{\overline{V}_N}\}$, depending on the region of the state space where the closed-loop state is located. That is, with (5.133), the CLF shape state $\pi$ is controlled to achieve $\overline{V} \to \overline{V}_i$ when $x \in \Omega_i^{clf}$, and $\overline{V} \to \overline{V}_r$ otherwise.

**Remark.** The strategy proposed in (5.133) effectively controls the curvature of the CLF level sets in order to achieve CLF compatibility with respect to the $i$-th active barrier. Moreover, while this strategy can make the system trajectories converge towards the CLF minimum, the occurrence of other types of attractors such as limit cycles is not theoretically discarded.

### 5.3.5 Simulation Results

In this section we present numerical examples demonstrating the viability of the proposed method[3].

**Example 1.** Consider again the two-dimensional LTI system $\dot{x} = -2x + u$, whose Q-function was shown in Fig. 5.8 for a given quadratic CLF $\overline{V}_r$ and CBF $h_1$. This CLF satisfies Assumption 10, and therefore no interior equilibrium points other than the origin exist. The with unsafe set of $h_1$ is shown in Fig. 5.10 as the red ellipse. The first row of Fig. 5.10 shows the results obtained using the nominal QP controller (2.41) with a fixed reference CLF $\overline{V}_r$ (level set at the state $x(t)$ are the blue ellipses) and a single CBF constraints for $h_1$. As expected, the trajectories converge towards a stable equilibrium point at the boundary $\partial\mathscr{C}_1$ (red dot), on top of the elliptical obstacle. Two other unstable boundary equilibria also exist at $\partial\mathscr{C}_1$ (blue dots). Each of the remaining CBFs only have one unstable boundary equilibrium. Therefore, $\overline{V}_r$ is not compatible with $h_1$.

The closest CLF to the reference $H_{\overline{V}_r}$ that is compatible with the CBF $h_1$ was computed using the optimization (5.130), producing $H_{\overline{V}_1}$. This Hessian defines an CLF $\overline{V}_1$ whose level sets are rotated ans scaled versions of the level sets of $\overline{V}_r$, in such a way that the stable boundary equilibrium points are removed.

The second row of Fig. 5.10 shows the results obtained using the QP controller (2.41) with a CLF $V$ obtained from the inverse transformation of 5.99 of the quadratic CLF $\overline{V}(x,\pi)$. In this example, we used the linear class $\mathscr{K}$ function $\alpha(V) = \alpha V$, $\alpha > 0$. Solving (5.99), the transformed CLF is simply $\overline{V}(x,\pi) = \frac{1}{2}\alpha V^2$, and the inverse transformation is $V = \sqrt{\frac{2}{\alpha}\overline{V}(x,\pi)}$. The Hessian

[3]The code repository used for producing the results of this section is publicly available at https://github.com/CaipirUltron/CompatibleCLFCBF/tree/mydevel

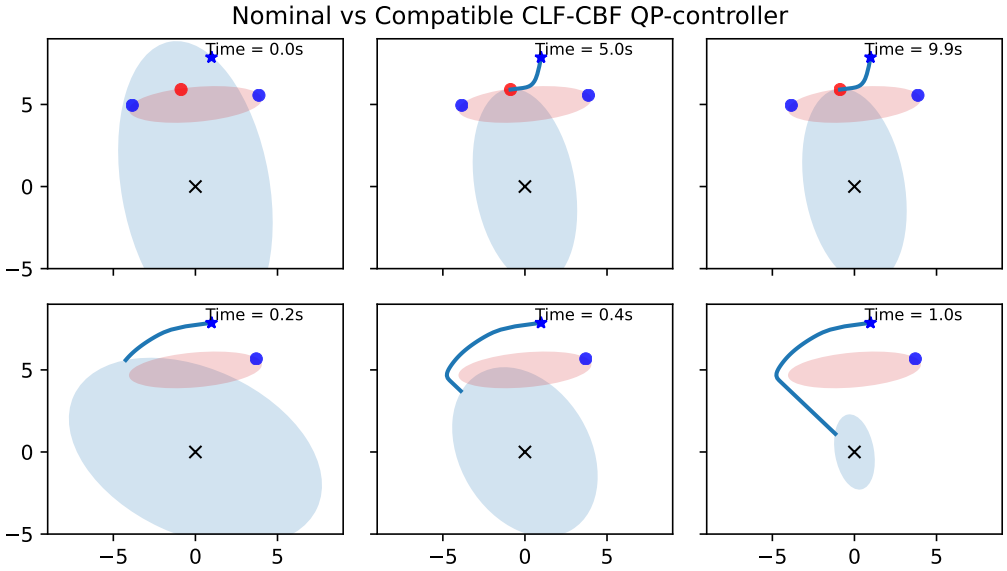



Figure 5.10: Compatible QP controller: fixed CLF vs adaptive strategy with a single obstacle.

$H_{\overline{V}}(\pi)$ is controlled by our proposed strategy using (5.133). From the timestamps, the level sets of $\overline{V}$ dynamically change to match those of $\overline{V}_1$ when $x(t) \in \Omega_1^{clf}$, inducing a bifurcation on the phase portrait of the feedback system that removes the stable point (and one of the unstable points as well). Only one stable point remains at the boundary $\partial\mathscr{C}_1$. The system trajectories converge towards the origin for all tested initial conditions, and the level sets of $\overline{V}$ converge back to match those of the reference $\overline{V}_r$ after the state leaves $\Omega_1^{clf}$, as seen from the second row of Fig. 5.10.

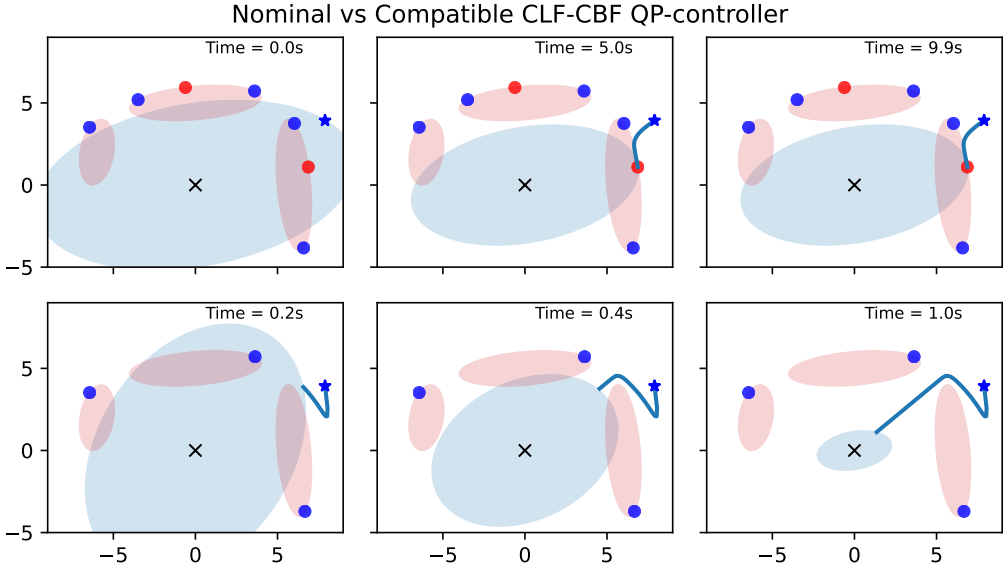



Figure 5.11: Compatible QP controller: fixed CLF vs adaptive strategy.

Now, consider another reference CLF $\overline{V}_r$, whose elliptical level set is shown in the first row of Fig. 5.11, with major-axis in the $y$-direction, and three quadratic barriers $h_1$, $h_2$ and $h_3$, with

unsafe sets shown in Fig. 5.11 as red ellipses (top, left and right, respectively). In this case, the reference CLF $\overline{V}_r$ is not compatible with $h_1$ (top obstacle) nor with $h_2$ (right obstacle), but it is compatible with $h_3$ (left obstacle). Notice the stable boundary equilibrium points on $\partial\mathscr{C}_1$ and $\partial\mathscr{C}_2$ (red points). Therefore, three Q-functions exist, one per CBF.

The first row of Fig. 5.11 shows the results obtained using the nominal QP controller (2.41) with a fixed reference CLF $\overline{V}_r$ (level set is the blue ellipse) and all three CBF constraints. For the initial condition shown at Fig. 5.11, with stabilizing CLF given by the reference $\overline{V}_r$, the state converges towards the red stable point at the boundary $\partial\mathscr{C}_2$, as expected.

Once again using our proposed strategy, three compatible Hessians are computed using the optimization (5.130): $H_{\overline{V}_1}$, $H_{\overline{V}_2}$ and $H_{\overline{V}_3}$, each being the $i$-th compatible Hessian closest to the reference CLF Hessian $H_{\overline{V}_r}$. Here, since $\overline{V}_r$ is already compatible with $h_3$, (only unstable equilibrium points exist in $\partial\mathscr{C}_3$), $H_{\overline{V}_3} = H_{\overline{V}_r}$. However, $H_{\overline{V}_1}$ and $H_{\overline{V}_2}$ are the Hessians of CLFs $\overline{V}_1$ (compatible with $h_1$) and $\overline{V}_2$ (compatible with $h_2$), respectively. Their level sets are also rotated and scaled versions of the level sets of $\overline{V}_r$. These computed Hessians are used in controller (5.133) to adapt the level sets of the stabilizing CLF $\overline{V}(x,\pi)$ while the state is inside $\Omega_1^{clf}$ or $\Omega_2^{clf}$, effectively removing the stable equilibrium points from the boundaries of $h_1$ and $h_2$ when the state approaches the regions $\Omega_1^{clf}$ or $\Omega_2^{clf}$ (where only the CLF, and the $h_1$ or $h_2$ constraints are active, respectively).

The second row of Fig. 5.11 shows the results obtained using the QP controller (2.41) with a CLF $V$ obtained from the inverse transformation of 5.99 of the quadratic CLF $\overline{V}(x,\pi)$. Once again, the system trajectory converge towards the origin and the CLF $\overline{V}$ converges towards $\overline{V}_r$ after the state leaves $\Omega_2^{clf}$.

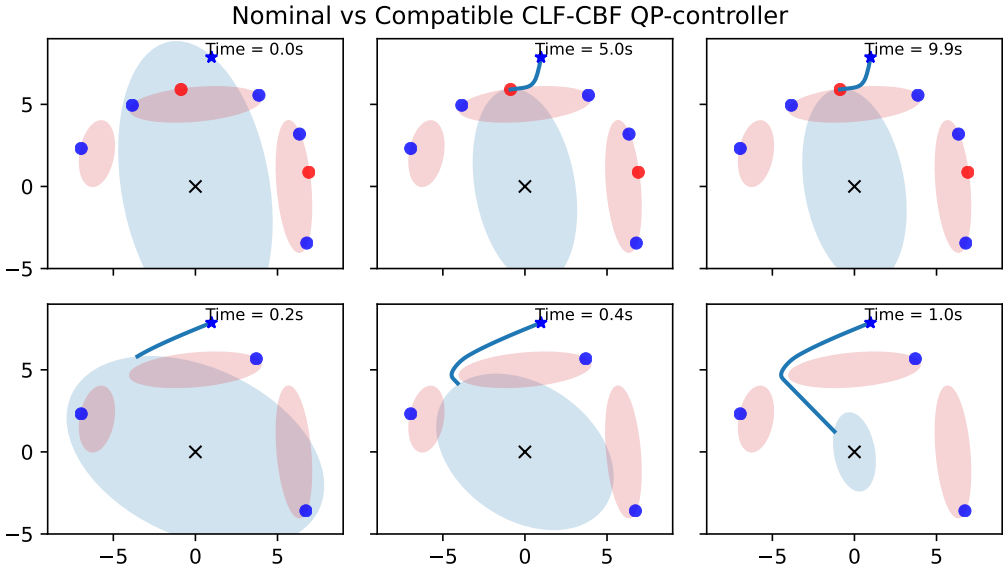



Figure 5.12: Compatible QP controller: fixed CLF vs adaptive strategy with a new initial condition.

Fig. 5.12 shows similar results to the ones obtained with the reference CLF $\overline{V}_r$ and another initial condition of Fig. 5.12, but with all three CBFs instead. One again, the first row shows the

results without the adaptive strategy for the Hessian of $\overline{V}$: the trajectory is attracted towards the stable boundary equilibrium point at the top obstacle. The second row shows the effect of dynamically changing the CLF Hessian with controller (5.133): the origin is globally asymptotically stable in this example. Notice that the presence of multiple barriers did not change the obtained results.

**Example 2.** Now, consider the *underactuated* two-dimensional LTI system $\dot{x} = Ax + Bu$ with

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Matrix $A$ is Hurwitz stable, and the pair $(A, B)$ is controllable. This example is of particular importance because matrix $B$ is not full-rank.

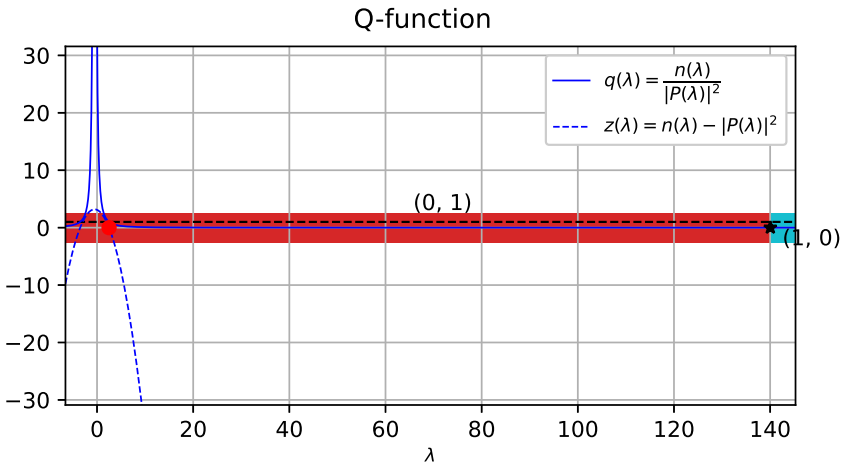



Figure 5.13: Q-function for underactuated system.

An example of a Q-function for the given system with a quadratic CLF that is not compatible with the corresponding CBF is shown in Fig. 5.13. As shown, the negative definite interval of $S(\lambda)$ (red strip) is $\approx (-\infty, 140)$, and a single *stable* boundary equilibrium solution exists at $\lambda_e \approx 2.5$. This is the boundary equilibrium point shown at the first row of Fig. 5.14, which shows the results for the QP controller (2.41) with the three CBFs from Example 1. As expected, the trajectory converges to this stable boundary equilibrium point. In the second row of Fig. 5.14, the results for the QP controller with the adaptive strategy for the CLF level sets are shown. Once again, three compatible Hessians were computed using the optimization (5.130): $H_{\overline{V}_1}$, $H_{\overline{V}_2}$ and $H_{\overline{V}_3}$, each being the $i$-th compatible Hessian closest to the reference CLF Hessian $H_{\overline{V}_r}$ (shown as the CLF with the level sets shown in the first row of Fig. 5.14). Here, $H_{\overline{V}_1} = H_{\overline{V}_3} = H_{\overline{V}_r}$ since the reference CLF $\overline{V}_r$ is already compatible with barriers $h_1$ and $h_3$ (no boundary equilibrium points exist at $\partial\mathscr{C}_1$ or $\partial\mathscr{C}_3$). The second computed compatible CLF Hessian $H_{\overline{V}_2}$ is such that the previously stable equilibrium point at $\partial\mathscr{C}_2$ was converted to an unstable one.

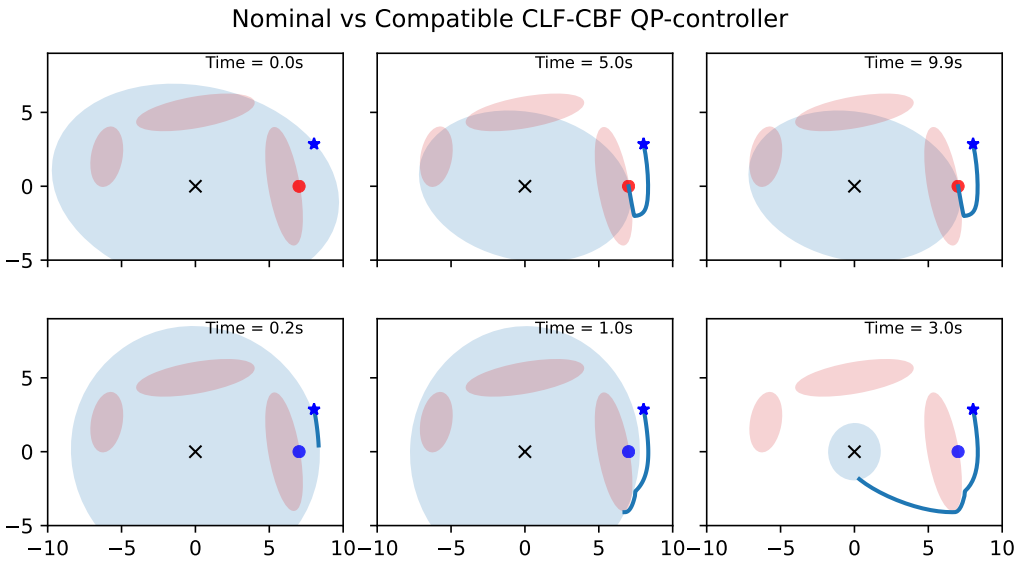



Figure 5.14: Compatible QP controller: fixed CLF vs adaptive strategy with underactuated system.

Instead of converging towards the boundary $\partial\mathscr{C}_2$, the trajectory circulates the second obstacle and converges to the origin, as expected. The stabilizing CLF $\overline{V}$ converges to the compatible $\overline{V}_2$ when close to barrier $h_2$, and after circulating the obstacles, it converges back to the reference $\overline{V}_r$, as expected.

## 5.4 Concluding Remarks and Discussions

In Section 5.1, we have presented a complete analysis for the existence and stability properties for all equilibrium points arising in the closed-loop system (2.14) with two CBF-based safety-critical controllers: (i) the safety filter QP and the (ii) the CLF-CBF QP. No particular assumptions were made for the nonlinear affine system model, such as the need for $g(x)$ to be full-rank.

The closed-loop Jacobian matrices for the boundary and interior equilibrium points, as well as theorems providing sufficient conditions for instability and stability of the equilibrium points arising in the QP-based formulation (Theorems 5.1.4, 5.1.5 and 5.1.6) were presented. In particular, the sufficient condition for instability of boundary equilibrium points on Theorem 5.1.6 was derived by using the QR decomposition on Definition 5.1.5, which has significantly reduced the complexity of the expression for the closed-loop Jacobian, clearing the way for deriving its main properties. The main conclusion of Section 5.1 is that the existence and stability of the closed-loop equilibrium points can be fully characterized by means of the vector-valued function $f_a(x,\lambda_a)$ and its Jacobian $J_{f_a}(x,\lambda_a)$, defined at (5.3) and (5.4), respectively, which was a surprising result. Another interesting theoretical development was the possibility of characterizing $f_a(x,\lambda_a)$ and its Jacobian $J_{f_a}(x,\lambda)$ in terms of an invertible transformation of the original CLF, defined at Definition 5.1.6, for the CLF-CBF QP controller. This fact was latter useful for the developments of Section 5.3.

Corollary 5.1.6.1 (considering the integrator system $\dot{x} = u$) and the technique for CLF compatibilization presented at Section 5.2 for systems with full rank $g(x)$ and a non-radial CLF were published at [64]. This work actually showed that undesirable equilibria exist in the CLF-CBF framework and can be asymptotically stable even for very simple systems. However, an important limitation of the results presented in [64] is that $g(x)$ is assumed to be full rank.

In Section 5.3, we introduce an algebraic theory for CLF compatibility using Linear Matrix Polynomials [33] for the classes of driftless full-rank systems and linear, time-invariant systems, and quadratic CLFs and CBFs under Assumption 9. Using the developed theory, we were able to demonstrate that certain boundary equilibrium points cannot be removed for certain types of systems, but their stability properties can be changed. We propose an algorithm for computing a compatible quadratic CLF with respect to a quadratic CBF, and a strategy to dynamically modify the CLF geometry, aiming to remove all stable equilibrium points from the closed-loop system. In particular, for linear systems, the developed theory does not require the input matrix $B$ to be full-rank.

# Chapter 6

# Conclusions and Future Works

## 6.1 Conclusions

In Chapter 3, we have presented novel results in robust control theory applied to the robotic navigation problems of Moving Path Following and Cooperative Path Following. The proposed strategies were able to provide superior control performance in experiments with marine robots in a real-world scenario, with many active disturbances such as unmodeled robot dynamics, wind, ocean currents and waves. As expected from the theory, both proposed sliding mode control strategies presented similar performance in terms of path errors, but the strategy with the disturbance observer has actually improved the levels of control chattering, which is a common issue of sliding mode controllers. As a limitation of the proposed schemes, we highlight that they rely on a fast response for the actuators and on large linear regions for the controller velocity outputs. Despite that, the proposed controllers were able to achieve satisfactory performance in scenarios where mild disturbances were present by only using the position of the target and reasonably accurate information about the bounds of the disturbances and the target velocity. We have also concluded that the robust MPF controller achieves acceptable performance even when the target velocities are not explicitly compensated. However, the robust gain $\rho$ must be sufficiently large to overcome this additional disturbance.

In Chapter 4, we have presented some safety-critical applications in robotics, ranging from how to achieve safe motion with vessels in marine environments to cooperative control with collision avoidance guarantees for autonomous cars operating on a traffic lane. We have focused particularly on the path following control strategy for navigation, moreover, using the CLF-CBF framework for safety-critical control. This research has concluded that both the safety-filter (using only a nominal controller for stabilization and CBFs for achieving safety) and the CLF-CBF QP-based controller (using CLFs for stabilization and CBFs for achieving safety) are efficient strategies for achieving safe behaviour but also suffer from similar limitations, namely the presence of undesirable equilibrium points (which is the topic of Chapter 5). However, they present some key important differences that need further clarification. The safety-filter from (2.36) relies on an external nominal controller for stabilization, which is minimally modified by the CBF

constraint when necessary, effectively achieving safe behaviour. However, it is not clear how this modification affects the stabilization of trajectories. A question that remains is if the stability of trajectories when the CBF constraint is active could be hampered, perhaps producing even unstable behaviour. Further research is needed on this topic. However, with the CLF-CBF QP-based controller, the CLF constraint *guarantees* stability, even if only local. It is also a more attractive strategy from the theoretical point of view, since desired properties for the stabilization process (such as convergence rate) and avoidance of unsafe states (such as repulsion from the unsafe sets) can be directly controlled by the CLF and CBFs, and the class $\kappa$ functions $\alpha$ and $\beta$ used in the CLF and CBF constraints, respectively.

We have also concluded that the problem of *trajectory deadlock* can easily arise in path following applications with multiple robots and intersecting paths, as shown in Section 4.3. In that section, we have solved this problem specifically for cooperative path following tasks by proposing a prioritization strategy for the vehicles. However, further investigation was needed, which led to the results presented in Chapter 5.

In Chapter 5, we have formally addressed the problem of undesirable equilibrium points in the CBF-based QP framework. The theory presented at Section 5.1 provides a formal and complete characterization of the problem which was directly used to propose possible solutions, by means of the definition of the concept of CLF compatibility. While Section 5.2 focused on avoiding the *existence* conditions of undesirable equilibria from occurring, thus providing a limited (in terms of its required assumptions), conservative and rather complex solution, Section 5.3 provides a simpler, direct and less conservative treatment to the problem (by allowing only *unstable points* to occur), but limited to a particular class of systems and CLF-CBF parametrizations. The fact that the developed theory for quadratic CLF compatibility using the Q-function defined in Section 5.3.3 is directly linked to Matrix Polynomial Theory [33] was also a surprising fact.

## 6.2 Future Works

As some future directions for the research on robust path following control, we highlight the need for generalizing the proposed controllers to handle the case of unknown bounds for the disturbances (perhaps by employing adaptive strategies on the control gains) and to take into account saturation limits for the robot actuators. Further investigation on the topic of robust MPF control using dynamic models for the vehicles (thus taking their full physical properties into account) would also help to advance research in the topic.

While in Section 5.3 we derive the CLF compatibility theory for quadratic CLF and CBFs only (by means of the Q-function), the definition of CLF compatibility introduced at Section 5.1.3 is general and does not require any assumption on its parametrization. It could also be extended to consider the safety-filter QP. This fact opens future directions of research where the nonlinear system model, CLFs and CBFs could potentially be given by more involved parametrizations, such as polynomial functions using techniques from Sums-of-Squares (SoS) optimization [27]. In that case, the expression for the stability manifold at (5.119) could still be written in terms of a singular

Matrix Pencil, instead of a regular one, which significantly elevates the complexity of the required theory for deriving explicit conditions for CLF compatibility. This research direction would also allow for the CLF-CBF framework to guarantee safety and *global* asymptotic stabilization in the case of highly nonlinear systems and non-convex CBFs (which could be used to model non-convex obstacles in collision avoidance) and even potentially to guide a robot inside of a maze without ever reaching deadlock configurations. Therefore, investigation on the topic of generalizing the Q-function theory developed in Section 5.3 for more general types of systems and CLF-CBF pairs is a crucial requirement for advancing the theory of CLF compatibility, with the final goal of achieving general conditions for quasi-global asymptotic stability of the origin in safety-critical control for general types of systems, where the objectives of stabilization and safety are fully compatibilized.

# References


[1] A. P. Aguiar. Single and multiple motion control of autonomous robotic vehicles. In *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, pages 172–184, July 2017.

[2] A. P. Aguiar and J. P. Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, 52(8):1362–1379, Aug 2007.

[3] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotovic. Path-following for nonminimum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, 50(2):234–239, Feb 2005.

[4] A. P. Aguiar and A. M. Pascoal. Dynamic positioning and way-point tracking of underactuated auvs in the presence of ocean currents. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 2, pages 2105–2110 vol.2, Dec 2002.

[5] A Pedro Aguiar and António M Pascoal. Coordinated path-following control for nonlinear systems with logic-based communication. In *2007 46th IEEE Conference on Decision and Control*, pages 1473–1479. IEEE, 2007.

[6] Bowen Alpern and Fred B Schneider. Defining liveness. *Information processing letters*, 21(4):181–185, 1985.

[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, Aug 2017.

[8] 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.

[9] 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.

[10] Zvi Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods & Applications*, 7(11):1163–1173, 1983.

[11] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.

[12] Joseph-Frédéric Bonnans, Jean Charles Gilbert, Claude Lemaréchal, and Claudia A Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

[13] José Braga, Anthony J Healey, and João Sousa. Navigation scheme for the LSTS SEACON vehicles: Theory and application. *IFAC Proceedings Volumes*, 45(5):69 – 75, 2012. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles.

[14] Sergio Carvalhosa, A. Pedro Aguiar, and A. Pascoal. Cooperative motion control of multiple autonomous marine vehicles: Collision avoidance in dynamic environments. *IFAC Proceedings Volumes*, 43(16):395 – 400, 2010. 7th IFAC Symposium on Intelligent Autonomous Vehicles.

[15] O. H. Dagci, U. Y. Ogras, and U. Ozguner. Path following controller design using sliding mode control theory. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 1, pages 903–908 vol.1, June 2003.

[16] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen. *Nonlinear Control Design for Mobile Robots*, pages 121–156. World Scientific Series in Robotics and Intelligent Systems, 1994.

[17] Giorgos A. Demetriou, Andreas Hadjipieri, Irene Erica Panayidou, Antonis Papasavva, and Stelios Ioannou. ERON: A PID controlled autonomous surface vessel. In *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, pages 1–5, 2016.

[18] H. Dias, P. Calado, R. Bencatel, R. Gomes, S. Ferreira, and J. Sousa. Operations with multiple unmanned systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3043–3044, Oct 2012.

[19] Zhe Du, Rudy R. Negenborn, and Vasso Reppa. Cooperative multi-agent control for autonomous ship towing under environmental disturbances. *IEEE/CAA Journal of Automatica Sinica*, 8(8):1365–1379, 2021.

[20] Denis Efimov and Wilfrid Perruquetti. Oscillating system design applying universal formula for control. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 1747–1752. IEEE, 2011.

[21] Denis Efimov, Wilfrid Perruquetti, and Mihály Petreczky. On necessary conditions of instability and design of destabilizing controls. In *53rd IEEE Conference on Decision and Control*, pages 3915–3917. IEEE, 2014.

[22] P. Encarnacao and A. Pascoal. 3D path following for autonomous underwater vehicle. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, volume 3, pages 2977–2982 vol.3, Dec 2000.

[23] Anders Forsgren, Philip E Gill, and Margaret H Wright. Interior methods for nonlinear optimization. *SIAM review*, 44(4):525–597, 2002.

[24] Randy A Freeman and Petar V Kokotovic. Inverse optimality in robust stabilization. *SIAM journal on control and optimization*, 34(4):1365–1391, 1996.

[25] E. W. Frew, D. A. Lawrence, C. Dixon, J. Elston, and W. J. Pisano. Lyapunov guidance vector fields for unmanned aircraft applications. In *2007 American Control Conference*, pages 371–376, July 2007.

[26] F.R. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, 1980.

[27] Karin Gatermann and Pablo A Parrilo. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192(1-3):95–128, 2004.

[28] Reza Ghabcheloo, A Pedro Aguiar, António Pascoal, Carlos Silvestre, Isaac Kaminer, and J Hespanha. Coordinated path-following in the presence of communication losses and time delays. *SIAM journal on control and optimization*, 48(1):234–265, 2009.

[29] N. Gu, Z. Peng, D. Wang, Y. Shi, and T. Wang. Anti-disturbance coordinated path-following control of robotic autonomous surface vehicles: Theory and experiment. *IEEE/ASME Transactions on Mechatronics*, pages 1–1, 2019.

[30] Wei Guan, Weinan Cao, Jinghai Sun, and Zuojing Su. Steering controller design for smart autonomous surface vessel based on CSF L2 gain robust strategy. *IEEE Access*, 7:109982–109989, 2019.

[31] Thomas Gurriet, Andrew Singletary, Jacob Reher, Laurent Ciarletta, Eric Feron, and Aaron Ames. Towards a framework for realizable safety critical control through active set invariance. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 98–106. IEEE, 2018.

[32] Joo P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

[33] Nicholas J Higham, D Steven Mackey, and Françoise Tisseur. Definite matrix polynomials and their linearization by definite pencils. *SIAM journal on matrix analysis and applications*, 31(2):478–502, 2009.

[34] R. P. Jain, A. P. Aguiar, and J. B. de Sousa. Cooperative path following of robotic vehicles using an event-based control and communication strategy. *IEEE Robotics and Automation Letters*, 3(3):1941–1948, July 2018.

[35] R. P. Jain, A. P. Aguiar, and J. Sousa. Self-triggered cooperative path following control of fixed wing unmanned aerial vehicles. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1231–1240, June 2017.

[36] R. P. Jain, A. Alessandretti, A. P. Aguiar, and J. B. De Sousa. Cooperative moving path following using event based control and communication. In *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*, pages 189–194, June 2018.

[37] R. Praveen Kumar Jain, Antonio Pedro Aguiar, Andrea Alessandretti, and Joao Borges de Sousa. Moving path following control of constrained underactuated vehicles: A nonlinear model predictive control approach. In *AIAA SciTech Forum*. American Institute of Aeronautics and Astronautics, Jan 2018.

[38] Mrdjan Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, 96:359–367, 2018.

[39] Isaac Kaminer, António Pascoal, Enric Xargay, Naira Hovakimyan, Chengyu Cao, and Vladimir Dobrokhodov. Path following for small unmanned aerial vehicles using L1 adaptive augmentation of commercial autopilots. *Journal of guidance, control, and dynamics*, 33(2):550–564, 2010.

[40] Yuri A Kapitanyuk, Hector Garcia de Marina, Anton V Proskurnikov, and Ming Cao. Guiding vector field algorithm for a moving path following problem. *IFAC-PapersOnLine*, 50(1):6983–6988, 2017.

[41] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002.

[42] John C Knight. Safety critical systems: challenges and directions. In *Proceedings of the 24th international conference on software engineering*, pages 547–550, 2002.

[43] Petar Kokotović and Murat Arcak. Constructive nonlinear control: a historical perspective. *Automatica*, 37(5):637–662, 2001.

[44] D. K. M. Kufoalor, E. Wilthil, I. B. Hagen, E. F. Brekke, and T. A. Johansen. Autonomous COLREGs-compliant decision making using maritime radar tracking and model predictive control. In *2019 18th European Control Conference (ECC)*, pages 2536–2542, 2019.

[45] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2):125–143, 1977.

[46] Peter Lancaster and Qiang Ye. Variational properties and Rayleigh quotient algorithms for symmetric matrix pencils. In *The Gohberg Anniversary Collection*, pages 247–278. Springer, 1989.

[47] Boqian Li, Shiping Wen, Zheng Yan, Guanghui Wen, and Tingwen Huang. A survey on the control lyapunov function and control barrier function for nonlinear-affine control systems. *IEEE/CAA Journal of Automatica Sinica*, 10(3):584–602, 2023.

[48] Yuanchang Liu, Richard Bucknall, and Xinyu Zhang. The fast marching method based intelligent navigation of an unmanned surface vehicle. *Ocean Engineering*, 142:363–376, 2017.

[49] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

[50] L. Ma and N. Hovakimyan. Cooperative target tracking in balanced circular formation: Multiple UAVs tracking a ground vehicle. In *2013 American Control Conference*, pages 5386–5391, June 2013.

[51] L. Ma and N. Hovakimyan. Cooperative target tracking with time-varying formation radius. In *2015 European Control Conference (ECC)*, pages 1699–1704, July 2015.

[52] Tamas G Molnar and Aaron D Ames. Composing control barrier functions for complex safety specifications. *IEEE Control Systems Letters*, 2023.

[53] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard. Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529, June 2007.

[54] Gennaro Notomista and Matteo Saveriano. Safety of dynamical systems with multiple non-convex unsafe sets using control barrier functions. *IEEE Control Systems Letters*, 6:1136–1141, 2022.

[55] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan 2007.

[56] T. Oliveira, A. P. Aguiar, and P. Encarnação. Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems. *IEEE Transactions on Robotics*, 32(5):1062–1078, Oct 2016.

[57] T. Oliveira, A. P. Aguiar, and P. Encarnação. Three dimensional moving path following for fixed-wing unmanned aerial vehicles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2710–2716, May 2017.

[58] Tiago Oliveira and Pedro Encarnação. Ground target tracking control system for unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, 69(1-4):373–387, 2013.

[59] Lynne E Parker. Multiple mobile robot systems. *Springer Handbook of Robotics*, pages 921–941, 2008.

[60] José Pinto, Paulo S Dias, Ricardo Martins, João Fortuna, Eduardo Marques, and João Sousa. The LSTS toolchain for networked vehicle systems. In *2013 MTS/IEEE OCEANS-Bergen*, pages 1–9. IEEE, 2013.

[61] Stephen Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

[62] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *International Workshop on Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.

[63] Pradya Prempraneerach, Manusak Janthong, Tanchanok Klanthip, Sittichai Boonyarak, Channarong Choosui, Kraisak Phothongkum, Surasak Timpitak, and Pasan Kulvanit. Autonomous way-point tracking navigation of surveying surface vessel with real-time positioning system. In *2015 International Computer Science and Engineering Conference (ICSEC)*, pages 1–6, 2015.

[64] M. F. Reis, A. P. Aguiar, and P. Tabuada. Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria. *IEEE Control Systems Letters*, 5(2):731–736, 2021.

[65] Matheus F. Reis and A. Pedro Aguiar. On the stability of undesirable equilibria in the quadratic program framework for safety-critical control. https://arxiv.org/abs/2402.08027, 2024.

[66] Matheus F Reis, Pallov Anand, and A Pedro Aguiar. Cooperative path following with collision avoidance guarantees using control lyapunov and barrier functions. In *APCA International Conference on Automatic Control and Soft Computing*, pages 181–193. Springer, 2022.

[67] Matheus F Reis, Gustavo Andrade, Francisco Neves, Paulo Silva, Rômulo T Rodrigues, and A Pedro Aguiar. A ROS implementation of the situational awareness and maneuvering systems for an autonomous marine vessel. In *OCEANS 2021: San Diego–Porto*, pages 1–9. IEEE, 2021.

[68] Matheus F. Reis, Gustavo A. Andrade, and A. Pedro Aguiar. Safe autonomous multi-vehicle navigation using path following control and spline-based barrier functions. In Lino Marques, Cristina Santos, José Luís Lima, Danilo Tardioli, and Manuel Ferre, editors, *Robot 2023: Sixth Iberian Robotics Conference*, pages 297–309, Cham, 2024. Springer Nature Switzerland.

[69] Matheus F. Reis, José P. Carvalho, and A. Pedro Aguiar. A unified stability analysis of safety-critical control using multiple control barrier functions. https://arxiv.org/abs/2503.15823, 2025.

[70] Matheus F. Reis, R. Praveen Jain, A. Pedro Aguiar, and João Borges de Sousa. Robust cooperative moving path following control for marine robotic vehicles. *Frontiers in Robotics and AI*, 6, 2019.

[71] Matheus F Reis, R Praveen Jain, A Pedro Aguiar, and João Borges de Sousa. Robust moving path following control for robotic vehicles: Theory and experiments. *IEEE Robotics and Automation Letters*, 4(4):3192–3199, 2019.

[72] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding Mode Control and Observation*. Springer, 1st edition, 2014.

[73] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[74] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[75] D. Soetanto, L. Lapierre, and A. Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 1765–1770 Vol.2, Dec 2003.

[76] Eduardo D Sontag. A universal construction of Artstein's theorem on nonlinear stabilization. *Systems & control letters*, 13(2):117–123, 1989.

[77] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 3:54–70, 2023.

[78] Wei Chian Tan, Ching-Yen Weng, Yu Zhou, Kie Hian Chua, and I.-Ming Chen. Historical data is useful for navigation planning: Data driven route generation for autonomous ship. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7478–7483, 2018.

[79] Xiao Tan and Dimos V Dimarogonas. On the undesired equilibria induced by control barrier function based quadratic programs. *Automatica*, 159:111359, 2024.

[80] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. Barrier Lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, 2009.

[81] Ultralytics. Yolov5. https://ultralytics.com/yolov5.

[82] Dong Wang, Jie Zhang, Jiucai Jin, and Xingpeng Mao. Local collision avoidance algorithm for a unmanned surface vehicle based on steering maneuver considering COLREGs. *IEEE Access*, 9:49233–49248, 2021.

[83] Li Wang. *Multi-robot coordination and safe learning using barrier certificates*. PhD thesis, Ph. D. thesis, Georgia Institute of Technology, 2018.

[84] Li Wang, Aaron D Ames, and Magnus Egerstedt. Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 2659–2664. IEEE, 2016.

[85] Y. Wang and D. Wang. Non-singular moving path following control for an unmanned aerial vehicle under wind disturbances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 6442–6447, Dec 2017.

[86] Peter Wieland and Frank Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467, 2007. 7th IFAC Symposium on Nonlinear Control Systems.

[87] Richard Wise and Rolf Rysdyk. UAV coordination for autonomous target tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6453, 2006.

[88] Elias Xidias and Dimitris Zissis. Real time autonomous maritime navigation using dynamic visibility graphs. In *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pages 1–6, 2018.

[89] J. K. Yook, D. M. Tilbury, and N. R. Soparkar. Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. *IEEE Transactions on Control Systems Technology*, 10(4):503–518, July 2002.

[90] Li-Jun Zhang, He-Ming Jia, and Da-Peng Jiang. Sliding mode prediction control for 3D path following of an underactuated AUV. *IFAC Proceedings Volumes*, 47(3):8799 – 8804, 2014. 19th IFAC World Congress.

[91] M. Zhang and H. H. T. Liu. Game-theoretical persistent tracking of a moving target using a unicycle-type mobile vehicle. *IEEE Transactions on Industrial Electronics*, 61(11):6222–6233, Nov 2014.

[92] Rubo Zhang, Pingpeng Tang, Yumin Su, Xueyao Li, Ge Yang, and Changting Shi. An adaptive obstacle avoidance algorithm for unmanned surface vehicle in complicated marine environments. *IEEE/CAA Journal of Automatica Sinica*, 1(4):385–396, 2014.

[93] Henryk Żołądek. The topological proof of Abel-Ruffini theorem. *Topological Methods in Nonlinear Analysis*, 16(2):253 – 265, 2000.