COOPERATIVE AUTONOMOUS SYSTEMS: MOTION PLANNING AND COORDINATED TRACKING CONTROL FOR MULTI-VEHICLE MISSIONS

BY

VENANZIO CICHELLA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Naira Hovakimyan, Chair
Professor Daniel Liberzon
Professor Isaac Kaminer
Professor Dusan M. Stipanović
Professor Petros G. Voulgaris

Abstract

In this dissertation a framework for planning and control of cooperative autonomous systems is presented, which allows a group of Unmanned Vehicle Systems (UxSs) to generate and follow desired trajectories, while coordinating along them in order to satisfy relative temporal constraints. The described methodology is based on two key results. First, a centralized optimal motion planning algorithm produces a set of feasible and flyable trajectories, which guarantee inter-vehicle safety, while satisfying specific temporal mission requirements, as well as dynamic constraints of the vehicles. Then, a distributed coordinated tracking controller ensures that the vehicles follow the trajectories while coordinating along them in order to arrive at the final destination at the same time, or with a predefined temporal separation, according to the mission requirements.

The optimal motion planning problem is formulated as a continuous-time optimal control problem, which is then approximated by a discrete-time formulation using Bernstein polynomials. Using the convergence properties of Bernstein polynomial approximation, the thesis provides a rigorous analysis that shows that the solution to the discrete-time approximation converges to the solution to the continuous-time problem. The motivation behind this approach lies in the fact that Bernstein polynomials possess favorable geometric properties that allow for efficient computation of various constraints along the entire trajectory, and are particularly convenient for generating trajectories for safe operation of multiple vehicles in complex environments.

The coordinated tracking algorithm relies on the presence of a virtual target tracking controller which guarantees that the distance between each vehicle and its assigned virtual target running along the desired trajectory remains bounded throughout the mission. Then, the speed of the virtual target is adjusted in order to satisfy the temporal constraints and achieve coordination. The coordination problem is formulated as a consensus problem, with the objective of regulating a suitably defined set of coordination variables to zero. Conditions are derived under which the consensus algorithm proposed solves the coordination problem in the presence of faulty communications and switching topologies.

"In that book which is my memory,

On the first page of the chapter that is the day when I first met you,

Appear the words 'Here begins a new life' "

Dante Alighieri, Vita Nuova

Dedicated to my wife Caterina

Acknowledgments

Most of all, I would like to express profound gratitude to my adviser, Naira Hovakimyan. You made me feel at home in Urbana, and taught me more than I could ever give you credit for here. Thank you for always encouraging my research, providing tremendous support, and allowing me to grow as a researcher. Your guidance, at both professional as well as personal levels, has been invaluable for me and for my future career goals.

I would like to thank the members of my PhD committee, Daniel Liberzon, Isaac Kaminer, Dusan M. Stipanović, and Petros G. Voulgaris, who have provided helpful feedback and have been great teachers, preparing me to get to this place in my academic life. I owe a special word of gratitude to Isaac Kaminer, who started me in my research journey well before UIUC, and whose ideas, creativity, and knowledge were crucial to the development of my research work.

The process that led to the completion of my PhD was strongly influenced by the dedicated work of many other brilliant researchers, to whom I wish to acknowledge my appreciation. These include Vladimir Dobrokhodov, Antonio Pascoal, Pedro Aguiar, Claire Walton, Anna Trujillo, Roberto Naldi, Lorenzo Marconi, Alex Kirlik, and Frances Wang.

I would also like to thank my past and present labmates and friends, Bilal, Hanmin, Zhiyuan, Hui, Evgeny, Ronald, Enric, Jan, Steve, Donglei, Thiago, Arun, Hyung-Jin, Hamid, Kasey, Javier, Gabriel, Alex, Arman, and Andrew, who made my PhD an enjoyable journey.

Finally, but by no means least, I would like to thank my parents, Patrizia and Paolo, whose unconditional love and guidance are with me in whatever I pursue; my brother Massimo, for being everything a little brother should be, and so much more; and my love and best friend Caterina, to whom I would like to dedicate this thesis: *to the chapters of the book that is our life together.*

* This work was supported in part by NASA, AFOSR and NSF.

Contents

Notation, symbols, and acronyms

$\mathbb{R}$	Field of real numbers.
$\mathbb{Z}$	Set of all integers.
$\mathcal{C}^r$	Space of functions with r continuous derivatives.
$\mathcal{C}^r_d$	Space of d-vector valued functions with r continuous derivatives.
$\mathsf{SO}(3)$	Special orthogonal group of all rotations about the origin of three-dimensional Euclidean space $\mathbb{R}^3$.
$\mathfrak{so}(3)$	Set of 3×3 skew-symmetric matrices over $\mathbb{R}$.
$\boldsymbol{I}_n$	Identity matrix of size n.
$\mathbf{0}$	Vector of appropriate dimension whose components are all 0.
$\mathbf{1}$	Vector of appropriate dimension whose components are all 1.
$\{\mathcal{F}\}$	Reference frame.
$\boldsymbol{\omega_{F1/F2}}$	Angular velocity of frame $\{\mathcal{F}1\}$ with respect to frame $\{\mathcal{F}2\}$.
$\boldsymbol{R^{F2}_{F1}}$	Rotation matrix from frame $\{\mathcal{F}1\}$ to frame $\{\mathcal{F}2\}$.
$\boldsymbol{v}^\top$	Transpose of vector $\boldsymbol{v}$.
$\|\boldsymbol{v}\|$	2-norm of vector $\boldsymbol{v}$.
$\|\boldsymbol{v}\|_\infty$	∞-norm of vector $\boldsymbol{v}$.
$\boldsymbol{M}^\top$	Transpose of matrix $\boldsymbol{M}$.
$\det(\boldsymbol{M})$	Determinant of matrix $\boldsymbol{M}$.
$\mathrm{tr}\,[\boldsymbol{M}]$	Trace of matrix $\boldsymbol{M}$.
$\lambda_{\max}(\boldsymbol{M})$	Maximum eigenvalue of matrix $\boldsymbol{M}$.
$\lambda_{\min}(\boldsymbol{M})$	Minimum eigenvalue of matrix $\boldsymbol{M}$.
$\|\boldsymbol{M}\|$	Induced 2-norm of matrix $\boldsymbol{M}$.
$(\cdot)^\wedge$	The *hat map*. (See Appendix A.1.)
$(\cdot)^\vee$	The *vee map*. (See Appendix A.1.)
■	End of Lemma, Corollary, and Theorem.

$\triangledown$	End of Problem.
$\square$	End of Property.
$\spadesuit$	End of Proof.
$\blacklozenge$	End of Remark.
$\triangle$	End of Assumption.
$\diamondsuit$	End of Example.

DoF	Degree of Freedom.
ISS	Input-to-State Stable.
PE	Persistency of Excitation.
QoS	Quality of Service.
UxS	Unmanned Vehicle System
UGS	Unmanned Ground System
UAS	Unmanned Aerial System
USS	Unmanned Space System
UMS	Unmanned Marine System
NPS	Naval Postgraduate School.
UIUC	University of Illinois at Urbana-Champaign.

Bold-face, lower-case letters refer to column vectors (e.g. $\boldsymbol{v}$), while bold-face, capital letters refer to matrices (e.g. $\boldsymbol{M}$). In general, the ith component of vector $\boldsymbol{v}$ is denoted by v_i, and the (i, j) entry of matrix $\boldsymbol{M}$ is represented by M_{ij}.

Chapter 1

Introduction

1.1 General description

The field of Unmanned Vehicle Systems (UxSs), including Unmanned Ground Systems (UGSs), Unmanned Aerial Systems (UASs), Unmanned Space Systems (USS), and Unmanned Marine Systems (UMSs), has gone through a major transformation in the past two decades. While in the nineties the focus was on developing large UxSs capable of carrying significant payloads at great distances, significant technological improvements have shifted academic, industrial, and governmental interest to operations that require multiple small UxSs functioning in cooperative ways. [1–19]. This stems from the fact that it is far more reliable and cost effective to deploy groups of heterogeneous UxSs with diverse capabilities and carrying different but complementary mission-dependent payloads. This yields considerable flexibility in the reconfiguration capabilities as well as graceful degradation of performance in case of failure of isolated UxS. To enable safe deployment of groups of UxSs, autonomous vehicles must be capable of performing missions in a cooperative fashion to achieve common objectives that may be dynamically assigned as the mission unfolds. During these missions, the vehicles must be able to operate safely and execute coordinated tasks in complex, highly uncertain environments while maneuvering in close proximity to each other and to obstacles. This poses multiple challenges inherent to the design, development, and operation of multiple UxSs. Central among them is the design of *motion planning and control* strategies for multi-agent systems. Addressing this challenge requires considerable effort from systems designers and poses a number of extremely interesting theoretical problems.

Over the past few years, there has been a wide range of topics related to motion planning and control of multiple UxSs that have been addressed in the literature. These topics include: formation control [15, 20–24], collective behaviors and flocking [9–12, 25], synchronization [6–8], multi-agent differential games [26–28], multi-agent adaptive dynamic programming and reinforcement learning [29–34], cooperative path and trajectory planning [3, 18, 35–41], coordinated motion control [16, 42–44], and graph theoretic methods for multi-agent systems [4, 5, 45–49]. Particularly relevant are the applications of the theory to enable multi-vehicle missions involving UASs [50–58], UGSs [59–63], USSs [64–66], UMSs [67–74], and heterogeneous

UxSs [75–77]. Nevertheless, in spite of the significant body of literature in the field, much work remains to be done to develop strategies capable of providing the levels of flexibility, performance, and safety required for the multiple UxSs cooperative missions envisioned in this thesis.

Motivated by these ideas, this thesis addresses the problem of steering a group of UxSs along desired trajectories while meeting relative temporal constraints. In particular, the cooperative missions considered require that a motion planning algorithm generates multiple trajectories that are feasible and collision free, that each vehicle tracks these trajectories, and that the group maintains a desired timing plan to ensure that all vehicles arrive at their respective final destinations at the same time, or at different times so as to meet a desired inter-vehicle schedule.

The framework developed in this thesis comprises of (i) a centralized optimal motion planning algorithm that generates trajectories for multiple UxSs, and (ii) coordination and tracking controllers that enable the UxSs to follow these trajectories while coordinating with each other, in the presence of partial vehicles failures and external disturbances. The advantages of this framework compared to solutions such as, for example, formation control [15, 20–24], collective behaviors and flocking [9–12, 25], and synchronization [6–8] are twofold:

- the strategy proposed is more general, and allows a large class of cooperative tasks to be executed. A compelling example of such tasks is a scenario where a number of UxS are required to maneuver from initial to desired target positions, with the constraint that they avoid collisions and arrive simultaneously at pre-assigned locations. Other illustrative examples fall in the scope of sequential auto-landing, cooperative ground target search, flocking, synchronization, and formation;

- seamless integration of tracking controllers and coordination algorithms allows to decouple the coordinated tracking control problem into separate ones. This decoupling, in turn, reduces the coordination problem into a simpler consensus problem on suitably defined variables with integrator dynamics [78–81]. Differently from works on consensus for multi-agent systems [82–87], where the vehicles dynamics are injected into the consensus problem, this simplification allows us to consider more compelling scenarios such as communication limited environments and rapidly changing topologies, while retaining rate of convergence guarantees.

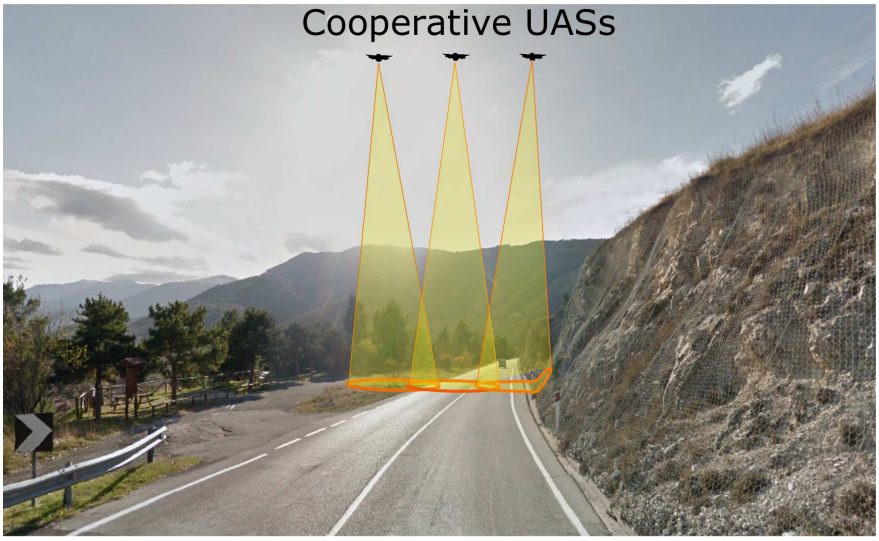
Nevertheless, with our approach much effort must be exert in the area of motion planning, in order to develop algorithms capable of generating trajectories for multiple vehicle missions (near) real-time.

This thesis presents a rigorous formulation of the problems of optimal motion planning and coordinated tracking control for multi-vehicle missions, and offers solutions to these problem for a general class of UxSs.

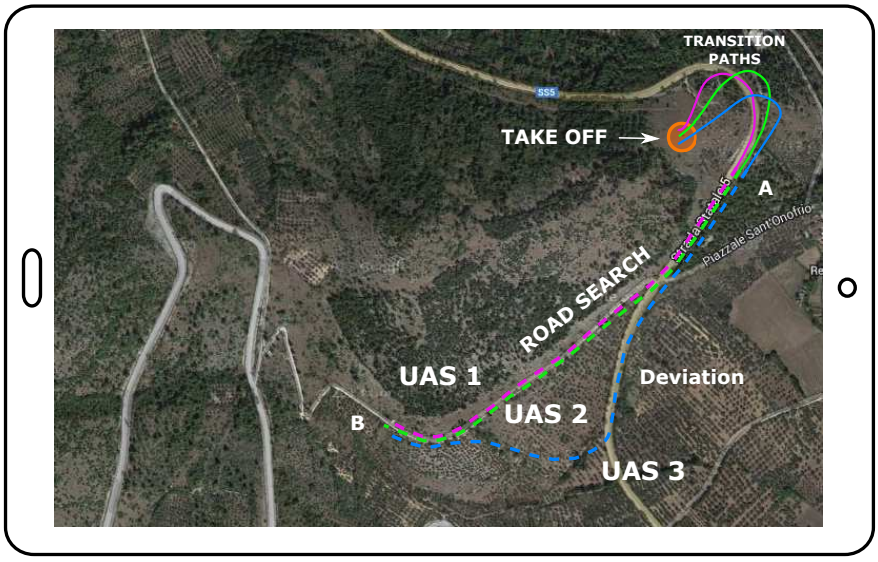
The thesis proposes a method for motion planning based on numerical approximation of optimal control problems using Bernstein polynomial approximation. The convergence properties of this method are analyzed, and the efficiency of the resulting motion planning algorithm is studied through several numerical examples. Furthermore, the thesis presents a coordination control algorithm, and studies its performance in terms of the Quality of Service (QoS) of the network over which the UxSs communicate, which is affected by temporary loss of communication links and switching communication topologies. To better motivate the theoretical developments presented in this dissertation, the next section describes a mission scenario that warrant the use of a groups of cooperating UxSs.

1.2 Motivational scenario: coordinated road search

One of the applications that motivates the use of multiple cooperative UxSs and poses several challenges to systems engineers, both from a theoretical and practical standpoint, is autonomous road search. In what follows we propose an example of road search mission scenario featuring three multirotor UASs. The example is depicted in Figure 1.1. The mission at hand is triggered by a user who selects an area on an electronic device displaying a digital map on a touch screen. The global coordinates of the selected region are sent to the multirotor UASs. An optimization algorithm computes *transition paths*, which start at the vehicles' initial positions, and end at the starting point of the road search mission. Additionally, the optimization algorithm generates *road search paths*, which follow the road to allow the vehicles to inspect the selected area. The transition and road search paths are deconflicted and satisfy the dynamic constraints of the vehicles. Further, the desired position and speed of each UAS at the end of the transition paths coincides with the position and speed at the beginning of the road search paths, respectively, to allow for a smooth progression of the mission. Then, the UASs can execute the cooperative road search mission by following the paths, and at the same time enforcing the mission's temporal constraints. Coordination along the transition paths ensures that the vehicles arrive at the start-points of the road search paths at the same time with desired speed profiles, and ensures inter-vehicle collision avoidance. Coordination along the road search paths guarantees overlapping of the fields of view of the three cameras, as emphasized by Figure 1.1a. Finally, it is possible that the fleet of multirotor UASs must address goals that were not initially planned and appear dynamically as the mission unfolds. This is the case for UAS 3 in Figure 1.1b, which is required to deviate and inspect a secondary road. After inspection, the vehicle re-converges to the original road search path synchronizing with the rest of the fleet. This last part brings to the reader's attention the benefits of employing cooperative control algorithms that —like the one presented in this dissertation— do not necessarily lead to swarming



(a) *Google Maps*. 3D view of the cooperative road search scenario.



(b) *Google Maps*. 2D projection of the cooperative road search scenario.

Figure 1.1: Cooperative road search using multiple multirotor UASs. The figures illustrate a scenario in which cooperation among the UASs is required to accomplish the task at hand. The UASs, starting from random initial positions, follow the transition paths, depicted as solid lines, and arrive at point A. Then, they proceed along the road search paths, represented by solid lines, while coordinating with each other to accomplish the cooperative road search mission. Cooperation along the road search paths guarantees non-zero intersection between the fields of view of the cameras.

behaviors.

In the mission scenario described above, the advantages of using a cooperative group of autonomous vehicles connected by means of a communication network —rather than a single, heavily equipped vehicle— can be immediately identified. In a cooperative scenario, the team can reconfigure the network in response to unplanned events as well as changing mission objectives, and optimize strategies for improved target detection and discrimination. Use of multiple vehicles also improves robustness of the mission execution against single-point system failures. Furthermore, in a multi-UAS approach, each vehicle of the team may be required to carry only a reduced number of sensors, making each of the vehicles in the fleet less complex, thus increasing overall system reliability. This cooperative approach requires, however, the implementation of robust cooperative control algorithms that will allow the fleet of UASs to maneuver in a coordinated manner and combine the complementary capabilities of the on-board sensors. In fact, flying in a coordinated fashion is critical to maximizing the overlap of the fields of view of multiple sensors while reliably maintaining a desired image resolution.

1.3 Literature review and statement of contributions

The framework adopted builds upon a number of important concepts and techniques that have been the subject of intensive research. In what follows we briefly review the literature on the most relevant topics exploited in this thesis, namely motion planning and coordinated tracking control, and outline the main contributions of this thesis. Additional bibliographic references are included throughout the thesis.

1.3.1 Optimal motion planning

Motion planning plays a key role in enabling autonomous systems accomplish tasks assigned to them safely and reliably. Over the past decades many approaches to generating trajectories have been proposed. Examples include bug algorithms, artificial potential functions, roadmap path planners, cell decomposition methods, and optimal control based trajectory generation. Discussions and details on these methods can be found in [88–95] and references therein. Each technique has different advantages and disadvantages, and is best-suited for certain types of problems. Motion planning based on optimal control –i.e. optimal motion planning– is particularly suitable for applications that require the trajectory to minimize (or maximize) some cost function while satisfying a complex set of vehicle and problem constraints.

In general, finding a closed-form solution to nonlinear constrained optimal control problems is difficult or even impossible. Direct methods can be used to approximate optimal control problems to simpler problems,

which are easier to solve [96–98]. Direct methods based on discretization, for example, approximate the states of the dynamic system, or its inputs or both, thus reducing the original problem into a nonlinear programming problem (NLP) [98], which can then be solved by nonlinear optimization solvers. An important role in the literature on direct methods based on discretization is played by the work of Polak on *consistency of approximation theory* (see [99, Section 3.3]). Borrowing tools from variational analysis, Polak provides a theoretical framework to assess the convergence properties of discretization schemes for optimal control problems. Motivated by the consistency of approximation theory, a wide range of methods that use different discretization schemes have been developed. Few examples include Euler [99], Runge-Kutta [100], pseudospectral (PS) [101] methods, as well as the method presented in this dissertation.

The pseudospectral optimal control method is one of the most popular direct methods nowadays, and it has been applied successfully for solving a wide range of optimization problems, e.g. [35,101–106]. Pseudospectral methods offer several advantages over many other discretization methods, mainly owing to their spectral (exponential) rate of convergence. For example, consider the Legendre PS optimal control method [101], one of the most widely used PS methods for motion planning. It is characterized by the following three features: (*i*) the continuous functions involved in the optimal control problem, i.e. the states and control inputs, are approximated at N quadrature nodes, which are determined by the Legendre polynomial; (*ii*) the integral in the cost function is approximated by Legendre-Gauss quadrature; (*iii*) orthogonal collocation (also deemed as PS method), such as Lagrange interpolation on the Legendre nodes, is used to approximate functions and their derivatives (dynamics constraints). It was proven that under some assumptions on the solution to the original optimal control problem, a solution to the discretized optimal control problem exists, and it converges to the solution of the original problem. One of the features that makes PS methods particularly attractive is the convergence rate of the polynomial interpolation at the quadrature nodes. In particular, letting $I_N f(t)$ be the polynomial interpolation of $f(t)$ at the Legendre nodes in the interval $[-1, 1]$, the following result holds:

$$||I_N f(t) - f(t)||_{L^2} \leq \frac{C}{N^m},$$

where C is a positive variable independent on N, and m is the smoothness of $f(t)$. However, as pointed out in [107, 108], there is one salient drawback associated with PS methods. When discretizing the trajectories, the constraints are enforced at the discretization nodes. Unfortunately, satisfaction of constraints cannot be guaranteed in between the nodes. To avoid violation of the constraints in between the nodes, the order of approximation (number of nodes) can be increased. However, this leads to larger nonlinear programming problems, which may become computationally expensive and too inefficient to solve. This problem does not limit itself to PS methods, but it is common to methods that are based on discretization.

This undesirable behaviour becomes obvious, for example, in the multi-vehicle missions considered in this thesis, where a large number of vehicles have to reach their final destinations by following trajectories that are spatially (rather than temporally) separated to guarantee inter-vehicle safety. Clearly, with a small order of approximation, spatial separation between the trajectories will be hardly satisfied. Increasing the number of nodes will eventually produce spatially separated trajectories, but will also drastically increase the number of collision avoidance constraints and thus the complexity of the problem (the problem has $\frac{n!}{(n-2)!2!}N^2$ deconfliction constraints, where n is the number of vehicles, and N is the number of nodes).

This thesis proposes a direct method based on Bernstein approximation of the trajectories. Bernstein approximants have several nice properties. First, Bernstein basis possesses optimal numerical stability properties [109, 110], and can handle large order of approximations without suffering from numerical instability issues. Second, the approximants converge uniformly to the functions that they approximate – and so do their derivatives [111, Chapter 3]. This, as we will discuss later, is useful to derive convergence properties of the proposed computational method. Third, due to their favorable geometric properties (see [111, Chapter 5]) Bernstein polynomials afford computationally efficient algorithms for the computation of constraints such as minimum and maximum velocity, acceleration, minimum distance between paths, etc., for the whole trajectory, and not only at discretization points (see [112, 113]). Hence, with the proposed approach the trajectories are guaranteed to be dynamically feasible and collision-free for all times, while retaining the computational efficiency of methods based on discretization.

Bernstein approximation converges slower than other interpolation or approximation techniques. This implies that the approach proposed in this thesis is outperformed by, for example, PS methods in terms of accuracy of approximation of the optimal solution. This is not surprising, since the choice of nodes and interpolating polynomial in PS methods is dictated by approximation accuracy and convergence speed, while sacrificing constraints satisfaction in between the nodes. On the other hand, our approach prioritizes safety and constraint satisfaction, at the expense of a slower convergence rate.

Bernstein polynomials are very useful tools to describe geometric paths, and a growing number of works in the literature exploit their properties for trajectory generation (see, for example, [114–117]). Using the notion of consistency of approximation introduced by Polak [99], the present thesis provides a theoretical foundation for the use of Bernstein polynomials in optimal motion planning. Similar consistency results have been demonstrated for various discretization schemes, including pseudospectral methods [118,119]. However, these results are limited to collocation methods [120]. The contribution of the present work is an extension of these results to a class of non-collocation methods.

1.3.2 Coordinated tracking control

The problem of coordinated tracking control, also referred to as coordinated path following control in the literature, can be briefly described as that of making a group of vehicles converge to and follow a set of desired trajectories, provided by a motion planning algorithm, while meeting pre-specified spatial and temporal constraints. Relevant work on this topic can be found in [22, 72, 121–129].

Coordinated tracking control was initially inspired by the work reported in [121], where the authors presented a solution for coordinated operation of a surface and underwater marine vehicles. One of the main drawbacks of this work lies in the fact that it requires the vehicles to exchange a large amount of information, and cannot be easily generalized to large scale multi-vehicle missions. This drawback was later overcome by the work in [123], where the authors built on the results on path following control presented in [130] to coordinate two underwater vehicles. The main idea in this approach is that the path following controller enables the vehicles to follow geometric paths, independently of the temporal assignments, and is thus in contrast to trajectory-tracking control, where the objective is to follow a predefined trajectory with a given timing law [131]. Therefore, one can exploit the progression of the desired references along the given paths to achieve coordination objectives. With this setup, the two vehicles only need to exchange a scalar value, which defines the along-path positions of their virtual targets, thus drastically reducing the amount of information to be exchanged among vehicles.

In [132], the authors extended the approach in [123] and addressed the problem of steering a group of vehicles along given spatial paths while holding a desired time-varying geometrical formation pattern. Conditions were derived under which the the proposed algorithm solves the coordinated control problem in the presence of communication losses, time delays and switching topologies. The approach in [123] was also extended in [51], where the authors addressed the problem of coordinated control of multiple fixed-wing UASs. To enforce the temporal constraints of the mission, the coordination algorithm relies on a distributed control law with a proportional-integral structure, which ensures that each vehicle travels along its path at the desired constant speed and also provides disturbance rejection capabilities against steady winds. The approach presented in [51] was later extended in [133] to the case of arbitrary feasible desired speed profiles.

The work presented in this thesis builds on the results presented in [133], and it departs from it in a fundamental way. In [133], a path following controller is designed so as to align the velocity vector of the UAS with the local tangent vector of the desired path, and it relies on the assumption that the speed of the vehicle is lower bounded by a positive constant. Then, coordination is achieved by varying the speed of the vehicles involved in the mission. One of the key steps in the approach proposed in [133] lies in the design of the path following solution, which significantly reduces the complexity of the problem at hand by reducing the

coordination dynamics to n simple integrators, where n is the number of UASs. However, while [133] offers an appealing solution for the cooperative control of fixed-wing UASs, it cannot be employed when dealing with UxSs that allow the existence of zero velocity vectors (e.g. UASs who can hover, such as multirotors, or UGSs). This limitation motivated us to reformulate the coordination problem in a different way. The goal of the work presented in this thesis is to provide a new solution to the coordination problem which is more general, and can be applied to a broader set of vehicles with different dynamics. In the approach proposed here, the *virtual target (VT) tracking* and the *coordination* control problems are decoupled. At the VT tracking level, we assume that a control law that enables a UxS to track a virtual target moving along its assigned path is given. At the coordination level, the synchronization problem is solved by adjusting a new set of suitably defined coordination variables, thus achieving vehicles' coordination. It is shown that the solution to the coordination problem exhibits guaranteed performance in the presence of time-varying communication networks, that arise due to temporary loss of communication links and switching communication topologies.

1.4 Thesis overview

In the remainder of this thesis, we present solutions to the optimal motion planning and coordinated tracking problems. The dissertation is organized as follows.

- Chapter 2 presents the general framework for cooperative vehicle missions proposed in this thesis, and provides a rigorous formulation of the problem at hand. The objective is to enable a group of UxSs to coordinate along a set of desired paths in order to meet strict spatial and temporal constraints. The chapter introduces the problems of optimal motion planning and coordinated tracking control, together with a set of assumptions and constraints on the tracking controllers implemented on-board the UxSs, as well as on the supporting communications network over which the vehicles exchange information.

- Chapter 3 addresses the problem of optimal motion planning for differentially flat systems [134]. This class of systems is particularly suited for motion planning, since the trajectory can be planned in (flat) output space, and the states and inputs can be computed through algebraic mappings. Thus, the optimal motion planning problem reduces to a simpler calculus of variations problem. Moreover, the majority of UxSs of our interest have been shown to be differentially flat [134, 135], making the approach presented in this chapter applicable to a wide range of applications. The chapter starts by formulating the optimal motion planning problem for differentially flat systems as a calculus of variations problem. Then, it approximates this problem into a nonlinear programming problem using Bernstein polynomials, and demonstrates consistency results for the proposed method. Numerical

examples are presented and discussed at the end of the chapter.

- Chapter 4 extends the results presented in Chapter 3 to a more general class of optimal control problems. Similarly to Chapter 3, this chapter provides an approximation of an optimal control problem into a nonlinear programming problem using Bernstein polynomials, and it demonstrates consistency results for this approximation method. Finally, it presents numerical results that demonstrate the advantages of the proposed approach.

- Chapter 5 presents the development of a control algorithm that solves the VT tracking problem for a multirotor UAS. An outer-loop VT tracking control law is presented that enables the vehicle, equipped with an autopilot tracking angular rates and thrust reference commands, to converge to and follow a desired virtual target. Limits in the performance of the autopilot are considered. The main advantage of considering angular rates and total thrust as control inputs is that such control strategy can be employed to a larger set of multirotor craft, independently of the number of propellers and geometric configurations.

- Chapter 6 addresses the problem of coordinating a group of UxSs. This problem is solved by regulating a set of suitably defined coordination variables to zero. The chapter defines a set of states that capture the coordination objective at hand, and proposes control laws that regulate these states. Then, it derives the performance of the proposed algorithms in the presence of time-varying communication networks, that arise due to temporary loss of communication links and switching communication topologies. Finally, simulation results that support the theoretical findings are presented.

- Chapter 7 presents flight test results for two quadrotor UAVs that verify the stability and convergence properties of the coordination control results presented in Chapter 6. In particular, two scenarios are proposed in which the quadrotors are required to accomplish simple cooperative tasks, namely *phase on orbit coordination* and *spatial coordination along one axis*. The chapter describes the system architecture and the indoor facility used to conduct the experiments, and discusses the flight test results in details.

- Concluding remarks are provided in Chapter 8.

Chapter 2

General framework and problem formulation

2.1 General framework

The integrated framework for cooperative vehicle missions proposed in this thesis uses a *hybrid* set-up, where a *central* unit is responsible for the mission planning and communicates with the vehicles before the beginning of the mission. Subsequently, *decentralized* controllers embedded on-board the vehicles ensure that the mission is accomplished in a safe manner by exchanging information with each other. The framework, which is depicted in Figure 2.1, can be summarized in three fundamental steps outlined below.

- **Motion planning**: given a multiple UxS mission, a set of desired trajectories is generated off-line by an optimal motion planning algorithm for all the vehicles involved in the mission. These trajectories optimize a cost function, and satisfy initial and final boundary conditions, flyability constraints (e.g. min/max speed, min/max acceleration, etc.), feasibility constraints (e.g. collision avoidance between the trajectories, collision avoidance with obstacles), and additional mission-specific constraints.

- **VT tracking**: the trajectories (which are *geometric paths* parameterized by *time*) are re-parameterized by independent variables (here referred to as *virtual times*). These re-parameterized trajectories, called *virtual targets* henceforth, provide the reference to be tracked by the UxSs. Then, the objective of the VT tracking controller is to make sure that each UxS follows its assigned virtual target with guaranteed performance. The VT tracking controller can be designed to handle external disturbance and nonlinear vehicle dynamics with uncertainties.

- **Coordination**: The progression of the virtual time of each vehicle is adjusted on-line in order to ensure that the group of UxSs meets the temporal requirements of the mission, i.e. coordination. This step relies on the underlying communications network as a means to exchange information among vehicles, and takes into account tracking errors that incur due to vehicles' partial failures, external disturbances, etc. This step allows each vehicle to directly react in a timely fashion to other vehicles failures and potentially hazardous maneuvers, without having to communicate with a central station.

The framework exhibits a multi-loop structure, in which the VT tracking controller stabilizes the position of the vehicle around a virtual target, while a coordination controller is designed to control the virtual target. To make these ideas more precise, we notice that the equation of motion of the ith UxS involved in the cooperative mission can be described by the following system of equations:

$$\mathcal{G}_{v,i}: \begin{cases} \dot{\boldsymbol{x}}_{v,i}(t) = \boldsymbol{f}_i(\boldsymbol{x}_{v,i}(t), \boldsymbol{u}_{v,i}(t)), \quad \boldsymbol{x}_{v,i}(0) = \boldsymbol{x}^i_{v,i} \\ \boldsymbol{p}_{v,i}(t) = \boldsymbol{g}_i(\boldsymbol{x}_{v,i}(t)), \end{cases} \tag{2.1}$$

where $\boldsymbol{p}_{v,i}(t)$ is the vehicle's position, $\boldsymbol{x}_{v,i}(t)$ is the state of the vehicle (which typically includes position, attitude, velocity), $\boldsymbol{u}_{v,i}(t)$ is the control input (e.g. angular rate, speed, thrust), and $\boldsymbol{f}_i$ and $\boldsymbol{g}_i$ are vectors of nonlinear functions describing the *nominal* behavior of the UxS. The model above is sufficiently general to capture six-degree-of-freedom (6DoF) dynamics of UxSs. On one hand, a VT tracking controller $\boldsymbol{u}_{v,i}(t)$ can be designed for system $\mathcal{G}_{v,i}$ in order to provide tracking capabilities, i.e. make the vehicle converge to and follow a virtual target moving along a desired path, while handling external disturbance and uncertainties that may affect the nominal behavior given by (2.1). On the other hand, a coordination controller is derived to control the motion of the virtual target (i.e. the reference to be followed by the VT tracking controller) in order to achieve coordination, while taking into account VT tracking errors. This multi-loop approach not only simplifies the design process, but also lends itself to a wide range of applications involving multiple UxSs with different dynamics.

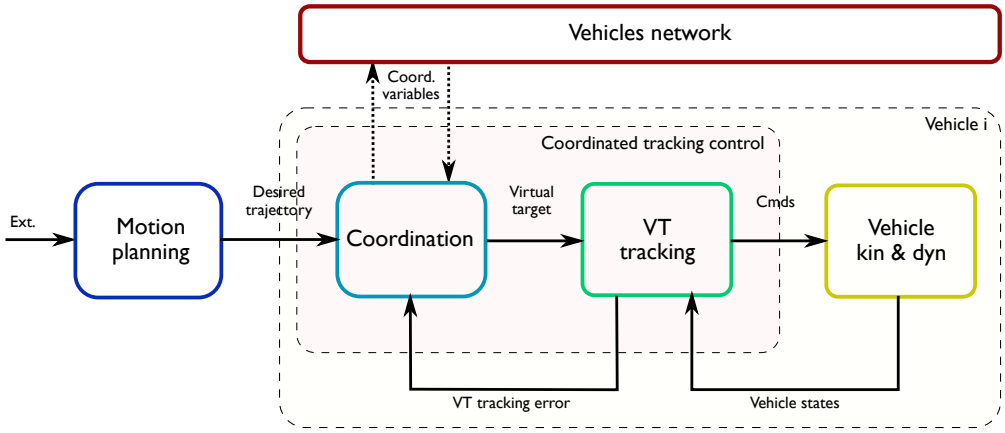


Figure 2.1: Architecture of the cooperative planning and execution framework adopted.

2.2 Problem formulation

2.2.1 Optimal motion planning

Given a set of n UxSs, the problem of optimal motion planning for multi-vehicle missions can be defined as follows:

Problem 1 *Compute a set of n desired trajectories $\boldsymbol{p}_{d,i} : [0, t_{f,i}] \to \mathbb{R}^d$, $i = 1, \ldots, n$, that minimize a given cost function, and satisfy boundary conditions, flyability constraints, feasibility constraints, and pre-defined mission-specific constraints.*

$\triangledown$

Boundary conditions

In the problem of motion planning for autonomous vehicles, typically the initial and final conditions of the trajectories, here referred to as boundary conditions, are pre-specified. For example, for the ith vehicle the following boundary conditions can be enforced:

$$\boldsymbol{p}_{d,i}(0) = \boldsymbol{p}_i^{\mathrm{i}}\,, \quad \|\dot{\boldsymbol{p}}_{d,i}(0)\| = v_i^{\mathrm{i}}\,, \quad \boldsymbol{p}_{d,i}(t_{f,i}) = \boldsymbol{p}_i^{\mathrm{f}}\,, \quad \|\dot{\boldsymbol{p}}_{d,i}(t_{f,i})\| = v_i^{\mathrm{f}}\,, \tag{2.2}$$

where $\boldsymbol{p}_i^{\mathrm{i}}$ and v_i^{i} are the initial position and speed, respectively, while $\boldsymbol{p}_i^{\mathrm{f}}$ and v_i^{f} are the specified quantities at the final endpoint of the trajectory.

Flyability and feasibility constraints

Flyable trajectories are trajectories that satisfy desired geometric constraints (such as curvature and flight path angle bounds), and can be tracked by a given vehicle without having it exceed pre-specified bounds on the vehicle dynamic state and control input (such as speed limits, angular rate bounds, or acceleration limits). These bounds depend on the UxSs considered, and on their physical limitations. For example, in [81, 136] it has been shown that a multirotor UAS, equipped with an autopilot in charge of tracking angular rate and total thrust commands, is capable of following trajectories subject to minimum and maximum speed limits, and maximum acceleration limits. Hence, the flyability constraints of multirotors can be specified as

$$v_{d,i}^{\min} \leq \|\dot{\boldsymbol{p}}_{d,i}(t)\| \leq v_{d,i}^{\max}\,, \qquad \|\ddot{\boldsymbol{p}}_{d,i}(t)\| \leq a_{d,i}^{\max}\,, \tag{2.3}$$

where $v_{d,i}^{\min} \geq v_i^{\min}$, $v_{d,i}^{\max} < v_i^{\max}$, and $a_{d,i}^{\max} < a_i^{\max}$, and $v_i^{\min} \geq 0$, $v_i^{\max} > 0$, and $0 < a_i^{\max} \leq g$ are the actual speed and acceleration limits of the multirotor, which can be determined from the maximum available thrust

of the rotors, and g is gravity. Note that during the motion planning phase, the more restrictive bounds $v_{d,i}^{\min}$, $v_{d,i}^{\max}$ and $a_{d,i}^{\max}$ are used so as to allow the vehicles to adjust on-line their dynamics, if necessary, in order to maintain coordination, or to react to unpredicted situations (e.g. pop-up obstacles, external disturbance).

Feasible trajectories are flyable trajectories that avoid collisions with obstacles, and that are spatially deconflicted in order to avoid inter-vehicle collision, thus ensuring safe simultaneous operation in a common airspace. Spatial deconfliction between trajectories can be guaranteed through *temporal separation* (the vehicles are separated in time) or *spatial separation* (the spatial paths are separated in space). Letting $\boldsymbol{p}_{d,i}(t)$, $i = 1, \ldots, n$, be the desired trajectories of the UxSs at time t, temporal separation can be enforced as follows:

$$\min_{\substack{i,j=1,\ldots,n \\ i \neq j}} \left\| \boldsymbol{p}_{d,i}(t) - \boldsymbol{p}_{d,j}(t) \right\|^2 \geq E_d^2 \,, \qquad \forall t \in [0, \max(t_{f,i}, t_{f,j})], \tag{2.4}$$

with $E_d \geq E$, where E is a minimum separation requirement, which depends, for example, on the dimension of the UxSs and other safety considerations. Similarly to the flyability constraints, we note that a more conservative bound E_d is imposed at the motion planning phase, in order to account for unpredicted situations (e.g. pop-up obstacles) or tracking errors of the UxSs (due to the presence of external disturbance or partial failures).

While temporal separation is more computationally efficient, it relies heavily on the performance of a time-coordination algorithm, which, in turn, depends on the quality and robustness of the communication network over which the vehicles exchange information with each other. On the other hand, spatial separation can be employed in situations where the communication network is faulty or jammed, and coordination cannot be guaranteed. A more in-depth discussion on both strategies can be found in [113, 137]. Spatial separation between the trajectories can be enforced by imposing the following constraints:

$$\min_{\substack{i,j=1,\ldots,n \\ i \neq j}} \left\| \boldsymbol{p}_{d,i}(t_i) - \boldsymbol{p}_{d,j}(t_j) \right\|^2 \geq E_d^2 \,, \qquad \forall t_i \in [0, t_{f,i}] \,, \forall t_j \in [0, t_{f,j}]. \tag{2.5}$$

Minimum separation between trajectories and obstacles can also be enforced in a similar fashion. Letting $\boldsymbol{p}_{o,k}$ be the position of the kth obstacle, with $k = 1, \ldots, n_o$, where n_o is the number of obstacles, in order to ensure deconfliction with the obstacles the following constraint can be enforced:

$$\min_{\substack{i=1,\ldots,n \\ k=1,\ldots,n_o}} \left\| \boldsymbol{p}_{d,i}(t) - \boldsymbol{p}_{o,k} \right\|^2 \geq E_d^2 \,, \qquad \forall t \in [0, t_{f,i}]. \tag{2.6}$$

Notice that in the above formulation the obstacles are assumed to be static. Moving obstacles can also be considered in the motion planning phase, as long as their trajectories are known a priori.

Mission-specific constraints

Additional constraints can be imposed on Problem 1, depending on the requirements of the mission considered. These constraints can be imposed, for example, to require that all vehicles arrive at their respective final destinations at the same time, or at different times so as to meet a desired inter-vehicle schedule. For simplicity, and without loss of generality, in this thesis we consider the problem of simultaneous arrival, i.e.

$$t_{f,i} - t_{f,j} = 0\,, \qquad i,j = 1,\ldots,n\,, \quad i \neq j\,, \tag{2.7}$$

with the understanding that the above constraints can be easily modified, and that the proposed approach can be employed for a larger set of scenarios.

Optimal motion planning problem

In what follows, we provide a formal definition of Problem 1. Recall that the ith vehicle involved in the cooperative mission is modelled by the system of equations introduced in Equation (2.1), with state $\boldsymbol{x}_{v,i}(t)$ and control input $\boldsymbol{u}_{v,i}(t)$. Let $\boldsymbol{x}_{d,i}(t)$ and $\boldsymbol{u}_{d,i}(t)$ be the vectors of desired states and control inputs for vehicle i. Let $\boldsymbol{x}_d(t) = [\boldsymbol{x}_{d,1}^\top(t),\ldots,\boldsymbol{x}_{d,n}^\top(t)]^\top \in \mathbb{R}^{n_x}$, and $\boldsymbol{u}_d(t) = [\boldsymbol{u}_{d,1}^\top(t),\ldots,\boldsymbol{u}_{d,n}^\top(t)]^\top \in \mathbb{R}^{n_u}$. Then, Problem 1 can be cast into the following optimal control problem with Bolza cost:

Problem 2 *Determine* $\boldsymbol{x}_d : [0,t_f] \to \mathbb{R}^{n_x}$ *and* $\boldsymbol{u}_d : [0,t_f] \to \mathbb{R}^{n_u}$ *(and possibly* t_f*) that minimizes*

$$I(\boldsymbol{x}_d(t),\boldsymbol{u}_d(t)) = E(\boldsymbol{x}_d(0),\boldsymbol{x}_d(t_f)) + \int_0^{t_f} F(\boldsymbol{x}_d(t),\boldsymbol{u}_d(t))dt\,, \tag{2.8}$$

subject to

$$\dot{\boldsymbol{x}}_d(t) = \boldsymbol{f}(\boldsymbol{x}_d(t),\boldsymbol{u}_d(t))\,, \quad \forall t \in [0,t_f] \tag{2.9}$$

$$\boldsymbol{e}(\boldsymbol{x}_d(0),\boldsymbol{x}_d(t_f)) = \boldsymbol{0}\,, \tag{2.10}$$

$$\boldsymbol{h}(\boldsymbol{x}_d(t),\boldsymbol{u}_d(t)) \leq \boldsymbol{0}\,, \quad \forall t \in [0,t_f]\,, \tag{2.11}$$

where $E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$, $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$, $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, $\boldsymbol{e} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_e}$, *and* $\boldsymbol{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$ *are nonlinear functions of their arguments.*

∇

Remark 1 *In Problem 2, the constraint in Equation* (2.9) *enforces the dynamics of the vehicles considered in the mission, with* $\boldsymbol{f} = [\boldsymbol{f}_1^\top, \ldots, \boldsymbol{f}_n^\top]^\top$*, where* $\boldsymbol{f}_i$ *is introduced in Equation* (2.1)*. Equation* (2.10) *describes the boundary conditions and simultaneous time of arrival constraints. The flyability and feasibility constraints are described by Equation* (2.11)*. The functions* E *and* F *in Equation* (2.8) *are the end point cost and the running cost of the Bolza-type cost functional, respectively.*

♦

Remark 2 *The outcome of Problem 2 is a set of optimal states* $\boldsymbol{x}_{d,i}^*(t)$ *and control inputs* $\boldsymbol{u}_{d,i}^*(t)$*,* $i = 1, \ldots, n$*, which, in turn, provide optimal trajectories* $\boldsymbol{p}_{d,i}^*(t)$ *computed as*

$$\boldsymbol{p}_{d,i}^*(t) = \boldsymbol{g}_i(\boldsymbol{x}_{d,i}^*(t))\,,$$

where $\boldsymbol{g}_i$ *is introduced in Equation* (2.1)*.*

♦

2.2.2 Coordinated tracking problem

The coordinated tracking problem is referred to as the problem of enabling a fleet of vehicles to execute collision-free maneuvers while satisfying temporal specifications, such as simultaneous time of arrival. In the adopted framework, it is assumed that the UxSs are equipped with a VT tracking algorithm —implemented on-board the vehicles— responsible for making each vehicle follow a virtual target running along the path generated by the motion planner. Then, the objective is to design distributed coordination control laws that adjust the rate of progression of these virtual targets so as to coordinate the entire fleet. One of the main benefits of this approach lies in the fact that the progression of the virtual targets to be tracked by the vehicles is adjusted on-line, adapting to external disturbances or vehicles' tracking errors. Thus, this approach is more robust than the coordinated trajectory tracking approach, where the coordination task is solved off-line and the UxSs are simply required to follow the trajectories generated by solving Problem 2.

The coordinated tracking problem is solved in three main steps: the first step consists of implementing a virtual target moving along the path computed by the motion planning algorithm described earlier. This objective is achieved by introducing a new parameter, virtual time, denoted here as $\gamma_i(t)$, and letting the virtual target to be tracked by the UxSs be $\boldsymbol{p}_{d,i}(\gamma_i(t))$, where the subscript i refers to the ith UxS involved in the cooperative mission; the second step consists of making each UxS track the virtual target assigned to it. This step, referred to as VT tracking, reduces to driving a suitably defined error vector to zero by using the vehicle's control inputs; third, to enforce the temporal constraints of the mission, a consensus problem is

formulated, in which the objective of the fleet of vehicles is to reach agreement on some distributed variables of interest that capture the objective of the coordination problem.

Virtual time and virtual target

Given the trajectory $\boldsymbol{p}_{d,i}(t)$ produced by the motion planning algorithm described above, we let the virtual time $\gamma_i(t)$, $t \geq 0$, be a temporal variable

$$\gamma_i : \mathbb{R}^+ \to [0, t_f] \,, \quad \text{for all} \quad i = 1 \,, \, ... \,, \, N \,, \tag{2.12}$$

and the virtual target $\boldsymbol{p}_{d,i}(\gamma_i(t))$ be a moving point to be tracked by vehicle i. With this formulation, the virtual time provides an abstraction of actual (clock) time. It can be *stretched* or *compressed* in order to adjust the progression of the virtual target along the path $\boldsymbol{p}_{d,i}(\cdot)$, so as to achieve pre-specified temporal requirements, i.e. coordination. If $\dot{\gamma}_i(t) \equiv 1$, then the progression of the virtual target in time is equal to the progression of the *desired* trajectory generated at the motion planning level. More precisely, assume that $\dot{\gamma}_i(t) = 1 \,, \quad \text{for all} \quad t \in [0, t_f]$, with $\gamma_i(0) = 0$. This assumption implies that $\gamma_i(t) = t$ for all t. In turn, the following equality holds

$$\boldsymbol{p}_{d,i}(\gamma_i(t)) = \boldsymbol{p}_{d,i}(t) \,,$$

which, in other words, means that the desired trajectory generated by the motion planner and the virtual target coincide for all time, and so do their speed and acceleration profiles. On the other hand, $\dot{\gamma}_i > 1$ ($\dot{\gamma}_i < 1$) implies a faster (slower) execution of the mission. This statement becomes evident when expressing the speed and the acceleration of the virtual target in terms of the derivatives of the virtual time:

$$\left\| \frac{d}{dt} \boldsymbol{p}_{d,i}(\gamma_i(t)) \right\| = \| \dot{\boldsymbol{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t)) \| = \left\| \frac{d\, \boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \frac{d\gamma_i(t)}{dt} \right\| = \left\| \frac{d\, \boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \dot{\gamma}_i(t) \right\| \,, \tag{2.13}$$

$$\begin{aligned} \left\| \frac{d}{dt} \left(\frac{d}{dt} \boldsymbol{p}_{d,i}(\gamma_i(t)) \right) \right\| &= \| \ddot{\boldsymbol{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t)), \ddot{\gamma}_i(t)) \| = \left\| \frac{d}{dt} \left(\frac{d\, \boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \frac{d\gamma_i(t)}{dt} \right) \right\| \\ &= \left\| \frac{d^2\, \boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i^2(t)} \dot{\gamma}_i^2(t) + \frac{d\, \boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \ddot{\gamma}_i(t) \right\| \,. \end{aligned} \tag{2.14}$$

Virtual target tracking

Recall that the equations of motion of the UxSs are given by Equation (2.1). Then, the objective of the VT tracking problem is to design a control law for $\boldsymbol{u}_{v,i}(t)$ such that the VT tracking error

$$\boldsymbol{e}_{p,i}(t) \triangleq \boldsymbol{p}_{v,i}(t) - \boldsymbol{p}_{d,i}(\gamma_i(t)) \tag{2.15}$$

converges to a neighborhood of zero. In order for the above problem to be solvable, the dynamics of the virtual target must not violate the flyability limits of the vehicle. As pointed out in Equations (2.13) and (2.14), the speed and acceleration of the virtual target, for example, depend not only on the speed and acceleration profiles of the trajectory $\boldsymbol{p}_{d,i}(t)$ generated by the motion planning algorithm, but also on the derivatives of the virtual time, $\dot{\gamma}_i(t)$ and $\ddot{\gamma}_i(t)$. Therefore, flyability limits on the dynamics of the virtual time must be enforced to ensure that the virtual target can be followed by the vehicle. Here we assume that the virtual target can be tracked by the UxS if the trajectories $\boldsymbol{p}_{d,i}(t)$, $i = 1, \ldots, n$ satisfy the flyability constraints imposed by Problem 2, and the derivatives of the virtual time satisfy

$$\dot{\gamma}_i^{\min} \leq \dot{\gamma}_i(t) \leq \dot{\gamma}_i^{\max}, \qquad |\ddot{\gamma}_i(t)| \leq \ddot{\gamma}_i^{\max}, \tag{2.16}$$

for given $0 \leq \dot{\gamma}_i^{\min} < 1$, $\dot{\gamma}_i^{\max} > 1$ and $\ddot{\gamma}_i^{\max} > 0$. As discussed later, the dynamics of $\gamma_i(t)$ (actually its second derivative $\ddot{\gamma}_i(t)$) are explicitly controlled and used as an extra degree-of-freedom to achieve coordination. Therefore, when deriving control laws for $\ddot{\gamma}_i(t)$, the saturation bounds in Equation (2.16) must be taken into consideration.

Remark 3 *The limits $\dot{\gamma}_i^{\min}, \dot{\gamma}_i^{\max}$ and $\ddot{\gamma}_i^{\max}$ can be derived from the actual flyability constraints of the vehicle under consideration, and the (more conservative) flyability constraints imposed by the motion planning algorithm. To clarify this argument, consider the example of a multirotor UAS introduced in Section 2.2.1, subject to minimum and maximum speed limits $v_i^{\min}$ and $v_i^{\max}$, and maximum acceleration limits $a_i^{\max} \leq g$. Then, the virtual target assigned to the vehicle must satisfy*

$$v_i^{\max} \leq \|\dot{\boldsymbol{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t))\| = \left\| \frac{d\,\boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \dot{\gamma}_i(t) \right\| \leq v_i^{\max}, \tag{2.17}$$

$$\|\ddot{\boldsymbol{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t), \ddot{\gamma}_i(t))\| = \left\| \frac{d^2\,\boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i^2(t)} \dot{\gamma}_i^2(t) + \frac{d\,\boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \ddot{\gamma}_i(t) \right\| \leq a_i^{\max} \leq g, \tag{2.18}$$

where we used Equations (2.13) *and* (2.14)*. We notice that the flyability constraints enforced by the motion planning algorithm (see Equation* (2.3)*) imply*

$$v_{\mathrm{d},i}^{\min} \leq \left\| \frac{d\,\boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \right\| \leq v_{d,i}^{\max}, \qquad \left\| \frac{d^2\,\boldsymbol{p}_{d,i}(\gamma_i(t))}{d\gamma_i^2(t)} \right\| \leq a_{d,i}^{\max},$$

where $v_{d,i}^{\min} \geq v_i^{\min}$, $v_{d,i}^{max} < v_i^{max}$, and $a_{d,i}^{\max} < a_i^{\max}$. Therefore, to ensure that the inequalities in (2.17) *and*

(2.18) *are satisfied, the dynamics of the virtual time must not exceed the limits in* (2.16)*, with*

$$\dot{\gamma}_i^{\min} = \frac{v_i^{\min}}{v_{d,i}^{\min}}\,, \qquad \dot{\gamma}_i^{\max} = \frac{v_i^{\max}}{v_{d,i}^{\max}}\,, \qquad \ddot{\gamma}_i^{\max} = \frac{a_i^{\max} - a_{d,i}^{\max}\left(\frac{v_i^{\max}}{v_{d,i}^{\max}}\right)^2}{v_{d,i}^{\max}}\,. \tag{2.19}$$

The equation above also suggests that the constraints imposed on $\dot{\gamma}_i(t)$ *and* $\ddot{\gamma}_i(t)$ *can be relaxed by enforcing more stringent flyability constraints, e.g. larger* $v_{d,i}^{\min}$ *and smaller* $v_{d,i}^{\max}$ *and* $a_{d,i}^{\max}$*, at the expense of an increase in computational complexity of the motion planning algorithm.*

♦

With this setup, the VT tracking problem can be stated as follows:

Problem 3 *Consider n UxSs with equations of motion given by Equation* (2.1)*, and a set of n virtual targets* $\boldsymbol{p}_{d,i}(\gamma_i(t))$*,* $i = 1, \ldots, n$*, assigned to each vehicle, where* $\boldsymbol{p}_{d,i}(t)$ *are trajectories generated by solving Problem 2, and* $\gamma_i(t)$ *are the virtual times that satisfy the bounds in Equation* (2.16)*. Then, the objective is to design a VT tracking control law for* $\boldsymbol{u}_{v,i}(t)$ *such that the VT tracking error defined in Equation* (2.15) *is uniformly bounded, i.e. there exists a positive constant c, and for every* $a \in (0, c)$ *there exists* $\rho = \rho(a) > 0$*, such that*

$$\|\boldsymbol{e}_p(0)\| \leq a \quad \Longrightarrow \quad \|\boldsymbol{e}_p(t)\| \leq \rho\,, \qquad \forall t \geq 0\,, \tag{2.20}$$

where $\boldsymbol{e}_p = [\boldsymbol{e}_{p,1}^\top, \ldots, \boldsymbol{e}_{p,n}^\top]^\top$*.*

▽

The design of control laws for $\boldsymbol{u}_{v,i}(t)$ that solve Problem 3 involves employing nonlinear analysis and/or robust/adaptive control techniques based upon the knowledge of the nominal model (see Equation (2.1)) which governs the motion of the vehicle under consideration. In [138] and [139], for example, two solutions are presented to the VT tracking problem for multirotor UASs. In particular, in [139] it is assumed that the vehicle is equipped with an autopilot capable of tracking angular rates and total thrust commands. Then, it is shown that the VT tracking controller drives the VT tracking error to a neighborhood of zero even in the case of nonideal tracking performance of the autopilot. An analogous result is obtained in [138], which presents a VT tracking control law for AR.Drone UASs equipped with control systems for Euler-angle and vertical-speed command tracking. Additionally, similar solutions to the VT tracking problem for fixed-wing UASs and ducted-fan UASs have been proposed in [139] and [140], respectively. Since this thesis deals with a general class of UxSs, in the remainder of this manuscript we assume that the vehicles involved in the cooperative missions are equipped with VT tracking controllers that solve Problem 3. Nevertheless, as an

example and for the sake of completeness, the development of a VT tracking controller for multirotor UASs that solves the Problem 3 is reported in Chapter 5.

Coordination among multiple vehicles

This section formulates the coordination problem of a group of n UxSs. Recall that the position of the virtual target assigned to the ith vehicle at time t is given by $\mathbf{p}_{d,i}(\gamma_i(t))$, where $\mathbf{p}_{d,i}(t)$ is the trajectory obtained from the motion planning algorithm, and the parameter $\gamma_i(t)$ is the virtual time. The virtual time and its first time derivative play a crucial role in the coordination problem. In fact, because the desired trajectory assigned to each vehicle satisfies the simultaneous time of arrival constraint in Equation (2.7), then if

$$\gamma_i(t) - \gamma_j(t) = 0\,, \quad \text{for all} \quad i, j \in \{1, \ldots, n\}\,, \quad i \neq j\,, \tag{2.21}$$

at time t, all the virtual targets are coordinated. In addition, if

$$\dot{\gamma}_i(t) - 1 = 0\,, \quad \text{for all} \quad i \in \{1, \ldots, n\}\,, \tag{2.22}$$

then the virtual targets run along the paths at the desired rate of progression computed by the motion planning algorithm. Therefore, Equations (2.21) and (2.22) capture the objective of coordination, and a control law for $\ddot{\gamma}_i(t)$ must be formulated to ensure that these equations are satisfied.

To achieve the coordination objective, information must be exchanged among the vehicles over a supporting communication network. In particular, with the method presented in this thesis, in order to minimize the amount of information that must be interchanged, the UxSs exchange only its own virtual time variable, $\gamma_i(t)$, with each other. The information flow as well as the constraints imposed by the communication topology can be modelled using tools from algebraic graph theory. The reader is referred to [141, 142] for key concepts and details on this topic.

Let $\boldsymbol{L}(t) \in \mathbb{R}^{n\times n}$ be the Laplacian of the graph $\Gamma(t)$ over which the UxSs communicate. Let $\boldsymbol{Q}_n \in \mathbb{R}^{(n-1)\times n}$ be a matrix such that $\boldsymbol{Q}_n\mathbf{1} = \mathbf{0}$ and $\boldsymbol{Q}_n(\boldsymbol{Q}_n)^\top = \boldsymbol{I}_{n-1}$, with $\mathbf{1}$ being a vector of appropriate dimension whose components are all 1.

Remark 4 *We notice that a matrix $\boldsymbol{Q}_k$ satisfying $\boldsymbol{Q}_k\mathbf{1} = \mathbf{0}$ and $\boldsymbol{Q}_k(\boldsymbol{Q}_k)^\top = \boldsymbol{I}_{k-1}$ can be found recursively as follows:*

$$\boldsymbol{Q}_k = \begin{bmatrix} \sqrt{\frac{k-1}{k}} & -\frac{1}{\sqrt{k(k-1)}}1_{k-1}^\top \\ 0 & \boldsymbol{Q}_{k-1} \end{bmatrix}\,,$$

with initial condition $\boldsymbol{Q}_2 = [\, \frac{1}{\sqrt{2}} \; -\frac{1}{\sqrt{2}} \,]$. *For simplicity, from now on we let* $\boldsymbol{Q} \triangleq \boldsymbol{Q}_n$, *where* n *is the number of vehicles involved in the cooperative mission.*

◆

Given the above notation, we can formulate the following assumption:

Assumption 1 *The* i*-th UxS communicates only with a neighbouring set of vehicles, denoted by* $\mathcal{N}_i(t)$. *The communication between two UxSs is bidirectional with no time delays. The communication network satisfies the (normalized) persistency of excitation (PE)-like assumption [42]:*

$$\frac{1}{nT} \int_t^{t+T} \boldsymbol{Q}\boldsymbol{L}(\tau)\boldsymbol{Q}^\top d\tau \geq \mu \boldsymbol{I}_{n-1}\,, \tag{2.23}$$

where the parameters $T > 0$ *and* $\mu \in (0, 1]$ *represent a measure of the level of connectivity of the communication graph.*

△

Remark 5 *Note that* $\mu \in (0, 1]$ *follows from the fact that* $||\boldsymbol{Q}\boldsymbol{L}(\tau)\boldsymbol{Q}^\top|| \leq n$ *[143].*

◆

Remark 6 *We note that the PE-like condition* (2.23) *requires the communication graph* $\Gamma(t)$ *to be connected only in an integral sense, not pointwise in time. As a matter of fact, the graph may be disconnected during some interval of time or may even fail to be connected at all times. In this sense, it is general enough to capture packet dropouts, loss of communication, and switching topologies.*

◆

With the above setup, the coordination problem can now be defined as follows:

Problem 4 (Coordination Problem) *Consider* n *UxSs with equations of motion given by Equation* (2.1), *and a set of* n *trajectories* $\boldsymbol{p}_{d,i}(t)$, $i = 1, \ldots, n$ *generated by solving Problem 2. Assume that the UxSs are equipped with a VT tracking controller that solves Problem 3, and assume that the vehicles communicate over a network that satisfies Assumption 1. Then, the objective of coordination is to design feedback control laws for* $\ddot{\gamma}_i(t)$ *for all vehicles such that:*

- *there exist a class* $\mathcal{KL}$ *function* $\beta_{cd}(\cdot)$ *and a class* $\mathcal{K}$ *function* $\alpha_{cd}(\cdot)$ *such that, for every pair of vehicles* i *and* j, $i, j \in \{1, \ldots, n\}$, $i \neq j$, *the coordination errors* $(\gamma_i(t) - \gamma_j(t))$ *and* $(\dot{\gamma}_i(t) - 1)$ *satisfy*

$$|\gamma_i(t) - \gamma_j(t)| \leq \beta_{cd}(||\boldsymbol{x}_{cd}(0)||, t) + \alpha_{cd}\left(\sup_{0 \leq \tau \leq t} (||\boldsymbol{e}_{\boldsymbol{p}}(\tau)||\right)$$

$$|\dot{\gamma}_i(t) - 1| \leq \beta_{cd}(||\boldsymbol{x}_{cd}(0)||, t) + \alpha_{cd}\left(\sup_{0\leq\tau\leq t}(||\boldsymbol{e}_{\boldsymbol{p}}(\tau)||\right)$$

where $\boldsymbol{x}_{cd}(0)$ *is a vector that characterizes the initial coordination error of the group of vehicles; and*

- *the dynamics of the virtual time do not violate the inequalities given by Equation* (2.16)*, i.e.*

$$\dot{\gamma}_i^{\min} \leq \dot{\gamma}_i(t) \leq \dot{\gamma}_i^{\max}, \qquad |\ddot{\gamma}_i(t)| \leq \ddot{\gamma}_i^{\max},$$

for given $0 \leq \dot{\gamma}_i^{\min} < 1$*,* $\dot{\gamma}_i^{\max} > 1$ *and* $\ddot{\gamma}_i^{\max} > 0$

$\triangledown$

Chapter 3

Optimal motion planning for differentially flat systems

This chapter presents a computational framework to efficiently generate feasible and optimal trajectories for differentially flat autonomous vehicle systems. The optimal motion planning problem defined in Section 2.2.1 is transcribed into a calculus of variations problem using the properties of differential flatness. The latter is then approximated into a discrete-time problem using Bernstein polynomials. A rigorous analysis is provided that shows that the solution to the discrete-time approximation converges to the solution to the continuous time calculus of variations problem. The advantages of the proposed method are investigated through numerical examples.

3.1 Optimal motion planning as a calculus of variations problem

This section deals with the problem of optimal motion planning, Problem 2, introduced in Section 2.2.1, for differentially flat systems. For ease of notation, we drop the subscript d from the variables involved in Problem 2, and restate the problem at hand as follows:

Problem 5 (Problem $P^{\mathbf{OC}}$) *Determine* $\boldsymbol{x}(t) : [0, t_f] \to \mathbb{R}^{n_x}$ *and* $\boldsymbol{u}(t) : [0, t_f] \to \mathbb{R}^{n_u}$ *(and possibly* t_f*) that minimize*

$$I(\boldsymbol{x}(t), \boldsymbol{u}(t)) = E(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) + \int_0^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))dt\,, \tag{3.1}$$

subject to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))\,, \quad \forall t \in [0, t_f] \tag{3.2}$$

$$\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{0}\,, \tag{3.3}$$

$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}\,, \quad \forall t \in [0, t_f]\,, \tag{3.4}$$

where $E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$, $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$, $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, $\boldsymbol{e} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_e}$, *and* $\boldsymbol{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$.

∇

In this chapter we assume that the system given by Equation (3.2) is differentially flat. Thus, there exists a flat output $\boldsymbol{y} : [0, t_f] \to \mathbb{R}^{n_y}$, and nonlinear functions $\boldsymbol{\varphi}(\cdot), \boldsymbol{\varphi}_1(\cdot)$, and $\boldsymbol{\varphi}_2(\cdot)$, such that

$$\boldsymbol{y}(t) = \boldsymbol{\varphi}(\boldsymbol{x}(t), \boldsymbol{u}(t), \dot{\boldsymbol{u}}(t), \ldots, \boldsymbol{u}^{(s)}(t)) \,,$$

and

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{\varphi}_1(\boldsymbol{y}(t), \dot{\boldsymbol{y}}(t), \ldots, \boldsymbol{y}^{(r-1)}(t)) \,, \\ \boldsymbol{u}(t) &= \boldsymbol{\varphi}_2(\boldsymbol{y}(t), \dot{\boldsymbol{y}}(t), \ldots, \boldsymbol{y}^{(r)}(t)) \,, \end{aligned} \tag{3.5}$$

see [134]. It follows that the optimal control problem, Problem P^{OC}, can be transcribed as a calculus of variations problem, here referred to as Problem P^{CV}. Letting

$$\boldsymbol{z}(t) = [\boldsymbol{y}(t)^\top, \dot{\boldsymbol{y}}(t)^\top, \ldots, \boldsymbol{y}^{(r)}(t)^\top]^\top \in \mathbb{R}^{(r+1)n_y} \,, \tag{3.6}$$

Problem P^{CV} can be stated as follows:

Problem 6 (Problem P^{CV}) *Determine $\boldsymbol{y}(t)$ (and possibly t_f) that minimizes*

$$\tilde{I}(\boldsymbol{y}(t)) = \tilde{E}(\boldsymbol{z}(0), \boldsymbol{z}(t_f)) + \int_0^{t_f} \tilde{F}(\boldsymbol{z}(t)) dt \,, \tag{3.7}$$

subject to

$$\tilde{\boldsymbol{e}}(\boldsymbol{z}(0), \boldsymbol{z}(t_f)) = \boldsymbol{0} \,, \tag{3.8}$$

$$\tilde{\boldsymbol{h}}(\boldsymbol{z}(t)) \leq \boldsymbol{0} \,, \quad \forall t \in [0, t_f] \,, \tag{3.9}$$

where $\tilde{E}(\boldsymbol{z}(0), \boldsymbol{z}(t_f))$, $\tilde{F}(\boldsymbol{z}(t))$, $\tilde{\boldsymbol{e}}(\boldsymbol{z}(0), \boldsymbol{z}(t_f))$, and $\tilde{\boldsymbol{h}}(\boldsymbol{z}(t))$ are obtained by expressing the functions $E(\boldsymbol{x}(0), \boldsymbol{x}(t_f))$, $F(\boldsymbol{x}(t), \boldsymbol{u}(t))$, $\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f))$, and $\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t))$ in terms of the flat output using the maps $\boldsymbol{\varphi}_1(\cdot)$ and $\boldsymbol{\varphi}_2(\cdot)$ introduced in Equation (3.5)*, i.e.*

$$E(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = E(\, \boldsymbol{\varphi}_1(\boldsymbol{y}(0), \dot{\boldsymbol{y}}(0), \ldots, \boldsymbol{y}^{(r-1)}(0)) \,, \, \boldsymbol{\varphi}_1(\boldsymbol{y}(t_f), \dot{\boldsymbol{y}}(t_f), \ldots, \boldsymbol{y}^{(r-1)}(t_f)) \,) = \tilde{E}(\boldsymbol{z}(0), \boldsymbol{z}(t_f)) \,,$$

$$F(\boldsymbol{x}(t), \boldsymbol{u}(t)) = F(\, \boldsymbol{\varphi}_1(\boldsymbol{y}(t), \dot{\boldsymbol{y}}(t), \ldots, \boldsymbol{y}^{(r-1)}(t)) \,, \, \boldsymbol{\varphi}_2(\boldsymbol{y}(t), \dot{\boldsymbol{y}}(t), \ldots, \boldsymbol{y}^{(r)}(t)) \,) = \tilde{F}(\boldsymbol{z}(t)) \,,$$

$$\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{e}(\, \boldsymbol{\varphi}_1(\boldsymbol{y}(0), \dot{\boldsymbol{y}}(0), \ldots, \boldsymbol{y}^{(r-1)}(0)) \,, \, \boldsymbol{\varphi}_1(\boldsymbol{y}(t_f), \dot{\boldsymbol{y}}(t_f), \ldots, \boldsymbol{y}^{(r-1)}(t_f)) \,) = \tilde{\boldsymbol{e}}(\boldsymbol{z}(0), \boldsymbol{z}(t_f)) \,,$$

$$\boldsymbol{h}(\boldsymbol{x}(t),\boldsymbol{u}(t)) = \boldsymbol{h}(\,\boldsymbol{\varphi}_1(\boldsymbol{y}(t),\dot{\boldsymbol{y}}(t),\ldots,\boldsymbol{y}^{(r-1)}(t))\,,\,\boldsymbol{\varphi}_2(\boldsymbol{y}(t),\dot{\boldsymbol{y}}(t),\ldots,\boldsymbol{y}^{(r)}(t))\,) = \tilde{\boldsymbol{h}}(\boldsymbol{z}(t))\,.$$

$\triangledown$

Imposed onto Problem P^{CV} are the following assumptions.

Assumption 2 *$\tilde{E}$, $\tilde{F}$, $\tilde{\boldsymbol{e}}$, and $\tilde{\boldsymbol{h}}$ are Lipschitz continuous with respect to their arguments; $\tilde{F} \in \mathcal{C}^2$.*

$\triangle$

Assumption 3 *An optimal solution $\boldsymbol{y}^*(t)$ to Problem P^{CV} exists and satisfies $\boldsymbol{y}^*(t) \in \mathcal{C}^{r+2}_{n_y}$.*

$\triangle$

3.2 Bernstein approximation

Let $0 = t_0 < t_1 < \ldots < t_N = t_f$ be a set of equidistant *time nodes*, i.e. $t_j = j\frac{t_f}{N}$, $j = 0,\ldots,N$. Consider the following Nth order Bernstein polynomial:

$$\boldsymbol{y}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_j b_{j,N}(t)\,, \qquad t \in [0,t_f]\,, \tag{3.10}$$

where $\boldsymbol{c}_j$, $j = 0,\ldots,N$, are the Bernstein coefficients, and $b_{j,N}(t)$, $j = 0,\ldots,N$, are the Nth order Bernstein polynomial basis

$$b_{j,N}(t) = \binom{N}{j}\frac{t^j(t_f-t)^{N-j}}{t_f^N}\,, \qquad t \in [0,t_f]\,,$$

and

$$\binom{N}{j} = \frac{N!}{j!(N-j)!}$$

(see Appendix A.2). The derivatives of (3.10), namely $\dot{\boldsymbol{y}}_N(t),\ldots,\boldsymbol{y}_N^{(r)}(t)$, can be easily computed using the properties of Bernstein polynomials (see Property 6 or Remark 16 in Appendix A.2.1). Define $\boldsymbol{z}_N(t) = [\boldsymbol{y}_N(t)^\top,\ldots,\boldsymbol{y}_N^{(r)}(t)^\top]^\top$, and $\boldsymbol{c} = [\boldsymbol{c}_0,\ldots,\boldsymbol{c}_N]$. Then, Problem P^{CV} can be approximated as follows.

Problem 7 (Problem $P_N^{\mathbf{CV}}$) *Let $0 < \delta_P < 1$. Determine $\boldsymbol{c}$ (and possibly t_f) that minimizes*

$$\tilde{I}_N(\boldsymbol{c}) = \tilde{E}(\boldsymbol{z}_N(0),\boldsymbol{z}_N(t_N)) + w\sum_{j=0}^{N}\tilde{F}(\boldsymbol{z}_N(t_j))\,, \tag{3.11}$$

subject to

$$||\tilde{\boldsymbol{e}}(\boldsymbol{z}_N(0), \boldsymbol{z}_N(t_N))|| \leq N^{-\delta_P}\,, \tag{3.12}$$

$$\tilde{\boldsymbol{h}}(\boldsymbol{z}_N(t_j)) \leq N^{-\delta_P}\mathbf{1}\,, \quad \forall j = 0, \ldots, N\,, \tag{3.13}$$

with $w = \frac{t_f}{N+1}$.

$\triangledown$

3.3 Feasibility and consistency of the approximation

The outcome of Problem P_N^{CV} is a set of optimal Bernstein coefficients $\boldsymbol{c}^* = [\boldsymbol{c}_0^*, \ldots, \boldsymbol{c}_N^*]$, which determine the optimal Bernstein polynomials (trajectories)

$$\boldsymbol{y}_N^*(t) = \sum_{j=0}^{N} \boldsymbol{c}_j^* b_{j,N}(t)\,. \tag{3.14}$$

In this section we address the following theoretical concerns:

1. the existence of a feasible solution to Problem P_N^{CV},
2. the convergence of $\boldsymbol{y}_N^*(t)$ to the optimal solution of Problem P^{CV}, $\boldsymbol{y}^*(t)$.

The following analysis assumes that the final time in the original optimal control problem is fixed; however, the results can be easily extended to the case where t_f is a decision variable. The main results of this chapter are summarized in Theorems 1 and 2 below.

Theorem 1 *There exists* N_1 *such that for any order of approximation* $N \geq N_1$ *Problem* P_N^{CV} *is feasible.*

■

Proof: The proof of Theorem 1 is given in Appendix B.1.1

♠

Theorem 2 *Assume that* $\boldsymbol{y}_N^*(t)$ *has a uniform accumulation point, i.e. there exists an infinite subset of indices* $V \subset \mathbb{Z}^+$ *such that*

$$\lim_{N \in V} \boldsymbol{y}_N^*(t) = \boldsymbol{y}^{\infty}(t)\,.$$

Assume that $\boldsymbol{y}^{\infty}(t) \in \mathcal{C}_{n_y}^{r+2}$. *Then,* $\boldsymbol{y}^{\infty}(t)$ *is an optimal solution to Problem* P^{CV}.

■

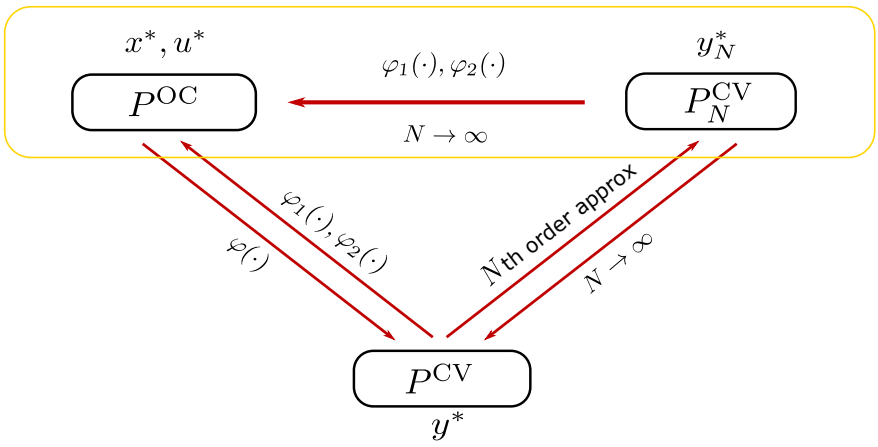


Figure 3.1: Graphical description of the consistency result. Using the differential flatness property of UxSs, the optimal motion planning problem, Problem P^{OC}, can be rewritten as a calculus of variations problem, Problem P^{CV}. Theorems 1 and 2 state that Problem P^{CV} can be approximated into a nonlinear programming problem, Problem P_N^{CV}, the solution of which converges to the solution of Problem P^{CV}. This, in turn, implies that the solution of P_N^{CV} converge to the solution of P^{OC}.

Proof: The proof of Theorem 2 is given in Appendix B.1.2

♠

Remark 7 *By virtue of the differential flatness property of the systems under consideration, Problem P^{CV} is equivalent to Problem P^{OC}. Therefore, Theorem 2 proves the convergence of the approximate solutions to optimal solutions of the original control problem, Problem P^{OC}. Figure 3.1 provides a graphical interpretation of the main contribution of this chapter.*

♦

Remark 8 *The present work focuses on Bernstein polynomial approximation of the trajectories. However, the results reported in Theorems 1 and 2 and their proofs (see Appendix B.1) apply to any approximation or interpolation method that satisfies the convergence properties detailed in Lemmas 4, 5 and 6 in Appendix A.2.2.*

♦

3.4 Illustrative examples

This section describes the benefits of the proposed approach through a simulation example. The results are obtained using MATLAB's built in *fmincon* function. The motion of the vehicle considered is governed by

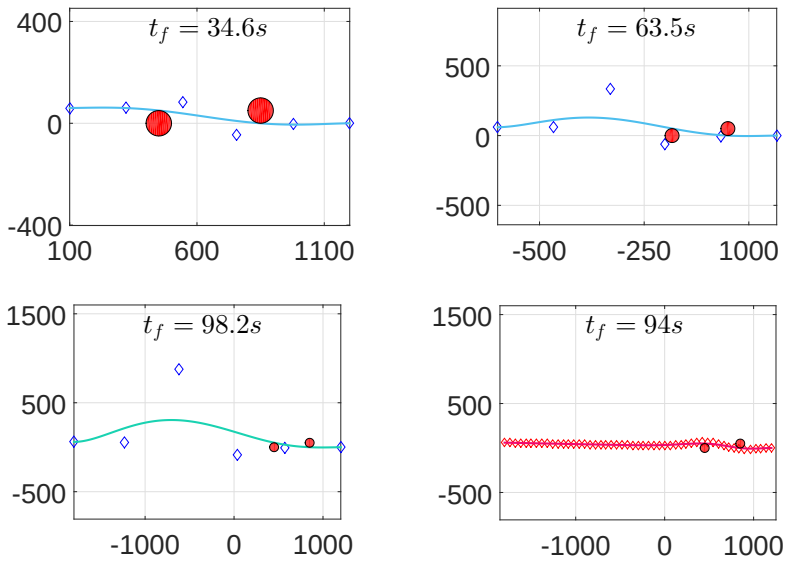


Figure 3.2: Motion planning for 1 vehicle: trajectories (solid lines), Bernstein coefficients (diamonds), and obstacles (red circles). Each plot depicts a mission with different initial positions and number of nodes: $\boldsymbol{y}(0) = [100, 60]^\top$, $N = 5$ (top-left); $\boldsymbol{y}(0) = [-800, 60]^\top$, $N = 5$ (top-right); $\boldsymbol{y}(0) = [-1500, 60]^\top$, $N = 5$ (bottom-left); $\boldsymbol{y}(0) = [-1500, 60]^\top$, $N = 50$ (bottom-right).

the following differential equations

$$\begin{cases} \dot{x}_1(t) &= V(t)\cos(x_3(t)) \\ \dot{x}_2(t) &= V(t)\sin(x_3(t)) \\ \dot{x}_3(t) &= \omega(t)\,, \end{cases} \tag{3.15}$$

with input $\boldsymbol{u}(t) = [V(t)\,,\,\omega(t)]^\top$, and flat output $\boldsymbol{y}(t) = [x_1(t)\,,\,x_2(t)]^\top$. The vehicle is subject to input constraints $V_{\min} \leq V(t) \leq V_{\max}$ and $-\omega_{\max} \leq \omega(t) \leq \omega_{\max}$. Additional constraints must be imposed to avoid collisions with two static obstacles positioned at $\boldsymbol{p}_{o,i}$, $i = 1, 2$. The objective is to generate a trajectory that, starting from a given initial position $\boldsymbol{y}_0$, arrives at the desired final destination $\boldsymbol{y}_f$, satisfies the above constraints, while minimizing the time of arrival. The Bernstein approximation of the flat output $\boldsymbol{y}(t)$ is defined as

$$\boldsymbol{y}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_j b_{j,N}(t) = [x_{1_N}(t)\,,\,x_{2_N}(t)]^\top\,. \tag{3.16}$$

The above problem is transcribed as follows: find $\boldsymbol{c} = [\boldsymbol{c}_0, \ldots, \boldsymbol{c}_N]$ and t_f that minimize $J = \int_0^{t_f} dt$ subject to

$$V_{\min}^2 \leq \dot{x}_{1_N}^2(t) + \dot{x}_{2_N}^2(t) \leq V_{\max}^2\,, \tag{3.17}$$

$$-\omega_{\max} \leq \frac{\dot{x}_{1_N}(t)\ddot{x}_{2_N}(t) - \ddot{x}_{1_N}(t)\dot{x}_{2_N}(t)}{\dot{x}_{1_N}^2(t) + \dot{x}_{2_N}^2(t)} \leq \omega_{\max}\,, \tag{3.18}$$

$$||\boldsymbol{y}_N(t) - \boldsymbol{p}_{o_i}|| \geq E\,, \quad \forall t \in [0, t_f]\,, \tag{3.19}$$

$$\boldsymbol{y}_N(0) = \boldsymbol{y}_0\,, \qquad \boldsymbol{y}_N(t_f) = \boldsymbol{y}_f\,. \tag{3.20}$$

It can be verified that the expression for the square of the speed in Equation (3.17) is a Bernstein polynomial, and the angular rate in Equation (3.18) is a *rational* Bernstein polynomial [111]. The properties of Bernstein polynomials given in Appendix A.2.1 carry over *rational* Bernstein polynomials [113]. Thus, the continuous-time expressions in Equations (3.17), (3.18), and (3.19) can be computed by means of the minimum distance algorithm (see Property 8 in Appendix A.2.1), and the above problem can be solved as a finite dimensional problem. Finally, the constraints in Equation (3.20) can be enforced directly on the first and last Bernstein coefficients, since $\boldsymbol{c}_0 = \boldsymbol{y}_N(0)$ and $\boldsymbol{c}_N = \boldsymbol{y}_N(t_f)$ (Property 1 in Appendix A.2.1 —end-point values).

Figure 3.2 illustrates the results of the proposed approach with $V_{\min} = 15$m/s, $V_{\max} = 32$m/s, $\omega_{\max} = 0.3$rad/s, $E = 50$m, and $\boldsymbol{y}_f = [1200\,,\, 0]^\top$ and for three different initial positions. The objective of this numerical example is to demonstrate that the collision avoidance constraints, for example, are satisfied independently of the length of the trajectory, for fixed (small) order of approximation, $N = 5$. If a more accurate approximation to the optimal solution is required, the number of nodes can be increased (see bottom-left and right plots in Figure 3.2). This is one aspect of the present approach that differs from other discretization methods, such as pseudospectral methods. When using pseudospectral methods, collision avoidance can be guaranteed only at the collocation points [114]. As the length of the path increases, for example, the order of approximation must grow to guarantee separation with the obstacles.

In the next scenario, the same time-optimal motion planning problem for one vehicle depicted in the top-left plot of Figure 3.2 is augmented with four additional vehicles. Each vehicle is required to satisfy the constraints given by Equations (3.17), (3.18), and (3.19), plus *temporal separation* between each pair of trajectories for inter-vehicle safety, i.e.

$$||\boldsymbol{y}_{i_N}(t) - \boldsymbol{y}_{j_N}(t)|| \geq E\,, \tag{3.21}$$

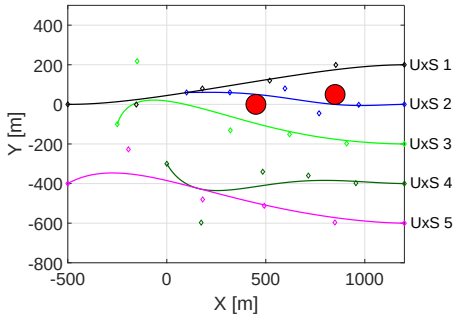
$\forall i, j = 1, \ldots, 5$, $i \neq j$ $\forall t \in [0, t_f]$, where $\boldsymbol{y}_{i_N}(t)$ is the Bernstein approximant of the flat output of vehicle i computed as in Equation (3.16). Similarly to the previous example, the above constraints can be efficiently computed using the minimum distance algorithm, and inter-vehicle safety can be guaranteed for the entire trajectories for any order of approximation. Figure 3.3 shows the results of the proposed method with $N = 5$. The 2D trajectories are depicted in Figure 3.3a. Figures 3.3b and 3.3c depict the speed and angular rates, respectively, which remain within the lower and upper limits. The temporal separation between the

trajectories is shown in Figure 3.4. The optimal time of arrival is $t_f \approx 55.01$s.

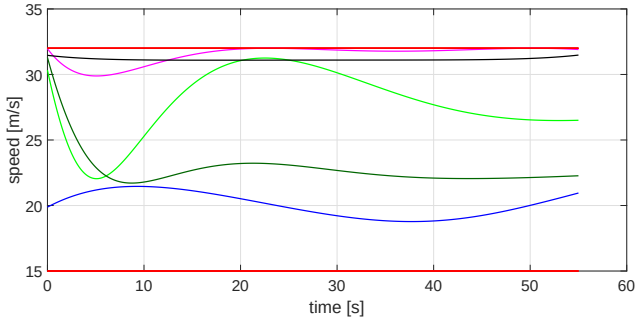
The advantages of the proposed method become even more evident when *spatial separation* constraints, i.e.

$$||\boldsymbol{y}_{i_N}(t_i) - \boldsymbol{y}_{j_N}(t_j)|| \geq E\,, \tag{3.22}$$

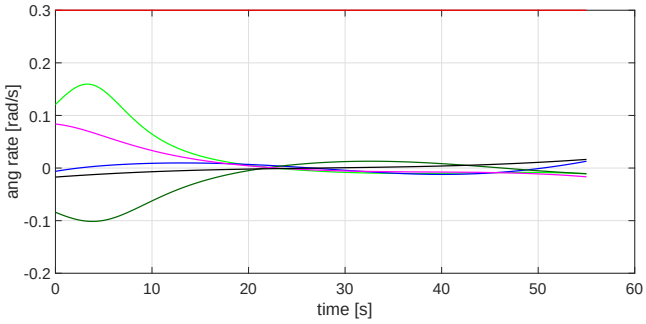
$\forall i, j = 1, \ldots, 5,\ i \neq j\ \forall t_i, t_j \in [0, t_f]$, must be enforced instead of temporal separation. Notice that with the above constraints, the trajectories are required to be separated by E for all times and cannot intersect. If one had to enforce (3.22) using pseudospectral methods, separation should be enforced between each node of each trajectory, resulting in $10N^2$ spatial separation constraints. It is clear that an increase of the number of nodes for safety could jeopardize the computational appeal of the method. On the other hand, when using the method proposed in this chapter, not only spatial separation can be guaranteed with low orders of approximation, but also, to obtain more accurate approximations, the order of approximation can be scaled up without drastically increasing the complexity of the NLP, i.e. the number of constraints is independent of N. Figure 3.5 depicts the results obtained by enforcing the spatial separation constraints instead of the less conservative temporal separation constraints. In Figure 3.5a the order of approximation is set to $N = 5$, while $N = 40$ in Figure 3.5b. The optimal time of arrivals are $t_f \approx 55.62$s and $t_f \approx 53.73$s, respectively. Finally, Figures 3.6 and 3.7 illustrate the spatial separation between the trajectories when temporal separation and spatial separation, respectively, are enforced. In particular, Figure 3.6 depicts the spatial separation of the trajectories introduced in Figure 3.3a. It can be clearly noticed that the spatial separation between the pairs of trajectories exceeds the lower limit $E = 50m$, even though the trajectories are temporally separated (Figure 3.4). On the other hand, the spatial separation between the trajectories depicted in Figure 3.5a always satisfies the minimum spatial separation requirement, as demonstrated by Figure 3.7.



(a) 2D Plot. $N = 5$, $t_f \approx 55.01$s.

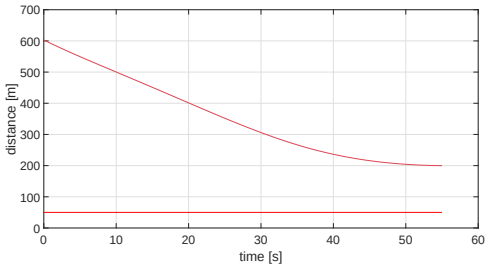


(b) Vehicles' speed.

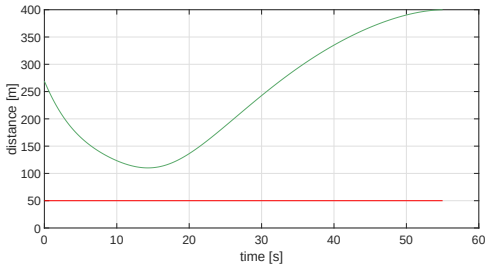


(c) Vehicles' angular rate.

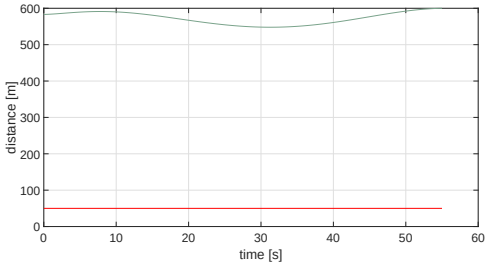
Figure 3.3: Motion planning for 5 vehicles. Temporal Separation.



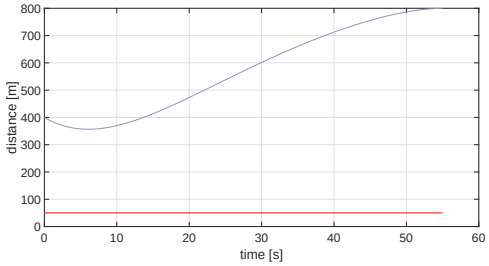
(a) Distance between UxSs 1 and 2.



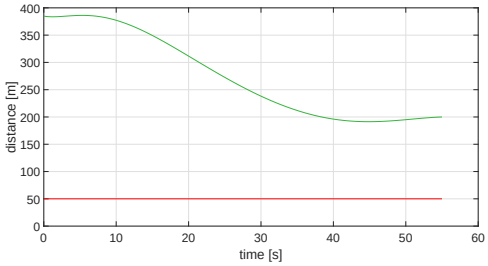
(b) Distance between UxSs 1 and 3.



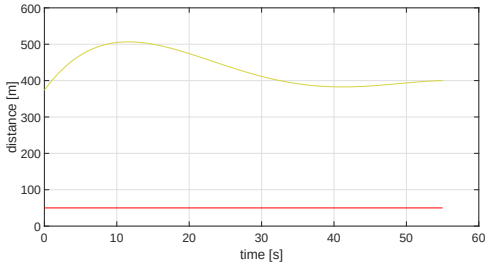
(c) Distance between UxSs 1 and 4.



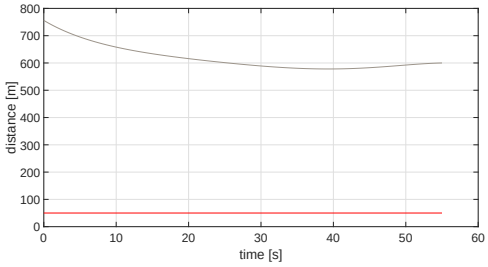
(d) Distance between UxSs 1 and 5.



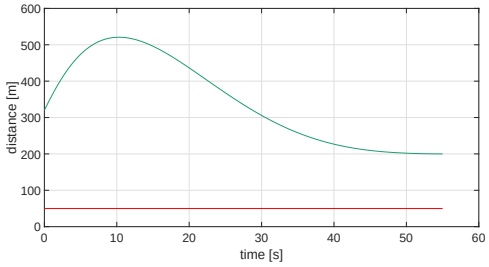
(e) Distance between UxSs 2 and 3.



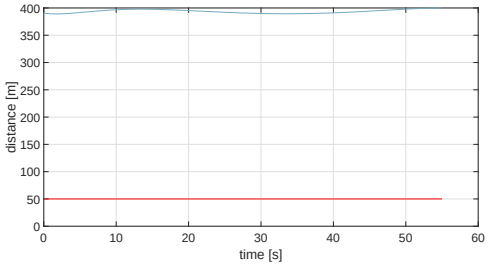
(f) Distance between UxSs 2 and 4.



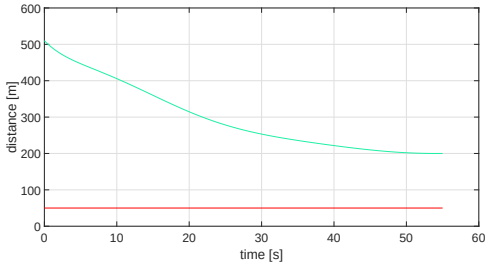
(g) Distance between UxSs 2 and 5.



(h) Distance between UxSs 3 and 4.

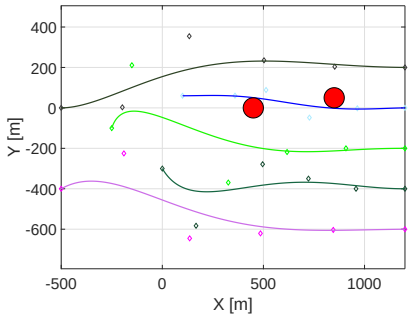


(i) Distance between UxSs 3 and 5.

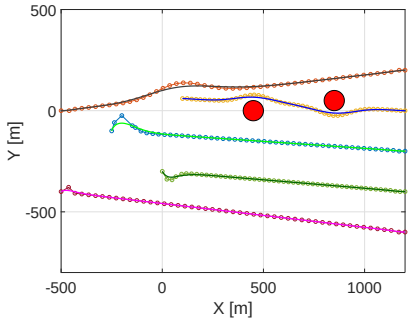


(j) Distance between UxSs 4 and 5.

Figure 3.4: Temporal separation between the UxSs (referred to the mission depicted in Figure 3.3a)

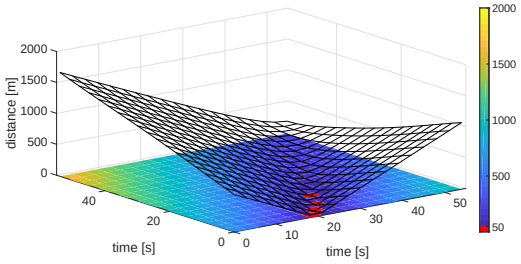


(a) 2D Plot. $N = 5$, $t_f \approx 55.62$s.

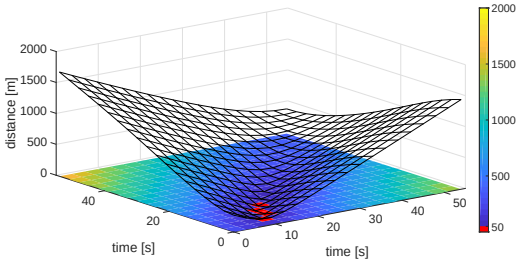


(b) 2D Plot. $N = 40$, $t_f \approx 53.73$s.

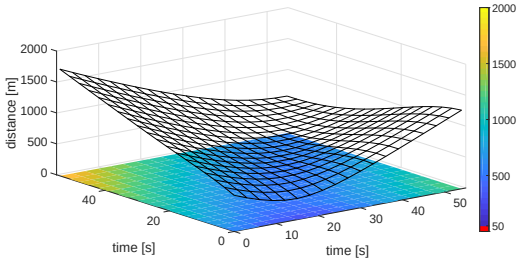
Figure 3.5: Motion planning for 5 vehicles. Spatial separation.



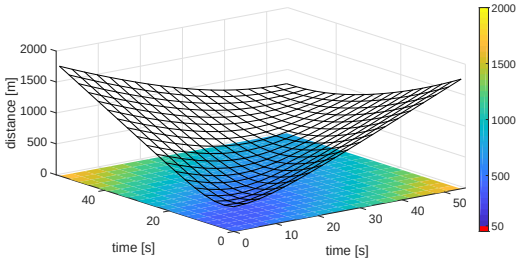
(a) Distance between UxSs 1 and 2.



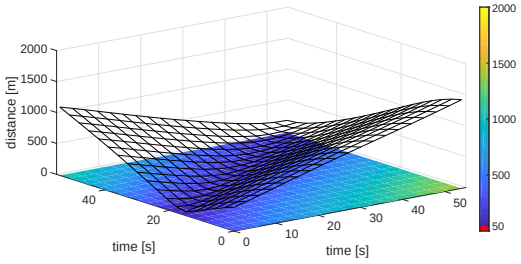
(b) Distance between UxSs 1 and 3.



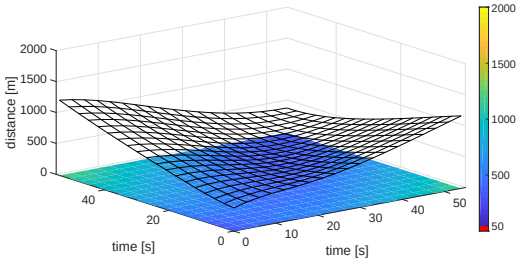
(c) Distance between UxSs 1 and 4.



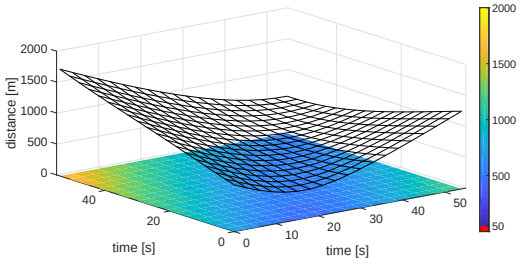
(d) Distance between UxSs 1 and 5.



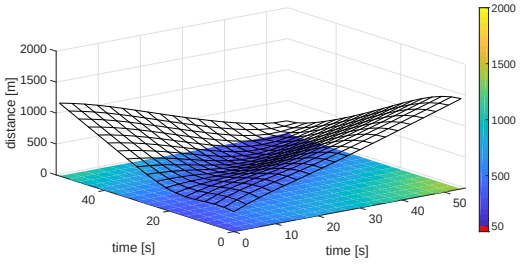
(e) Distance between UxSs 2 and 3.



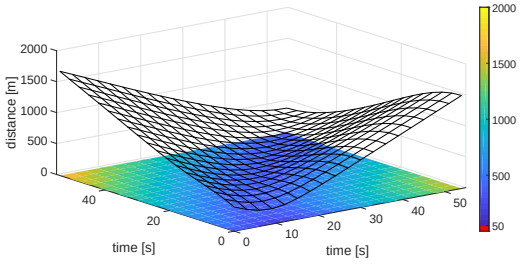
(f) Distance between UxSs 2 and 4.



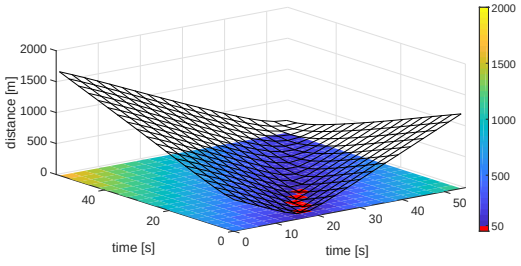
(g) Distance between UxSs 2 and 5.



(h) Distance between UxSs 3 and 4.

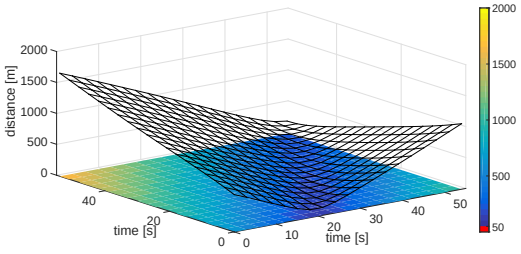


(i) Distance between UxSs 3 and 5.

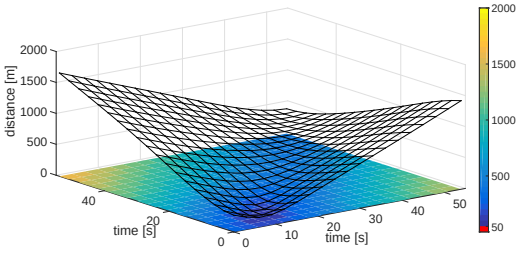


(j) Distance between UxSs 4 and 5.

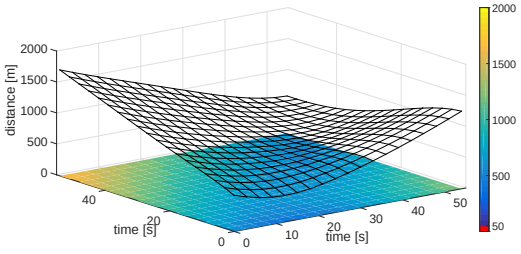
Figure 3.6: Spatial separation between the UxSs when temporal separation is enforced (referred to the mission depicted in Figure 3.3a)



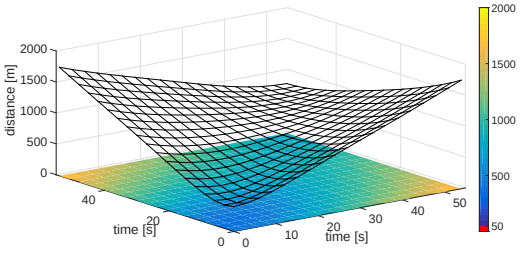
(a) Distance between UxSs 1 and 2.



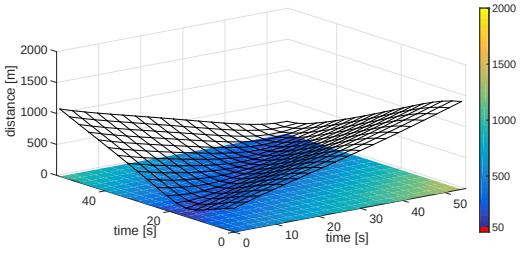
(b) Distance between UxSs 1 and 3.



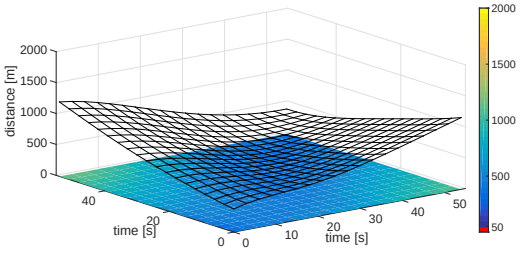
(c) Distance between UxSs 1 and 4.



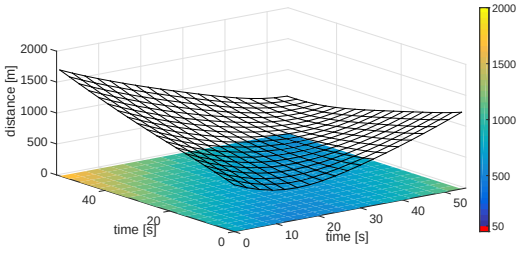
(d) Distance between UxSs 1 and 5.



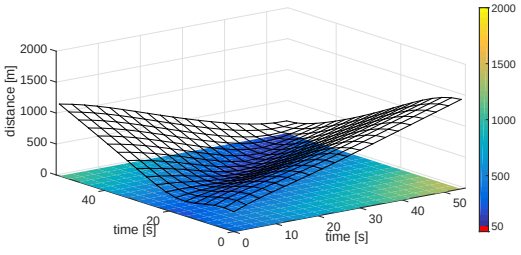
(e) Distance between UxSs 2 and 3.



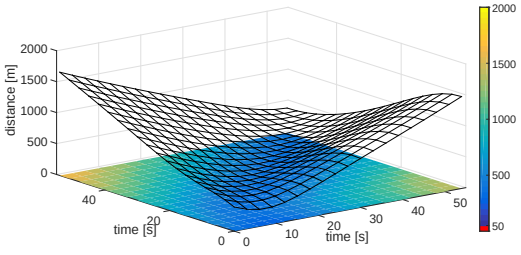
(f) Distance between UxSs 2 and 4.



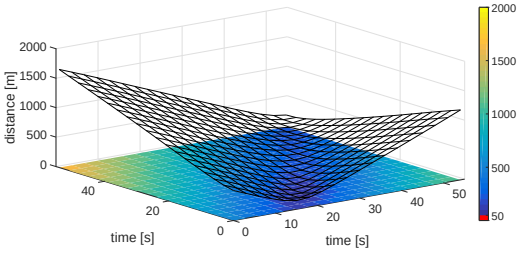
(g) Distance between UxSs 2 and 5.



(h) Distance between UxSs 3 and 4.



(i) Distance between UxSs 3 and 5.



(j) Distance between UxSs 4 and 5.

Figure 3.7: Spatial separation between the UxSs when spatial separation is enforced (referred to the mission depicted in Figure 3.5a)

Chapter 4

Optimal motion planning

This chapter extends the results presented in Chapter 3 to a more general class of optimal motion planning problems. In particular, while in Chapter 3 it was assumed that the UxSs under consideration are differentially flat, in the present chapter we consider UxSs governed by general nonlinear dynamics. In this case, the optimal motion planning problem is captured by a general optimal control problem of Bolza type. Similarly to Chapter 3, this chapter provides an approximation of the optimal control problem into a nonlinear programming problem using Bernstein polynomials, and it demonstrates consistency results for this approximation method. Finally, it presents numerical results that demonstrate the advantages of the proposed approach.

4.1 Optimal motion planning for a general class of systems

This chapter deals with the optimal motion planning problem introduced in Section 2.2.1, Problem P^{OC}. For the reader's convenience, the problem at hand is reported below.

Problem 8 (Problem $P^{\mathbf{OC}}$) *Determine $\boldsymbol{x} : [0, t_f] \to \mathbb{R}^{n_x}$ and $\boldsymbol{u} : [0, t_f] \to \mathbb{R}^{n_u}$ (and possibly t_f) that minimize*

$$I(\boldsymbol{x}(t), \boldsymbol{u}(t)) = E(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) + \int_0^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))dt \tag{4.1}$$

subject to

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall t \in [0, t_f], \tag{4.2}$$

$$\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{0}, \tag{4.3}$$

$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}, \quad \forall t \in [0, t_f], \tag{4.4}$$

where $E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$, $F : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ are the terminal and running costs, respectively, $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the system dynamics, $\boldsymbol{e} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_e}$ is the vector of boundary conditions, and $\boldsymbol{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$ is the vector of state and input constraints.

∇

Notice that for ease of notation we dropped the subscript 'd' from the variables involved in the above problem.

Imposed onto Problem P^{OC} are the following assumptions:

Assumption 4 *E, F, $\boldsymbol{f}$, $\boldsymbol{e}$, and $\boldsymbol{h}$ are Lipschitz continuous with respect to their arguments. $F(\cdot) \in \mathcal{C}^2$.*

$\triangle$

Assumption 5 *Optimal state $\boldsymbol{x}^*(t)$ and control $\boldsymbol{u}^*(t)$ to Problem P^{OC} exist and satisfy $\dot{\boldsymbol{x}}^*(t) \in \mathcal{C}^2_{n_x}$ and $\boldsymbol{u}^*(t) \in \mathcal{C}^2_{n_u}$.*

$\triangle$

Notice that the above problem definition is general enough to support several applications, including trajectory planning for multiple vehicles with dynamics given by Equation (4.2), which is the focus of this chapter.

4.2 Bernstein approximation

The purpose of this section is to formulate a discretized version of Problem P^{OC}, here referred to as Problem P_N^{OC}, where N denotes the *order of approximation*. In order to accomplish this, we must approximate the components that make up Problem P^{OC}, which are the input and the state, the cost function, the system dynamics, and the equality and inequality constraints.

Similarly to Chapter 3, we start by considering the following Nth order Bernstein polynomials:

$$\boldsymbol{x}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_{j,x} b_{j,N}(t)\,, \quad \boldsymbol{u}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_{j,u} b_{j,N}(t), \tag{4.5}$$

with $\boldsymbol{x}_N : [0, t_f] \to \mathbb{R}^{n_x}$, $\boldsymbol{u}_N : [0, t_f] \to \mathbb{R}^{n_u}$, $\boldsymbol{c}_{j,x} \in \mathbb{R}^{n_x}$ and $\boldsymbol{c}_{j,u} \in \mathbb{R}^{n_u}$. Let $\boldsymbol{c}_x \in \mathbb{R}^{n_x \times (N+1)}$ and $\boldsymbol{c}_u \in \mathbb{R}^{n_u \times (N+1)}$ be defined as

$$\boldsymbol{c}_x = [\boldsymbol{c}_{0,x}\,,\, \ldots\,,\, \boldsymbol{c}_{N,x}], \qquad \boldsymbol{c}_u = [\boldsymbol{c}_{0,u}\,,\, \ldots\,,\, \boldsymbol{c}_{N,u}].$$

The derivative of $\boldsymbol{x}_N(t)$ is computed as follows

$$\dot{\boldsymbol{x}}_N(t) = \sum_{j=0}^{N} \left(\sum_{i=0}^{N} \boldsymbol{c}_{i,x} D_{ij} \right) b_{j,N}(t)\,,$$

where D_{ij} is the (i,j) entry of the differentiation matrix $\boldsymbol{D} \in \mathbb{R}^{(N+1)\times(N+1)}$, which can be computed using the properties of Bernstein polynomials (see Properties 4 and 6, and Remark 16 in Appendix A.2.1). Let $0 = t_0 < t_1 < \ldots < t_N = t_f$ be a set of equidistant *time nodes*, i.e. $t_j = j\frac{t_f}{N}$, $j = 0, \ldots, N$. Then, Problem P_N^{OC} can be stated as follows.

Problem 9 (Problem P_N^{OC}) *Let $0 < \delta_P < 1$. Determine $\boldsymbol{c}_x$ and $\boldsymbol{c}_u$ (and possibly t_f) that minimize*

$$I_N(\boldsymbol{c}_x, \boldsymbol{c}_u) = E(\boldsymbol{x}_N(0), \boldsymbol{x}_N(t_N)) + w \sum_{j=0}^{N} F(\boldsymbol{x}_N(t_j), \boldsymbol{u}_N(t_j)) \,, \tag{4.6}$$

subject to

$$\|\dot{\boldsymbol{x}}_N(t_j) - \boldsymbol{f}(\boldsymbol{x}_N(t_j), \boldsymbol{u}_N(t_j))\| \leq N^{-\delta_P} \,, \tag{4.7}$$

$$\forall j = 0, \ldots, N \,,$$

$$\boldsymbol{e}(\boldsymbol{x}_N(0), \boldsymbol{x}_N(t_N)) = \boldsymbol{0} \,, \tag{4.8}$$

$$\boldsymbol{h}(\boldsymbol{x}_N(t_j), \boldsymbol{u}_N(t_j)) \leq N^{-\delta_P}\boldsymbol{1} \,, \quad \forall j = 0, \ldots, N \,. \tag{4.9}$$

with $w = \frac{t_f}{N+1}$.

∇

4.3 Feasibility and consistency of the approximation

The outcome of Problem P_N^{OC} is a set of optimal Bernstein coefficients $\boldsymbol{c}_x^*$ and $\boldsymbol{c}_u^*$ which determine the Bernstein polynomials $\boldsymbol{x}_N^*(t)$ and $\boldsymbol{u}_N^*(t)$, i.e.

$$\boldsymbol{x}_N^*(t) = \sum_{j=0}^{N} \boldsymbol{c}_{j,x}^* b_{j,N}(t) \,, \quad \boldsymbol{u}_N^*(t) = \sum_{j=0}^{N} \boldsymbol{c}_{j,u}^* b_{j,N}(t) \,. \tag{4.10}$$

Similarly to Chapter 3, we now address the following theoretical concerns:

1. the existence of a feasible solution to Problem P_N^{OC},
2. the convergence of the optimal solution of Problem P_N^{OC} to the optimal solution of Problem P^{OC}.

For the sake of simplicity the following analysis assumes that the variable t_f in Problem P^{OC} is fixed; however, the results and the proofs can be easily extended to the case where t_f is a decision variable. The main results of this chapter are summarized in Theorems 3 and 4 below.

Theorem 3 *There exists N_1 such that for any order of approximation $N \geq N_1$ Problem P_N^{OC} is feasible.*

■

Proof: The proof of Theorem 3 is given in Appendix B.2.1.

♠

Theorem 4 *Let $\boldsymbol{c}_x^*, \boldsymbol{c}_u^*$ be an optimal solutions to Problem P_N^{OC}, and $\boldsymbol{x}_N^*(t), \boldsymbol{u}_N^*(t)$ be the Bernstein polynomials given by (4.10). Assume $\boldsymbol{x}_N^*(t), \boldsymbol{u}_N^*(t)$ has a uniform accumulation point, i.e. there exists an infinite subset of indices $V \in \mathbb{Z}^+$ such that*

$$\lim_{N \in V} (\boldsymbol{x}_N^*(t), \boldsymbol{u}_N^*(t)) = (\boldsymbol{x}^\infty(t), \boldsymbol{u}^\infty(t)),$$

and assume that $\dot{\boldsymbol{x}}^\infty(t)$ and $\boldsymbol{u}^\infty(t)$ are continuous on $[0, t_f]$. Then, $(\boldsymbol{x}^\infty(t), \boldsymbol{u}^\infty(t))$ is an optimal solution to Problem P^{OC}.

■

Proof: The proof of Theorem 4 is given in Appendix B.2.2.

♠

Remark 9 *Notice that the above results extend the results of Chapter 3 to a more general class of systems with nonlinear dynamics. In fact, the proofs of Theorems 3 and 4 follow similar steps as the proofs of Theorems 1 and 2. However, the dynamics constraint in Problem P_N^{CV} (see Equation (4.7)), which is absent in the problem formulation of Chapter 3, poses additional technical challenges to demonstrate the consistency results stated above. These challenges are addressed in Appendix A.2.*

♦

4.4 Simulation results

This section discusses the benefits of the proposed approach in solving the optimal trajectory generation problem through two simulation scenarios. The results are obtained using MATLAB's built in *fmincon* function.

In the first scenario, the 2D trajectory generation problem for a single vehicle is considered. The vehicle, modelled as a single integrator, is required to navigate from the initial position $\boldsymbol{x}_0 = [-500, -900]$m to the final destination $\boldsymbol{x}_f = [1500, -600]$m while minimizing the time of arrival. The algorithm must ensure a minimum separation of $E = 50$m with three obstacles positioned at $\boldsymbol{p}_{o,1} = [0 - 800]^\top$m, $\boldsymbol{p}_{o,2} = [450 - 750]^\top$m,

and $\boldsymbol{p}_{o,3} = [850 - 730]^\top$m. Finally, the norm of the input must remain within minimum and maximum saturation limits $u_{\min} = 15$m/s and $u_{\max} = 32$m/s. This problem can be formally stated as follows:

Determine $\boldsymbol{x}(t), \boldsymbol{u}(t)$ and t_f that minimize

$$I(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \int_0^{t_f} dt$$

subject to

$$\begin{aligned}
&\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(t)\,, \quad \forall t \in [0, t_f],\\
&\boldsymbol{x}(0) = \boldsymbol{x}_0\,, \quad \boldsymbol{x}(t_f) = \boldsymbol{x}_f,\\
&||\boldsymbol{x}(t) - \boldsymbol{p}_{o,i}|| \geq E\,, \quad \forall t \in [0, t_f], \quad i = 1, 2, 3\,,\\
&u_{\min} \leq ||\boldsymbol{u}(t)|| \leq u_{\max}\,, \quad \forall t \in [0, t_f].
\end{aligned}$$

The discretization method proposed in this chapter is compared to the Legendre PS method based on Lagrange interpolation at Legendre-Gauss-Lobatto nodes (for implementation details the reader is referred to [144]). The results are enclosed in Figure 4.1. The top-left, top-center, and bottom-left figures show the trajectories obtained using the PS method with orders of approximation 5, 20, and 100, respectively. The PS method enforces the constraints only at the discretization nodes, and not in between them. By increasing the number of nodes, the distance between the entire trajectory and the obstacles increases towards the desired value $E = 50m$. However, as demonstrated by the top-right figure, which depicts the distance between the trajectories and the obstacles for the three order of approximations indicated above, the minimum separation constraint is never satisfied. On the other hand, the bottom-center figure shows that with the proposed method, even by choosing a small number of nodes (N=5 in this example), the collision avoidance constraint can be computed for the entire curve using, for example, the algorithm in [112], and is thus guaranteed along the whole trajectory. The bottom-right figure supports this claim by showing that the distance between the trajectory and the obstacles is always greater than the required value.

The possibility of choosing low order of approximations while guaranteeing constraint satisfaction is the strength of our approach, which prioritizes safety and feasibility of the trajectories over optimality. The advantage of our method becomes more evident in multiple vehicles missions, where the trajectories assigned to the vehicles must be spatially or temporally separated. Recall that spatial separation is guaranteed if the minimum distance between any two points on two paths is greater than or equal to a minimum spatial

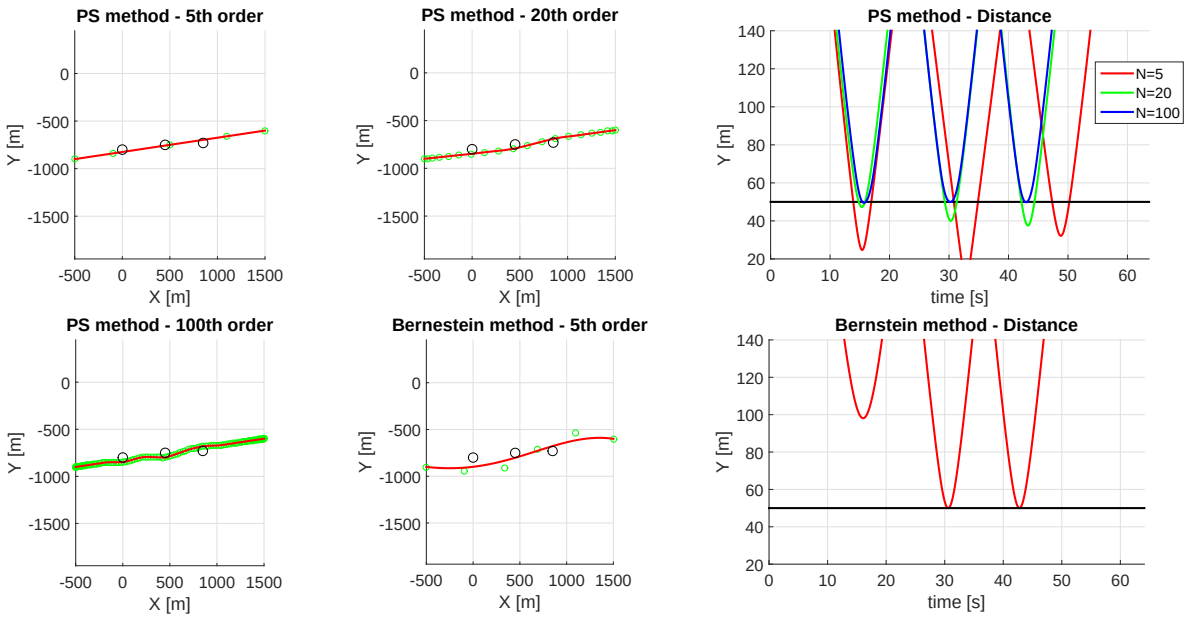


Figure 4.1: Legendre PS vs Bernstein approximation method: collision avoidance with multiple obstacles.

clearance, i.e.

$$\|\boldsymbol{p}_i(t_i) - \boldsymbol{p}_j(t_j)\| \geq E\,, \qquad \forall t_i, t_j \in [0, t_f]\,, \quad \forall i, j = 1, \ldots, n\,, \quad i \neq j. \tag{4.11}$$

Temporal separation, on the other hand, is achieved if for any time t the minimum distance between two vehicles is greater than or equal to the minimum spatial clearance, i.e.

$$\|\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)\| \geq E\,, \qquad \forall t \in [0, t_f], \quad \forall i, j = 1, \ldots, n\,, \quad i \neq j. \tag{4.12}$$

Consider a mission scenario in which n vehicles, starting from their initial positions, have to follow spatially separated trajectories to reach predefined final destinations. By adopting the PS method described above, spatial separation would have to be enforced by imposing separation constraints between every node of every trajectory. Thus, the problem would have $\binom{n}{2}N^2$ separation constraints, where N is the number of nodes and $\binom{n}{2}$ is the binomial coefficient. An increased number of nodes (dictated, perhaps, by reasons similar to the ones discussed in the first simulation scenario of this section), would increase the complexity in the searching for the optimal solution, making the PS approach practically infeasible for these types of applications. On the other hand, with our approach constraint satisfaction is achieved independently on the number of nodes. This is discussed in the next simulation scenario described below.

Figure 4.2 illustrates the results of a multiple vehicles mission in which $n = 11$ vehicles, starting from their

initial positions, have to reach a 'V' shaped formation while minimizing the time of arrival. The Bernstein approximation method is employed with order of approximation $N = 8$. The dynamics of the ith vehicle are governed by the following differential equations

$$\begin{cases} \dot{x}_{1,i}(t) &= V_i(t)\cos(x_{3,i}(t)) \\ \dot{x}_{2,i}(t) &= V_i(t)\sin(x_{3,i}(t)) \\ \dot{x}_{3,i}(t) &= \omega_i(t)\,, \end{cases} \tag{4.13}$$

with input $\boldsymbol{u}_i(t) = [V_i(t)\,,\, \omega_i(t)]^\top$, and state $\boldsymbol{x}_i(t) = [x_{1,i}(t)\,,\, x_{2,i}(t)\,,\, x_{3,i}(t)]^\top$. The input constraints are as follows:

$$V_{\min} \leq V_i(t) \leq V_{\max}\,, \tag{4.14}$$

$$-\omega_{\max} \leq \omega_i(t) \leq \omega_{\max}\,, \tag{4.15}$$

with $V_{\min} = 15$m/s, $V_{\max} = 32$m/s, and $\omega_{\max} = 0.3$rad/s. Finally, temporal separation constraints are imposed between each pair of trajectories, i.e.

$$||\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)|| \geq E\,, \tag{4.16}$$

$\forall i, j = 1, \ldots, 11,\ i \neq j\ \forall t \in [0, t_f]$, where $E = 50$m, and $\boldsymbol{p}_i(t) = [x_{1,i}(t)\,,\, x_{2,i}(t)]^\top$.

The constraints in Equations (4.14), (4.15), and (4.16) are computed using the minimum distance algorithm introduced in Appendix A.2.1, Property 8. Figures 4.3 and 4.4 show the time history of the speeds and angular rates, respectively, demonstrating that the input saturation constraints are satisfied for all times.

At last, the same simulation is repeated, but the temporal separation constraint given by Equation (4.16) is replaced by the (more stringent) spatial separation requirement

$$||\boldsymbol{p}_i(t_k) - \boldsymbol{p}_j(t_p)|| \geq E\,, \tag{4.17}$$

$\forall i, j = 1, \ldots, 11,\ i \neq j\ \forall t_k, t_p \in [0, t_f]$. Figure 4.5 depicts the 2D trajectories. Figures 4.6 and 4.7 illustrate the speed and angular rate commands, respectively.

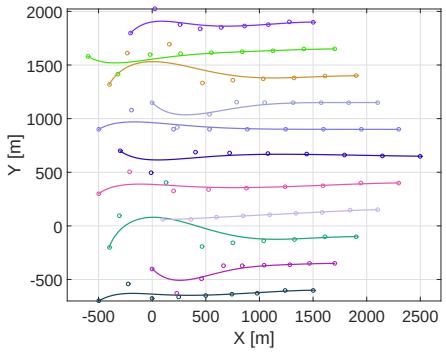


Figure 4.2: Multiple vehicles mission - temporal separation: 2D paths.

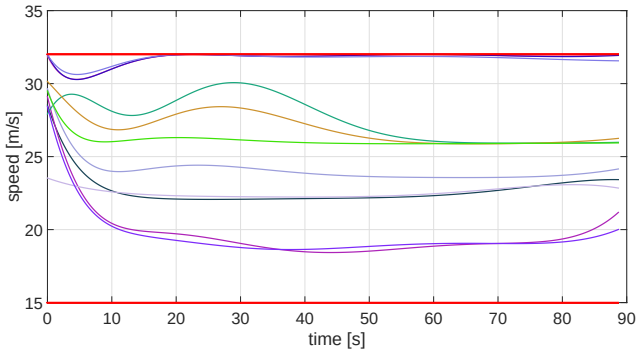


Figure 4.3: Multiple vehicles mission - temporal separation: speed profiles.

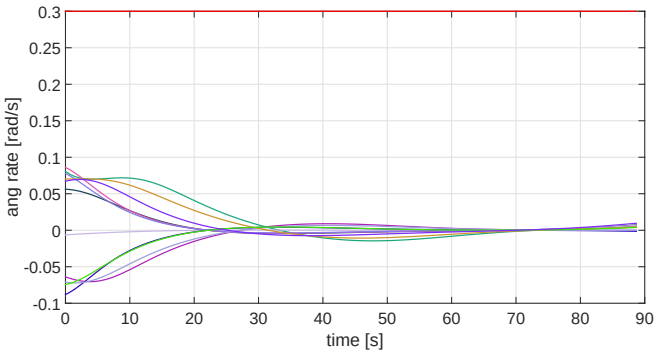


Figure 4.4: Multiple vehicles mission - temporal separation: angular rates.

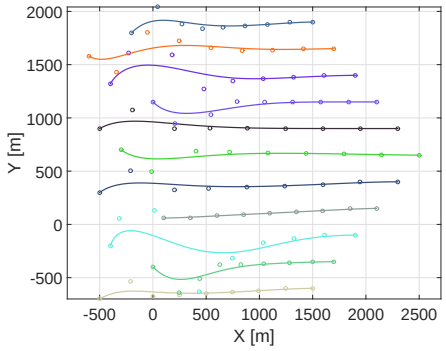


Figure 4.5: Multiple vehicles mission - spatial separation: 2D paths.

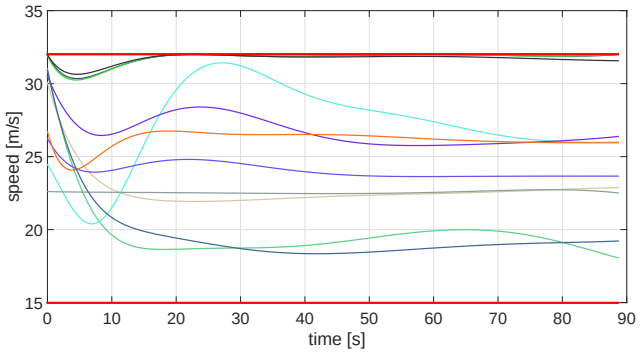


Figure 4.6: Multiple vehicles mission - spatial separation: speed profiles.

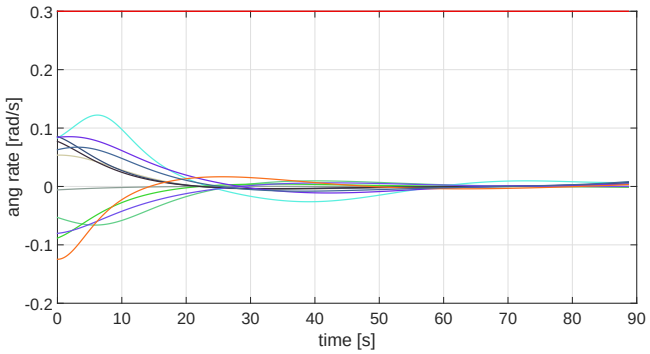


Figure 4.7: Multiple vehicles mission - spatial separation: angular rates.

Chapter 5

Virtual target tracking of multirotor UASs

This chapter describes an outer-loop 3D VT tracking control algorithm for multirotor UASs. The control law derived enables a multirotor equipped with an autopilot capable of tracking angular-rate and thrust reference commands to converge to and follow a virtual target. The method that we propose employs key concepts and techniques introduced in [145] for trajectory tracking of a quadrotor where the total thrust force and the moment vector generated by the four rotors act as control inputs. Since most multirotors today come equipped with an autopilot capable of tracking total thrust and angular-rate commands, the objective of this chapter is to develop a VT tracking control algorithm that exploits this feature by using the autopilot commands as control inputs. A rigorous stability analysis is performed to assess the convergence properties of this algorithm for the cases of ideal and non-ideal tracking performance of the autopilot. The main advantage of this approach is the fact that it can be easily used with a variety of commercially available multirotors.

5.1 Problem formulation

This section formulates the problem of VT tracking for a multirotor UAS, that is the main topic of the present chapter. We first present the equations of motion of a multirotor UAS. This is followed by the definition of a set of VT tracking error variables and the formulation of the VT tracking problem at hand.

5.1.1 6-DoF model for a multirotor UAS

Let $\{\mathcal{I}\}$ denote an inertial frame $\{\hat{\mathbf{e}}_\mathbf{1}, \hat{\mathbf{e}}_\mathbf{2}, \hat{\mathbf{e}}_\mathbf{3}\}$ and $\{\mathcal{B}\}$ the body frame $\{\hat{\mathbf{b}}_\mathbf{1}, \hat{\mathbf{b}}_\mathbf{2}, \hat{\mathbf{b}}_\mathbf{3}\}$ attached to the center of gravity of the multirotor. It is assumed that the vehicle is equipped with an inner-loop autopilot that controls the UAS moments and uses the angular rates and total thrust as command inputs [136, 138]. Furthermore, it is assumed that atmospheric forces are negligible [146, 147]. Then, the six degree of freedom model for the

multirotor UAS kinematics and dynamics is given by

$$\mathcal{G}_v = \begin{cases} \dot{\boldsymbol{p}} & = \boldsymbol{v}\,, \\ m\dot{\boldsymbol{v}} & = T\hat{\mathbf{b}}_{\mathbf{3}} - mg\hat{\mathbf{e}}_{\mathbf{3}}\,, \\ \dot{\boldsymbol{R}} & = \boldsymbol{R}(\boldsymbol{\omega})^{\wedge}\,, \end{cases} \tag{5.1}$$

where $\boldsymbol{p}$ and $\boldsymbol{v}$ are the position and velocity of the vehicle's center of mass in the inertial frame with respect to the basis $\{\hat{\mathbf{e}}_{\mathbf{1}}, \hat{\mathbf{e}}_{\mathbf{2}}, \hat{\mathbf{e}}_{\mathbf{3}}\}$, m is the vehicle's mass, T is the total thrust of the propellers, $\boldsymbol{R} = \boldsymbol{R}_{\boldsymbol{B}}^{\boldsymbol{I}}$ is the rotation matrix from the body frame to the inertial frame, $\boldsymbol{\omega} = \{\boldsymbol{\omega}_{\boldsymbol{B/I}}\}_B = [p,\, q,\, r]$ is the vector of the angular rates of the vehicle with respect to $\{\mathcal{I}\}$, resolved in $\{\mathcal{B}\}$, and $(\cdot)^{\wedge}$ denotes the hat map (see Appendix A.1). The multirotor UAS system is depicted in Figure 5.2a.

5.1.2 Virtual target tracking error

Recall from Section 2.2.2 that $\boldsymbol{p}_{d,i}(\gamma_i(t))$ is the virtual target to be tracked by the ith multirotor UAS involved in the cooperative mission, where $\gamma_i(t)$ is the virtual time defined as

$$\gamma_i : \mathbb{R}^+ \to [0, t_f]\,. \tag{5.2}$$

For ease of notation, and since this chapter deals with the VT tracking problem for a single vehicle, in the remainder of this chapter we drop the subscript i from the variables of interest. Moreover, for the purpose of brevity, we omit the time-dependency.

We start by defining the position error vector $\boldsymbol{e}_p \in \mathbb{R}^3$ as

$$\boldsymbol{e}_p = \boldsymbol{p}_d(\gamma) - \boldsymbol{p}\,, \tag{5.3}$$

and the velocity error vector $\boldsymbol{e}_v \in \mathbb{R}^3$ as

$$\boldsymbol{e}_v = \dot{\boldsymbol{p}}_d(\gamma) - \dot{\boldsymbol{p}}\,. \tag{5.4}$$

Following [145] we now introduce an auxiliary frame $\{\mathcal{D}\}$, which is used (i) to shape the approach to the path as a function of the error components $\boldsymbol{e}_p$ and $\boldsymbol{e}_v$, and (ii) to impose a desired orientation on the vehicle. Let the rotation matrix from the frame $\{\mathcal{D}\}$ to the inertial frame $\{\mathcal{I}\}$ be

$$\boldsymbol{R}_{\boldsymbol{c}} := \boldsymbol{R}_{\boldsymbol{D}}^{\boldsymbol{I}} = [\hat{\mathbf{b}}_{\mathbf{1D}}\,,\, \hat{\mathbf{b}}_{\mathbf{2D}}\,,\, \hat{\mathbf{b}}_{\mathbf{3D}}]\,,$$

where

$$\hat{\mathbf{b}}_{\mathbf{3D}} = \frac{(k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)}{\|(k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)\|}\,, \tag{5.5}$$

for some k_p, $k_v > 1$, and

$$s_p = \begin{cases} \text{sign}(\boldsymbol{e}_p^\top \hat{\mathbf{e}}_{\mathbf{3}})\,, & \text{if} \quad \|k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)\| = 0 \\ 0 & \text{otherwise} \end{cases},$$

$$s_v = \begin{cases} \text{sign}(\boldsymbol{e}_v^\top \hat{\mathbf{e}}_{\mathbf{3}})\,, & \text{if} \quad \|k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)\| = 0 \\ 0 & \text{otherwise} \end{cases}.$$

The vector $\hat{\mathbf{b}}_{\mathbf{3D}}$ defines the desired orientation of the $\hat{\mathbf{b}}_{\mathbf{3}}$-axis of the multirotor required in order to converge to the desired position $\boldsymbol{p}_d(\gamma)$ and velocity $\dot{\boldsymbol{p}}_d(\gamma)$. As an example, Figure 5.1 illustrates the case where the displacement between the UAS position and the virtual target is along the $\hat{\mathbf{e}}_{\mathbf{1}}$ axis; then, the desired orientation of the multirotor's $\hat{\mathbf{b}}_{\mathbf{3}}$-axis is $\hat{\mathbf{b}}_{\mathbf{3D}} = [\hat{\mathbf{b}}_{\mathbf{3D}1}\,,\, 0\,,\, \hat{\mathbf{b}}_{\mathbf{3D}3}]$, i.e. the horizontal component of the $\hat{\mathbf{b}}_{\mathbf{3D}}$ axis along the $\hat{\mathbf{e}}_{\mathbf{2}}$ direction is zero, and the UAS moves solely along the $\hat{\mathbf{e}}_{\mathbf{1}}$ axis, thus reaching the virtual target. Figure 5.1 also shows how different approaches to the path can be obtained by tuning the control gain k_p (for the sake of simplicity in the illustration the value of the control gain k_v is set to 0). The vector $\hat{\mathbf{b}}_{\mathbf{1D}}$ describes the desired orientation of the multirotor's $\hat{\mathbf{b}}_{\mathbf{1}}$ axis, and it can be arbitrarily chosen as long as it is orthonormal to $\hat{\mathbf{b}}_{\mathbf{3D}}$. Consequently, $\hat{\mathbf{b}}_{\mathbf{2D}}$ is chosen to be orthonormal to $\hat{\mathbf{b}}_{\mathbf{1D}}$ and $\hat{\mathbf{b}}_{\mathbf{3D}}$.

Remark 10 *Given the vector* $\hat{\mathbf{b}}_{\mathbf{3D}}$ *defined in* (5.5)*, let* $\hat{\mathbf{b}}^{\star}_{\mathbf{1D}}$ *be a vector that describes the desired orientation of* $\hat{\mathbf{b}}_{\mathbf{1}}$*, but that is not orthonormal to* $\hat{\mathbf{b}}_{\mathbf{3D}}$*. Assume that* $\hat{\mathbf{b}}^{\star}_{\mathbf{1D}}$ *is not parallel to* $\hat{\mathbf{b}}_{\mathbf{3D}}$*. Then,* $\hat{\mathbf{b}}_{\mathbf{1D}}$ *and* $\hat{\mathbf{b}}_{\mathbf{2D}}$ *can be found as follows [145]:*

$$\hat{\mathbf{b}}_{\mathbf{2D}} = \frac{\hat{\mathbf{b}}_{\mathbf{3D}} \times \hat{\mathbf{b}}^{\star}_{\mathbf{1D}}}{||\hat{\mathbf{b}}_{\mathbf{3D}} \times \hat{\mathbf{b}}^{\star}_{\mathbf{1D}}||}\,, \quad \hat{\mathbf{b}}_{\mathbf{1D}} = \hat{\mathbf{b}}_{\mathbf{2D}} \times \hat{\mathbf{b}}_{\mathbf{3D}}\,.$$

Figure 5.2 depicts the geometry of the problem at hand.

♦

Lemma 1 *From the definition of* s_p *and* s_v*, the denominator of* $\hat{\mathbf{b}}_{\mathbf{3D}}$ *is always greater than* 0 *. Therefore the vectors* $\hat{\mathbf{b}}_{\mathbf{1D}}$*,* $\hat{\mathbf{b}}_{\mathbf{2D}}$*, and* $\hat{\mathbf{b}}_{\mathbf{3D}}$ *are always well defined.*

■

Proof: The proof of Lemma 1 is given in Appendix B.3.1.

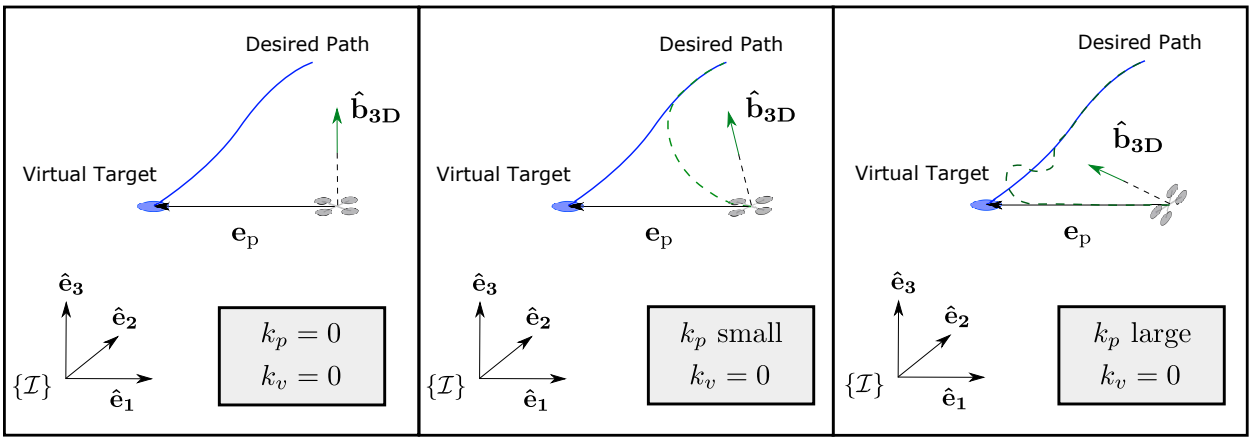


Figure 5.1: Approach to the path.

Next we introduce a set of variables that will be used later to formulate the VT tracking problem. In particular, let $\tilde{\boldsymbol{R}}$ be the rotation matrix from $\{\mathcal{B}\}$ to $\{\mathcal{D}\}$, that is

$$\tilde{\boldsymbol{R}} = \boldsymbol{R}_{\boldsymbol{B}}^{\boldsymbol{D}} = \boldsymbol{R}_{\boldsymbol{c}}^{\top} \boldsymbol{R}\,.$$

Then

$$\dot{\tilde{\boldsymbol{R}}} = \tilde{\boldsymbol{R}}(\tilde{\boldsymbol{\omega}})^{\wedge}\,,$$

where

$$\tilde{\boldsymbol{\omega}} = \{\boldsymbol{\omega}_{\boldsymbol{B/D}}\}_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} - \tilde{\boldsymbol{R}}^{\top}\{\boldsymbol{\omega}_{\boldsymbol{D/I}}\}_D\,, \tag{5.6}$$

and $\{\boldsymbol{\omega}_{\boldsymbol{D/I}}\}_D$ can be computed as follows:

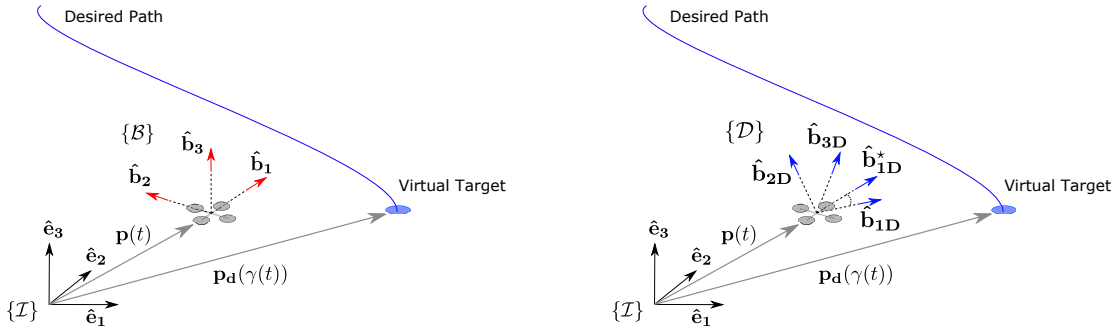
$$(\{\boldsymbol{\omega}_{\boldsymbol{D/I}}\}_D)^{\wedge} = \boldsymbol{R}_{\boldsymbol{c}}^{\top}\dot{\boldsymbol{R}}_{\boldsymbol{c}}\,.$$

Note that, if $\tilde{\boldsymbol{R}} = \mathbb{I}_{\mathbf{3}}$, then the frame $\{\mathcal{B}\}$ coincides with the desired frame $\{\mathcal{D}\}$. Furthermore, consider the following real-valued function on $\mathsf{SO(3)}$

$$\Psi(\tilde{\boldsymbol{R}}) = \frac{1}{2} tr(\mathbb{I}_{\mathbf{3}} - \tilde{\boldsymbol{R}})\,, \tag{5.7}$$

and its time derivative

$$\dot{\Psi}(\tilde{\boldsymbol{R}}) = -\frac{1}{2} tr(\tilde{\boldsymbol{R}}(\tilde{\boldsymbol{\omega}})^{\wedge}).$$



(a) Actual orientation of the vehicle

(b) Desired orientation to approach the path

Figure 5.2: Geometry of the problem.

Finally, let

$$\boldsymbol{e}_{\tilde{R}} = \frac{1}{2}(\tilde{\boldsymbol{R}} - \tilde{\boldsymbol{R}}^{\top})^{\vee}\,, \tag{5.8}$$

where $(\cdot)^{\vee}$ denotes the vee map (see Appendix A.1). Using Equation (A.1) in Appendix A.1 we obtain:

$$\dot{\Psi}(\tilde{\boldsymbol{R}}) = \boldsymbol{e}_{\tilde{R}} \cdot \tilde{\boldsymbol{\omega}}\,. \tag{5.9}$$

Therefore, the dynamics of the VT tracking errors can be summarized by the following system of equations:

$$\begin{cases} \dot{\boldsymbol{e}}_p = \boldsymbol{e}_v \\ m\dot{\boldsymbol{e}}_v = m\ddot{\boldsymbol{p}}_d(\gamma) - T\hat{\mathbf{b}}_{\mathbf{3}} + mg\hat{\mathbf{e}}_{\mathbf{3}} \\ \dot{\Psi}(\tilde{\boldsymbol{R}}) = \boldsymbol{e}_{\tilde{R}} \cdot \tilde{\boldsymbol{\omega}}\,, \end{cases} \tag{5.10}$$

and the VT tracking error vector $\boldsymbol{x}_{\boldsymbol{pf}}(t)$ can be formally defined as

$$\boldsymbol{x}_{\boldsymbol{pf}} = [\boldsymbol{e}_p^{\top}\,,\, \boldsymbol{e}_v^{\top}\,,\, \boldsymbol{e}_{\tilde{R}}^{\top}]^{\top}\,.$$

Notice that in the region $\Psi(\tilde{\boldsymbol{R}}) < 1$, if $\boldsymbol{x}_{\boldsymbol{pf}} = 0$, then the VT tracking position error, the VT tracking velocity error, and the VT tracking attitude error are equal to zero, i.e.

$$\boldsymbol{e}_p = \mathbf{0}\,, \quad \boldsymbol{e}_v = \mathbf{0}\,, \quad \Psi(\tilde{\boldsymbol{R}}) = 0\,.$$

Using the above notation, we now define the VT tracking problem for a multirotor UAS.

Problem 10 *Consider a multirotor UAS and a virtual target $\boldsymbol{p}_d(\gamma(t))$, where $\boldsymbol{p}_d(t)$ is the trajectory generated by solving the optimal motion planning problem, Problem 2, and $\gamma(t)$ is the virtual time. Assume that*

the dynamics of the virtual time satisfy the bounds given in Equation (2.16)*, with* $\dot{\gamma}_i^{min}, \dot{\gamma}_i^{max}$ *and* $\ddot{\gamma}_i^{max}$ *given by Equation* (2.19)*. The objective is to design feedback control laws for the total thrust* $T(t)$*, roll rate* $p(t)$*, pitch rate* $q(t)$*, and yaw rate* $r(t)$ *such that the generalized VT tracking error vector* $\boldsymbol{x_{pf}} = [\boldsymbol{e}_p^\top\,,\, \boldsymbol{e}_v^\top\,,\, \Psi(\tilde{\boldsymbol{R}})]^\top$*, with the dynamics described in* (5.10)*, converges to a neighborhood of the origin.*

▽

Remark 11 *Notice that the VT tracking problem definition above adapts the definition in Section 2.2.2, Problem 3, to the VT tracking problem for multirotor UASs.*

◆

With this setup, we can now design control laws that solve the VT tracking control problem as defined above.

5.2 Virtual target tracking controller

First, let the total thrust command be governed by

$$T_c = ((k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma))^\top\, \hat{\mathbf{b}}_{\mathbf{3}}\,. \tag{5.11}$$

In addition, let the angular rate commands be given by

$$\begin{bmatrix} p_c \\ q_c \\ r_c \end{bmatrix} = \tilde{\boldsymbol{R}}^\top \{\boldsymbol{\omega_{D/I}}\}_D - k_{\tilde{R}}\boldsymbol{e}_{\tilde{R}}\,, \tag{5.12}$$

for some $k_{\tilde{R}} > 0$.

Then, the Lemma below states one of the main results of this chapter:

Lemma 2 *Let the total thrust* $T_c(t)$ *and the angular rate commands* $[p_c(t)\,,\, q_c(t)\,,\, r_c(t)]$ *be governed by* (5.11) *and* (5.12)*. Assume ideal performance for the existing inner-loop autopilot (i.e.* $T(t) = T_c(t)$ *and* $[p(t)\,,\, q(t)\,,\, r(t)] = [p_c(t)\,,\, q_c(t)\,,\, r_c(t)] \quad \forall t \geq 0$*). Then, there exist* $k_p\,, k_v\,, k_{\tilde{R}}$ *such that the error vector*

$$\boldsymbol{x_{pf}} = [\boldsymbol{e}_p^\top\,,\, \boldsymbol{e}_v^\top\,,\, \boldsymbol{e}_{\tilde{R}}^\top]^\top \tag{5.13}$$

converges exponentially to zero *with rate of convergence*

$$\lambda_{pf} < \frac{c_1(1-c^2)}{m}\,, \tag{5.14}$$

in the corresponding domain of attraction

$$\Omega_{pf} \triangleq \left\{ (\boldsymbol{e}_p\,, \tilde{\boldsymbol{R}}) \,|\, \Psi(\tilde{\boldsymbol{R}}) \leq c^2\,, ||\boldsymbol{e}_p|| \leq e_{p\,\max} \right\}, \tag{5.15}$$

for some $c_1, e_{p\,\max} > 0$ *and* $c^2 < 1/2$. *In other words, the error vector* $\boldsymbol{x_{pf}}$ *satisfies*

$$\|\boldsymbol{x_{pf}}(t)\| \leq k_{pf} \|\boldsymbol{x_{pf}}(0)\| \mathrm{e}^{-\lambda_{pf} t}\,,$$

where

$$k_{pf} := \sqrt{\frac{\lambda_{\max}(\boldsymbol{W_2})}{\lambda_{\min}(\boldsymbol{W_1})}}\,, \tag{5.16}$$

and

$$\boldsymbol{W_1} = \begin{bmatrix} \frac{k_p}{2} & -\frac{c_1}{2} & 0 \\ -\frac{c_1}{2} & \frac{m}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{W_2} = \begin{bmatrix} \frac{k_p}{2} & -\frac{c_1}{2} & 0 \\ -\frac{c_1}{2} & \frac{m}{2} & 0 \\ 0 & 0 & \frac{1}{1-c^2} \end{bmatrix}. \tag{5.17}$$

♢

Proof: The proof of Lemma 2 is given in Appendix B.3.2.

Note that we modified the controller used in [145] in two important ways. First, we reparametrized it as VT tracking controller and second we modified it to use angular rates and thrust as control inputs to be tracked by an existing inner-loop autopilot. In turn, this compels us to consider the underlying performance limitations due to the constraints of the inner-loop controller. Thus, as our last step, we consider the case of non perfect tracking performance of the autopilot, which is the main result of this chapter.

Lemma 3 *Let the total thrust* $T_c(t)$ *and the angular rate commands* $[p_c(t)\,, q_c(t)\,, r_c(t)]$ *be governed by* (5.11) *and* (5.12). *Let the inner-loop autopilot satisfy the following performance bounds*

$$|p_c(t) - p(t)| \leq \gamma_p\,, \quad |q_c(t) - q(t)| \leq \gamma_q\,, \quad |r_c(t) - r(t)| \leq \gamma_r\,, \quad |T_c(t) - T(t)| \leq \gamma_T\,,$$

where $T(t)\,, p(t)\,, q(t)$, *and* $r(t)$ *are the actual total thrust and angular rates, and let* $\gamma_\omega = \sqrt{\gamma_p^2 + \gamma_q^2 + \gamma_r^2}$.

Let these performance bounds verify

$$\frac{(c_1/m+1)\gamma_T+\gamma_\omega}{\lambda_{pf}\lambda_{\min}(\boldsymbol{W_2})\delta_\lambda} \leq \min(e_{p\max}\,,\,(1-c^2)c^2)\,, \tag{5.18}$$

where δ_λ *satisfies* $0<\delta_\lambda<1$*, and* $\boldsymbol{W_2}$ *was defined in* (5.17)*. Then, there exist* $k_p\,, k_v\,, k_{\tilde{R}}$ *such that, for any initial state* $\boldsymbol{x_{pf}}(0)\in\Omega_{pf}$*, where* Ω_{pf} *was defined in* (5.15)*, the VT tracking error is uniformly ultimately bounded. More precisely, for any initial state* $\boldsymbol{x_{pf}}(0)\in\Omega_{pf}$*, there exists a time* $T_b\geq 0$ *such that the following bounds are satisfied*

$$\|\boldsymbol{x_{pf}}(t)\| \leq k_{pf}\|\boldsymbol{x_{pf}}(0)\|\mathrm{e}^{-\lambda_{pf}(1-\delta_\lambda)t}\,, \qquad \textit{for all } 0\leq t<T_b\,, \tag{5.19}$$

$$\|\boldsymbol{x_{pf}}(t)\| \leq \rho\,, \qquad \textit{for all } t\geq T_b\,, \tag{5.20}$$

where k_{pf} *was defined in* (5.16)*, and*

$$\rho := \sqrt{\frac{\lambda_{\max}(\boldsymbol{W_2})}{\lambda_{\min}(\boldsymbol{W_1})}}\left(\frac{(c_1/m+1)\gamma_T+\gamma_\omega}{\lambda_{pf}\lambda_{\min}(\boldsymbol{W_2})\delta_\lambda}\right)\,. \tag{5.21}$$

Proof: The proof of Lemma 3 is given in Appendix B.3.4.

5.3 Simulation example

In this section we present simulation results for a scenario in which a multirotor UAS is required to follow a predefined trajectory while pointing at a ground vehicle moving at constant speed. We implement the six degree-of-freedom model given by Equation (5.1). The initial position and velocity of the ground vehicle are $\boldsymbol{p}_t=[0\,,\,0\,,\,0]^\top$ and $\boldsymbol{v}_t=[0.4\,,\,0.4\,,\,0]^\top$, respectively. The UAS is initially positioned at $\boldsymbol{p}=[1\,,\,0\,,\,0.5]^\top$, with orientation

$$\mathbf{R}=\begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{bmatrix}\,,$$

i.e., the $\hat{\mathbf{b}}_{\mathbf{1}}$ axis is pointing in the negative x-axis direction, while the $\hat{\mathbf{b}}_{\mathbf{3}}$ axis is pointing upwards. The desired orientation for the multirotor UAS's $\hat{\mathbf{b}}_{\mathbf{1}}$-axis is computed according to Remark 10 as follows:

$$\hat{\mathbf{b}}^\star_{\mathbf{1D}}=\frac{\boldsymbol{p}_t-\boldsymbol{p}}{||\boldsymbol{p}_t-\boldsymbol{p}||}\,,\quad \hat{\mathbf{b}}_{\mathbf{2D}}=\frac{\hat{\mathbf{b}}_{\mathbf{3D}}\times\hat{\mathbf{b}}^\star_{\mathbf{1D}}}{||\hat{\mathbf{b}}_{\mathbf{3D}}\times\hat{\mathbf{b}}^\star_{\mathbf{1D}}||}\,,\quad \hat{\mathbf{b}}_{\mathbf{1D}}=\hat{\mathbf{b}}_{\mathbf{2D}}\times\hat{\mathbf{b}}_{\mathbf{3D}}\,.$$

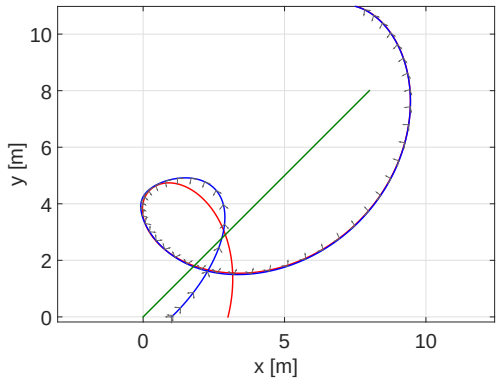
Figure 5.3 includes the simulation results. In particular, Figures 5.3a and 5.3b illustrate the 2D and 3D plots of the actual and desired trajectories of the UAS, as well as the ground vehicle's trajectory, while Figures 5.3c and 5.3d show the speed and acceleration profiles, respectively. The execution of the mission at three different time steps is depicted in Figure 5.4, which illustrates the desired (red) and actual (blue) trajectories of the UAS. The multirotor's body frame $\{\mathcal{B}\}$ (positioned on the vehicle's center of gravity) and the desired frame $\{\mathcal{D}\}$ (positioned on the desired path) are also depicted in the same figure, with the $\hat{\mathbf{b}}_{\mathbf{1}}$ and $\hat{\mathbf{b}}_{\mathbf{1D}}$ axes always pointing towards the ground vehicle. Figure 5.5 highlights the performance of the VT tracking algorithm. In particular, it is shown that the VT tracking errors converge to a neighborhood of zero at $t \approx 8$ s. The VT tracking control efforts, namely angular rates and total thrust, are also depicted in the same figure.

For the sake of providing a more realistic simulation scenario, we repeat the above experiment, and implement a simple inner-loop autopilot for angular rate and total thrust commands tracking. Additionally, measurement noise and transmission delays have been added. The 2D and 3D plots of the actual and desired trajectories are illustrated in Figures 5.6a and 5.6b, respectively, while Figures 5.6c and 5.6d depict the actual and desired speed and acceleration profiles. Figure 5.7 depicts the performance of the mission at three different times. The performance of the VT tracking controller is illustrated in Figure 5.8. Figure 5.8a shows the VT tracking error, converging to a neighborhood of zero slower with respect to the previous case (see Figure 5.5a). Finally, the VT tracking commands and actual angular rates and total thrust of the vehicle are shown in Figures 5.8b–5.8e.

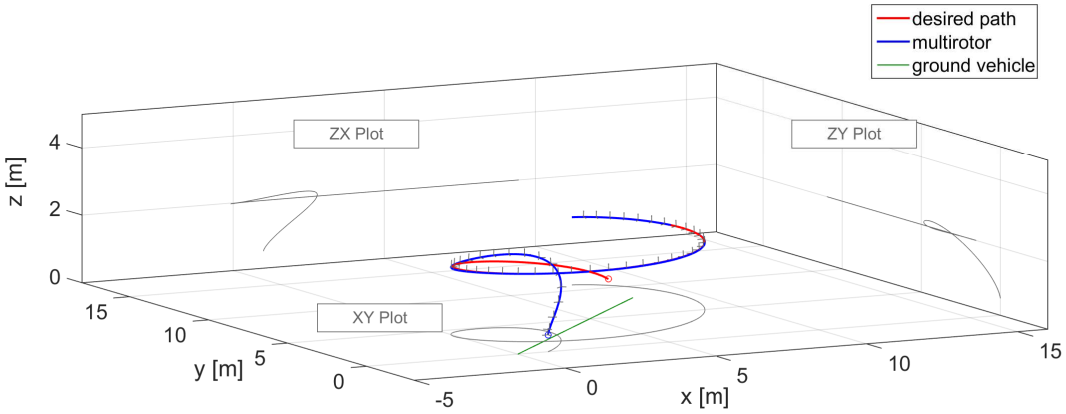
Finally, Figure 5.9 presents an additional simulation scenario in which the multirotor UAS is asked to perform more aggressive maneuvers by "drawing" the phrase *path following* in the air. The overall mission is executed in 260 s. However, for the sake of clarity, Figures 5.9c, 5.9d, and 5.9e show the VT tracking error in the first 20 s of the mission. The desired path is produced by connecting four Bernstein polynomials approximants using a total of 400 points.

In the simulation scenarios described in this section, the control gains have been selected as follows:

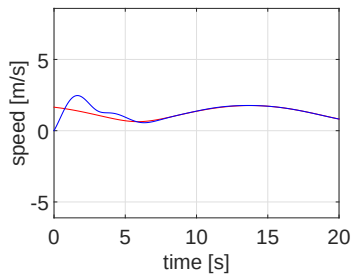
$$k_p = 3\,, \quad k_v = 3\,, \quad k_{\tilde{R}} = 5\,.$$



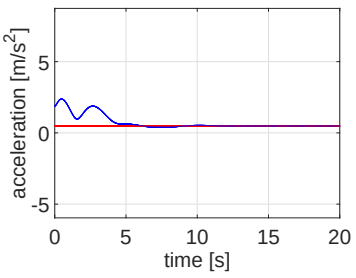
(a) 2D Path



(b) 3D Path



(c) Speed profile



(d) Acceleration profile

Figure 5.3: 3D VT tracking: simulation scenario.

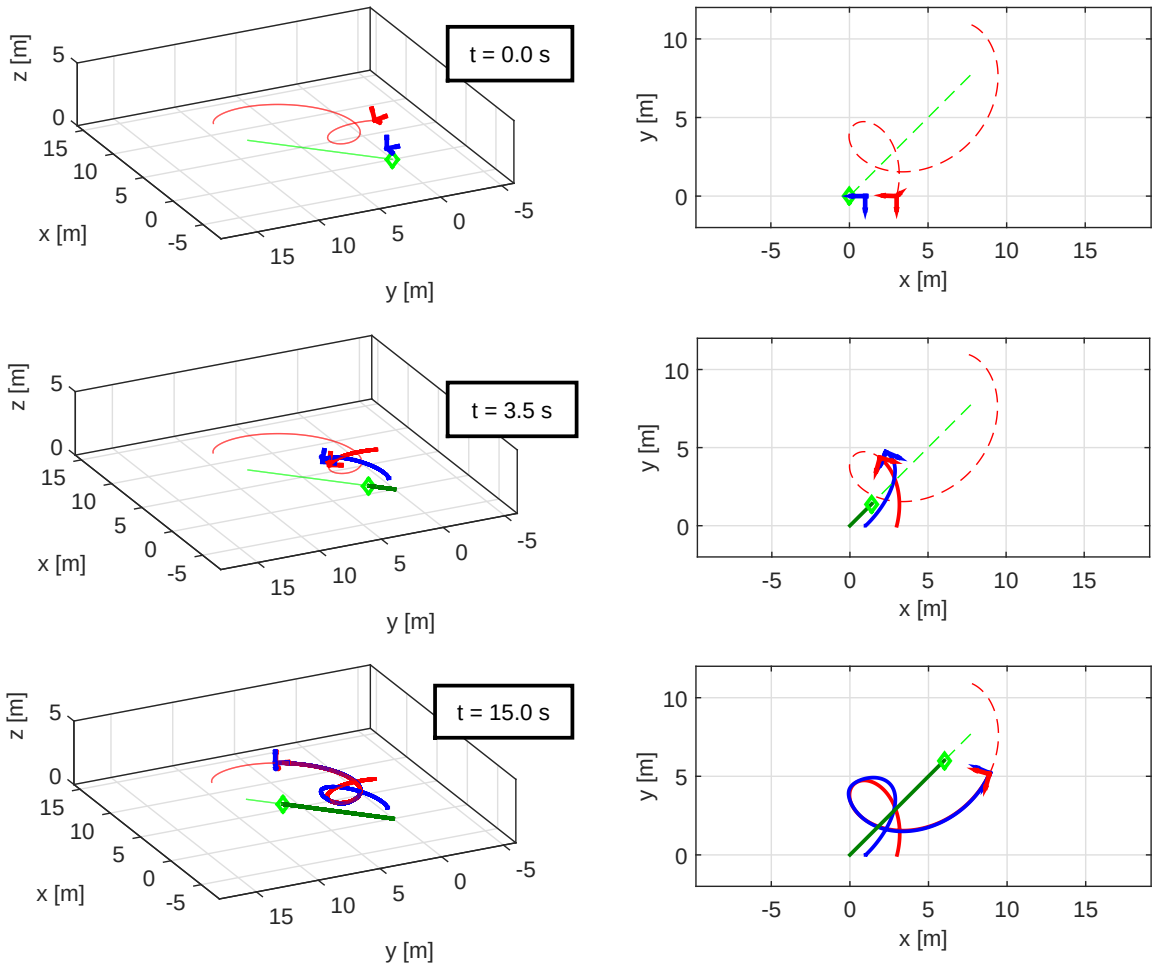
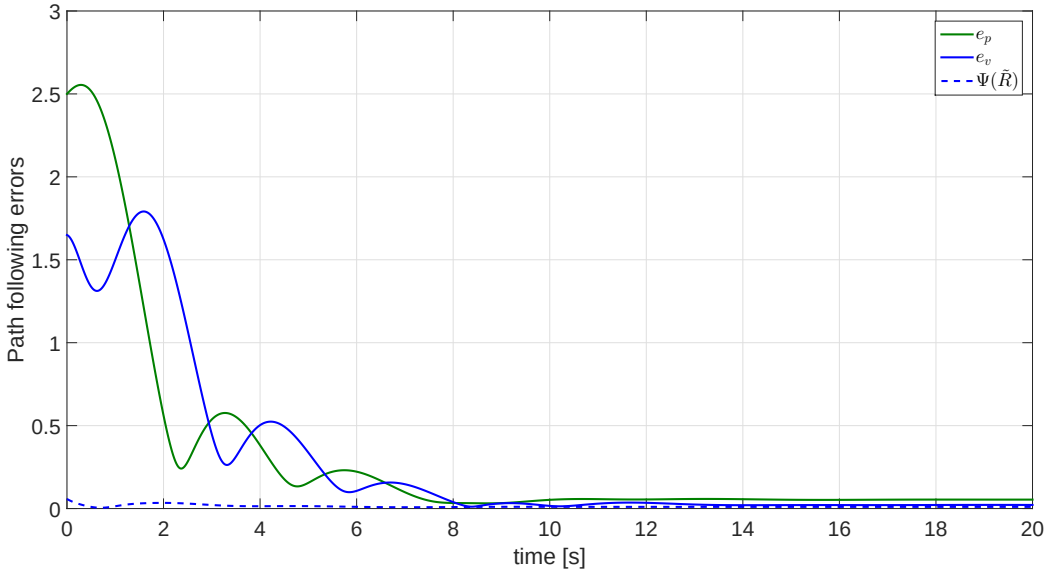
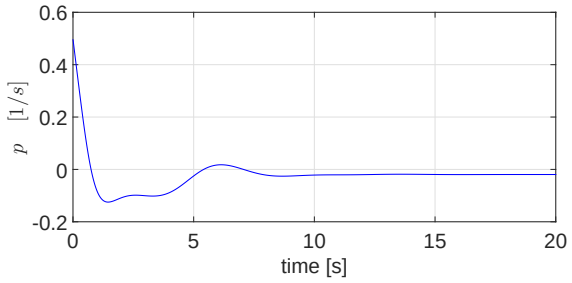


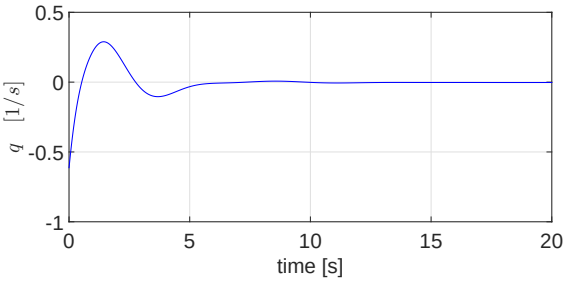
Figure 5.4: 3D VT tracking of a multirotor UAS.



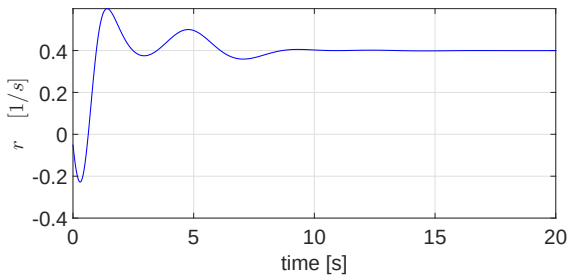
(a) VT tracking error



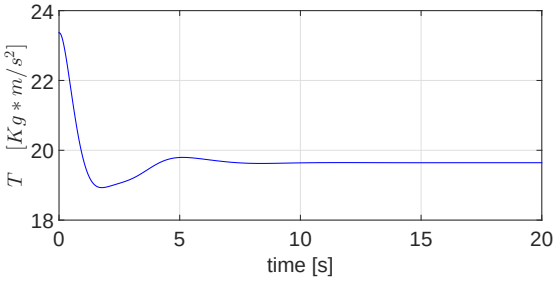
(b) Angular rate, $p(t)$.



(c) Angular rate, $q(t)$.

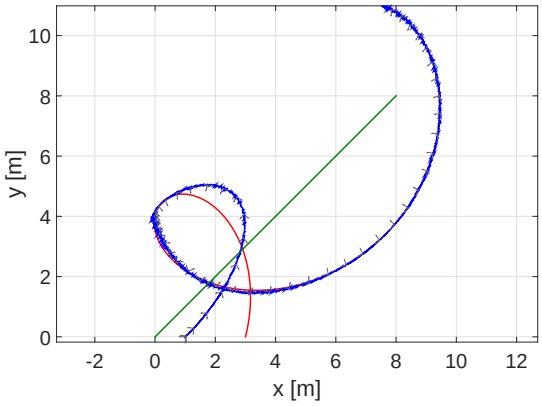


(d) Angular rate, $r(t)$.

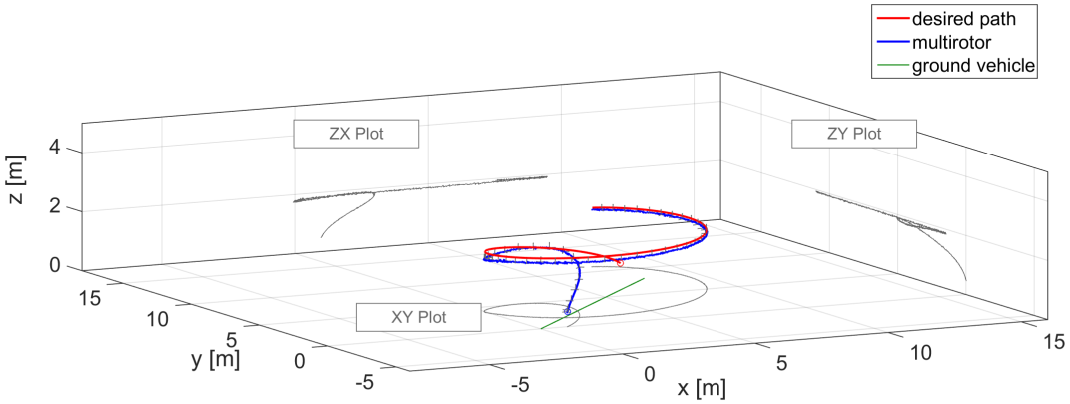


(e) Total thrust, $T(t)$.

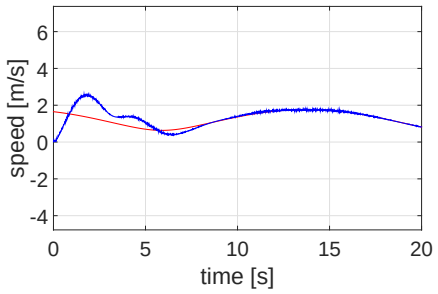
Figure 5.5: VT tracking errors commands.



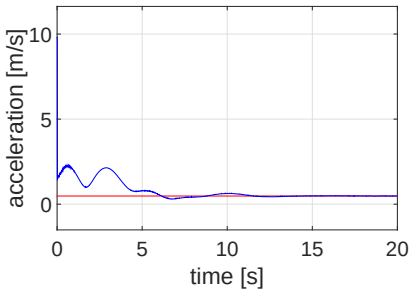
(a) 2D Path



(b) 3D Path



(c) Speed profile



(d) Acceleration profile

Figure 5.6: 3D VT tracking: simulation scenario.

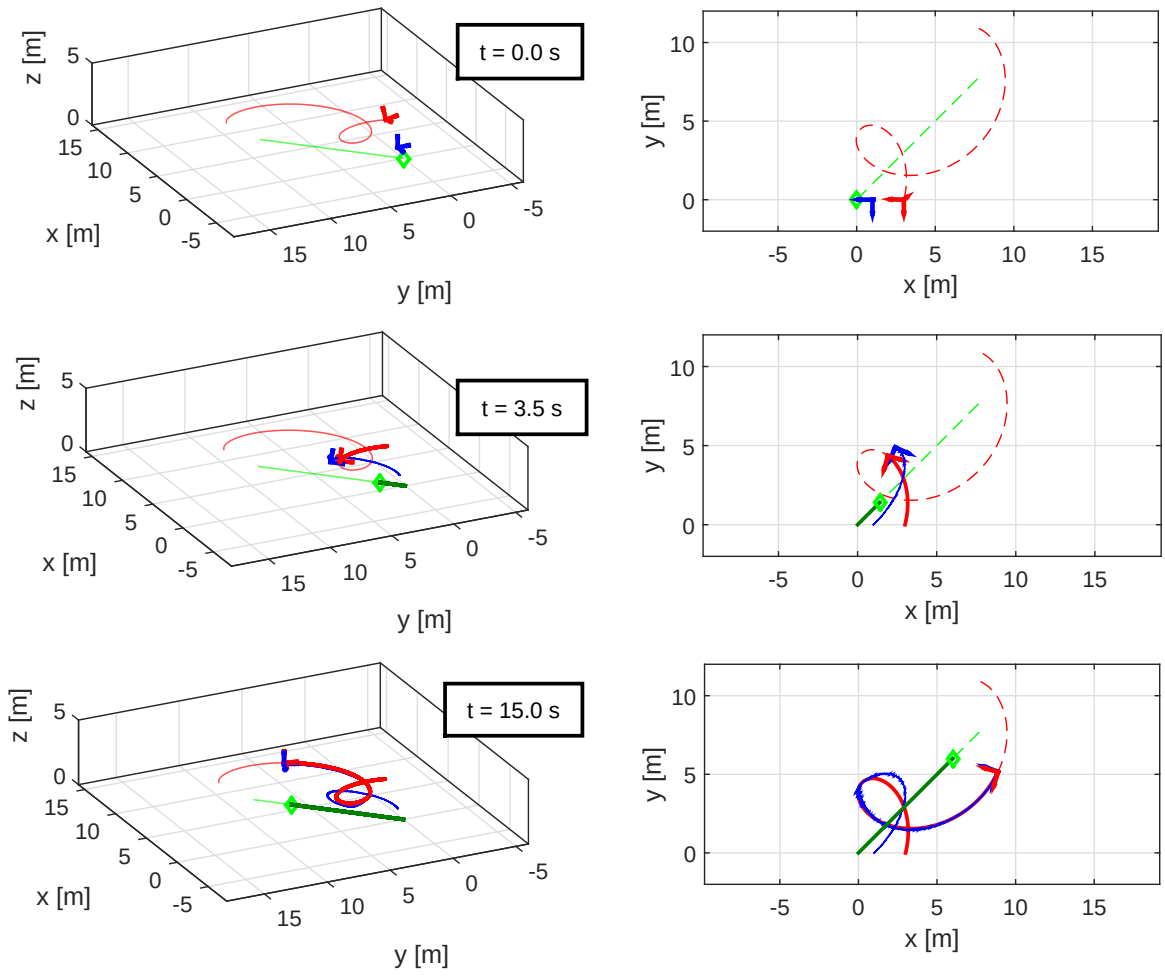
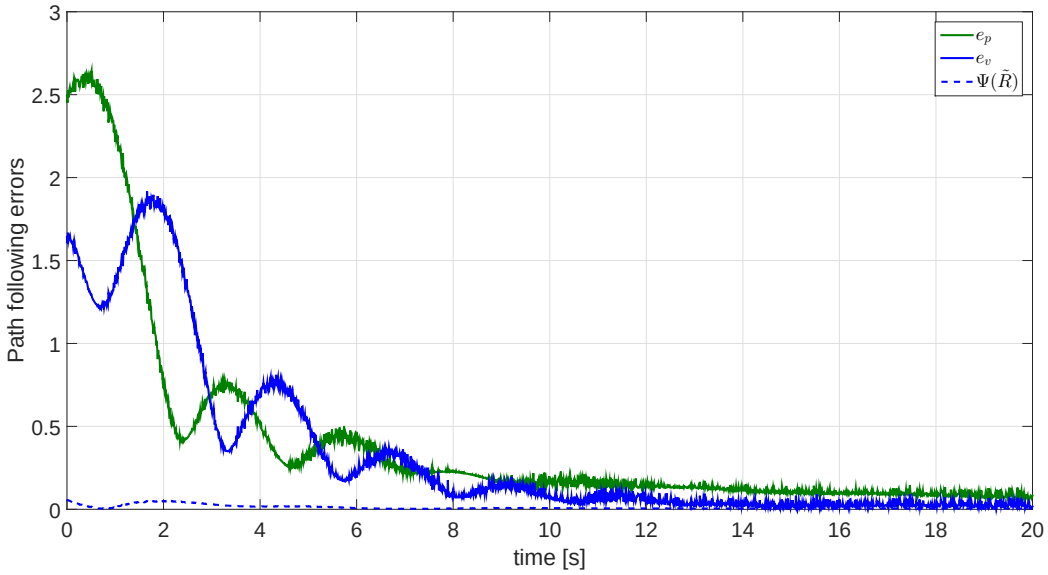
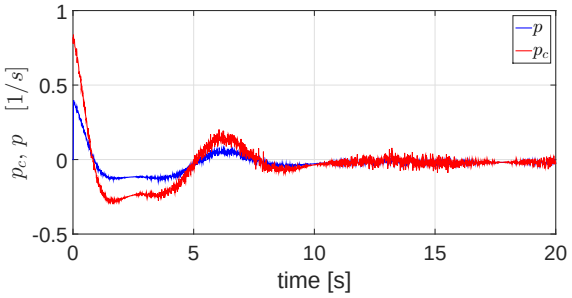


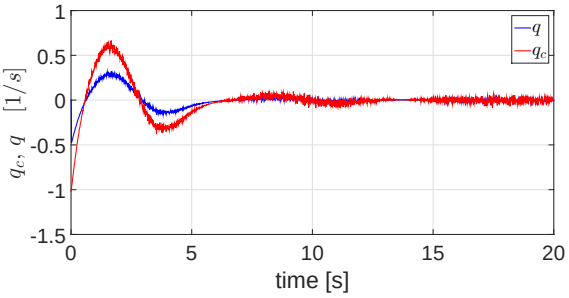
Figure 5.7: 3D VT tracking of a multirotor UAS.



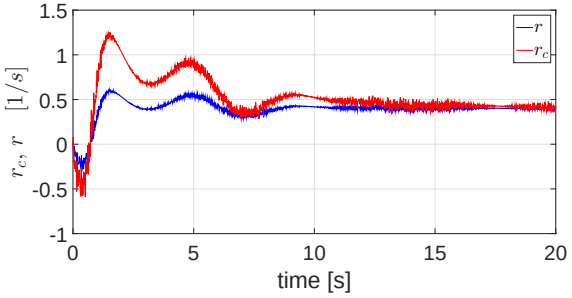
(a) VT tracking error



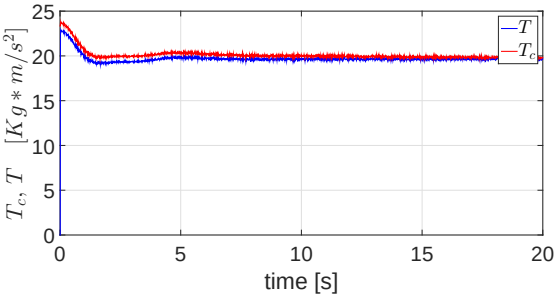
(b) Angular rate, $p(t)$



(c) Angular rate, $q(t)$

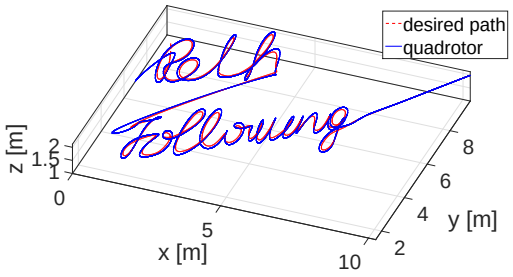


(d) Angular rate, $r(t)$

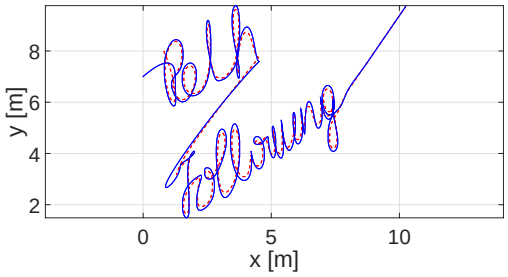


(e) Total thrust, $T(t)$

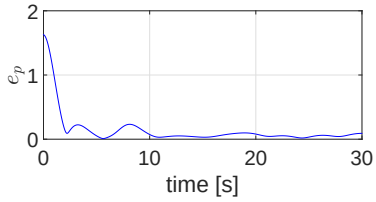
Figure 5.8: VT tracking commands.



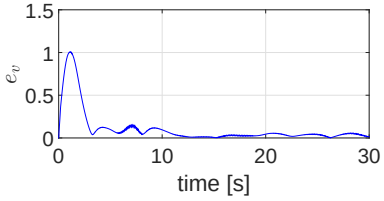
(a) 3D Path



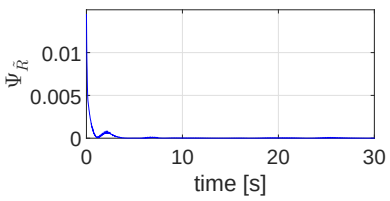
(b) 2D Path



(c) VT tracking error, $e_p(t)$



(d) VT tracking error, $e_v(t)$



(e) VT tracking error, $\Psi_{\tilde{R}}(t)$

Figure 5.9: Drone drawing a Bézier curve.

Chapter 6

Coordination of multiple autonomous vehicles

This chapter addresses the problem of coordinating a group of UxSs. As described in previous chapters, the cooperative missions considered require that each vehicle follow a feasible collision-free path, and that all vehicles arrive at their respective final destinations at the same time, or at different times but meeting a desired inter-vehicle schedule. In this chapter we assume that the UxSs under consideration are equipped with VT tracking controllers, which enable the vehicles to track a set of feasible virtual targets. Then, the coordination problem is solved by controlling the rate of progression of the virtual targets along given desired trajectories. We tackle the problem of coordination control with time-varying communications networks using a Lyapunov-based approach and derive performance bounds as a function of the quality of service of the communications network. Furthermore, we study the degradation in performance that arises when the VT tracking controller exhibits bounded tracking errors.

6.1 Coordination states and maps

Recall from Section 2.2.2 that the desired position assigned to the ith vehicle at time t is given by $\boldsymbol{p}_{d,i}(\gamma_i(t))$, where $\boldsymbol{p}_{d,i}(t)$ is the trajectory produced by the motion planning algorithm, and the parameter $\gamma_i(t)$ is the virtual time. We notice that, if $\dot{\gamma}_i(t) = 1$, the desired speed profile to which the vehicle is required to fly is equal to the speed profile given by the trajectory generation algorithm (i.e. $\dot{\gamma}_i(t) = 1$ implies that the mission is executed at the desired pace). On the other hand, $\dot{\gamma}_i(t) > 1$ ($\dot{\gamma}_i(t) < 1$) implies a faster (slower) execution of the mission. The virtual time and its first time derivative play a crucial role in the coordination problem. In fact, given that the paths produced by the trajectory generation algorithm satisfy the simultaneous time-of-arrival requirement (see Equation (2.7)), we say that if

$$\gamma_i(t) - \gamma_j(t) = 0\,, \quad \forall i, j \in \{1, \ldots, n\}\,, \quad i \neq j\,, \tag{6.1}$$

then, at time t, all the vehicles are coordinated. Moreover, as already discussed in Section 2.2.2, if

$$\dot{\gamma}_i - 1 = 0\,,\ \forall i \in \{1, \ldots, n\}\,, \tag{6.2}$$

then the desired speed at which the vehicles are required to converge is equal to the desired speed profile established at the motion planning level. Thus, Equations (6.1) and (6.2) capture the objective of coordination, and require that a control law for $\ddot{\gamma}_i$ must be formulated to ensure that the objective will be achieved.

Remark 12 *We notice that if the desired trajectories $\boldsymbol{p}_{d,i}(t)$ produced by the motion planning algorithm satisfy the temporal separation requirement, i.e.*

$$\|\boldsymbol{p}_{d,i}(t) - \boldsymbol{p}_{d,j}(t)\| \ \geq\ E_d\,, \quad \forall t \in [0, t_f]\,, \quad i, j = 1, \ldots, n, \quad i \neq j$$

(see Section 2.2.1), and if the vehicles are equipped with VT tracking algorithms that solve Problem 3 in Section 2.2.2, then satisfaction of Equations (6.1) *and* (6.2) *ensures inter-vehicle collision avoidance.*

◆

Now, let $\boldsymbol{\gamma}(t) = [\dot{\gamma}_1(t), \ldots, \dot{\gamma}_n(t)]^\top$, and define the coordination error vectors as

$$\boldsymbol{\zeta_1}(t) = \boldsymbol{Q}\boldsymbol{\gamma}(t) \qquad \in \mathbb{R}^{n-1}\,, \tag{6.3}$$

$$\boldsymbol{\zeta_2}(t) = \dot{\boldsymbol{\gamma}}(t) - \mathbf{1} \qquad \in \mathbb{R}^{n}\,. \tag{6.4}$$

Recall from Section 2.2.2 that the $n-1 \times n$ matrix $\boldsymbol{Q}$ satisfies $\boldsymbol{Q}\mathbf{1} = \mathbf{0}$; it follows that if $\boldsymbol{\zeta_1}(t) = \mathbf{0}$, then $\gamma_i - \gamma_j = 0$, $\forall i, j \in \{1, \ldots, n\}$. Furthermore, convergence of $\boldsymbol{\zeta_2}(t)$ to zero implies that the individual parameterizing variables $\gamma_i(t)$ evolve at the desired rate 1.

Using the above notation, the coordination problem in Section 2.2.2, Problem 4, can be restated as follows:

Problem 11 *Consider a set of n UxSs. Assume that the ith UxS, with $i \in \{1, \ldots, n\}$, is equipped with (i) a motion planning algorithm that solves Problem 2, Section 2.2.1, and (ii) a VT tracking controller that solves Problem 3, Section 2.2.2. Then, the objective of coordination is to design feedback control laws for $\ddot{\gamma}_i(t)$ for all vehicles such that the coordination vector defined as*

$$\boldsymbol{x_{cd}} = [\boldsymbol{\zeta_1}^\top\,,\, \boldsymbol{\zeta_2}^\top]^\top\,,$$

converges to a neighborhood of zero, and such that the dynamics of the virtual times satisfy Equation (2.16).

∇

6.2 Coordination control law

To solve the coordination problem, we let the evolution of $\gamma_i(t)$ be driven by the following control law:

$$\ddot{\gamma}_i = -b(\dot{\gamma}_i - 1) - a \sum_{j \in \mathcal{N}_i} (\gamma_i - \gamma_j) - \bar{\alpha}_i(\boldsymbol{x}_{pf,i}) \,,$$
$$\gamma_i(0) = 0 \,, \quad \dot{\gamma}_i(0) = 1 \,,$$

where a and b are positive coordination control gains, while $\bar{\alpha}_i(\boldsymbol{e}_{p,i})$ is defined as

$$\bar{\alpha}_i(\boldsymbol{e}_{p,i}) = \frac{\dot{\boldsymbol{p}}_{d,i}(\gamma_i)^\top \boldsymbol{e}_{p,i}}{\|\dot{\boldsymbol{p}}_{d,i}(\gamma_i)\| + \delta} \,,$$

with δ being a positive design parameter. The dynamics of $\boldsymbol{\gamma}(t)$ can be written in compact form as

$$\ddot{\boldsymbol{\gamma}} = -b\boldsymbol{\zeta_2} - a\boldsymbol{L}\boldsymbol{\gamma} - \bar{\boldsymbol{\alpha}}(\boldsymbol{e}_p) \,, \quad \boldsymbol{\gamma}(0) = \boldsymbol{0_n}, \; \dot{\boldsymbol{\gamma}}(0) = \boldsymbol{1} \,, \tag{6.5}$$

where

$$\boldsymbol{e}_p = [\boldsymbol{e}_{p,1}^\top, \ldots, \boldsymbol{e}_{p,n}^\top]^\top \quad \in \mathbb{R}^{9n} \,,$$
$$\bar{\boldsymbol{\alpha}}(\boldsymbol{e}_p) = [\bar{\alpha}_1(\boldsymbol{e}_{p,1}), \ldots, \bar{\alpha}_n(\boldsymbol{e}_{p,n})]^\top \quad \in \mathbb{R}^n \,.$$

The following result is obtained:

Theorem 5 *Consider a set of n UxSs. Assume that the ith UxS, with $i \in \{1, \ldots, n\}$, is equipped with (i) a motion planning algorithm that solves Problem 2 (see Section 2.2.1), and (ii) a VT tracking controller that solves Problem 3 (see Section 2.2.2). Assume that the communication network over which the UxSs communicate satisfies Assumption 1 given in Section 2.2.2. Let the coordination error vector $\boldsymbol{x_{cd}}(t)$ at time $t = 0$ and the VT tracking performance bound ρ introduced in Problem 3, Equation 2.20, satisfy*

$$\max\left(||x_{TC}(0)||, \rho\right) \leq \min\left(\frac{1 - \dot{\gamma}^{\min}}{(\kappa_1 + \kappa_2)}, \frac{\dot{\gamma}^{\max} - 1}{(\kappa_1 + \kappa_2)}, \frac{\ddot{\gamma}^{\max}}{(2b\kappa_1 + 2b\kappa_2 + 1)}, \right) , \tag{6.6}$$

for some $\kappa_1, \kappa_2 > 0$, with $\dot{\gamma}^{\min} = \max_{i=1,\ldots,n} \dot{\gamma}_i^{\min}$, $\dot{\gamma}^{\max} = \min_{i=1,\ldots,n} \dot{\gamma}_i^{\max}$, and $\ddot{\gamma}^{\max} = \min_{i=1,\ldots,n} \ddot{\gamma}_i^{\max}$. Finally, let $\ddot{\boldsymbol{\gamma}}$ be governed by the control law given by Equation (6.5)*. Then, there exist control gains a, b,*

and δ such that the coordination error is uniformly bounded. In particular, the coordination error satisfies

$$||\boldsymbol{x_{cd}}(t)|| \leq \kappa_1 ||\boldsymbol{x_{cd}}(0)|| e^{-\lambda_{cd} t} + \kappa_2 \sup_{t \geq 0} (||\boldsymbol{e}_p(t)||)\,, \tag{6.7}$$

with guaranteed rate of convergence

$$\lambda_{cd} \triangleq \frac{a}{b} \frac{n\mu\delta_{\bar{\lambda}}}{T(1+\frac{a}{b}nT)^2}\,, \quad 0 < \delta_{\bar{\lambda}} < 1\,. \tag{6.8}$$

Furthermore, the dynamics of the virtual times $\dot{\gamma}_i(t)$ and $\ddot{\gamma}_i(t)$ satisfy the bounds given by Equation (2.16)*.*

■

Proof: The proof of Theorem 5 is given in Appendix B.4.1.

♠

Remark 13 *Notice that the maximum convergence rate λ_{cd} is obtained when the control gains a and b satisfy*

$$\frac{a}{b} = \frac{1}{nT}\,. \tag{6.9}$$

Substituting (6.9) *in* (6.8)*, one obtains*

$$\max_{a,b>0}(\lambda_{cd}) = \frac{\mu\delta_{\bar{\lambda}}}{4T^2}\,,$$

i.e. the rate of convergence depends on the quality of the network only.

♦

Corollary 1 *Suppose that each UxSs is equipped with an ideal VT tracking controller such that the VT tracking error converges exponentially fast to zero (e.g. the VT tracking algorithm designed in Chapter 5 is exponentially stable under ideal performance of the inner-loop autopilot, see Lemma 2). Then, the coordination error converges exponentially fast to zero as follows:*

$$||\boldsymbol{x_{cd}}(t)|| \leq \bar{\kappa}_1 ||\boldsymbol{x_{cd}}(0)|| e^{-\lambda_{cd} t} + \bar{\kappa}_2 ||\boldsymbol{e}_p(0)|| e^{-\frac{\lambda_{pf}+\lambda_{cd}}{2} t}\,, \tag{6.10}$$

for some $\bar{\kappa}_1, \bar{\kappa}_2 > 0$.

■

Proof: The proof of Corollary 1 is given in Appendix B.4.2.

♠

Remark 14 *Notice that the coordination control law introduced in* (6.5) *depends also on the VT tracking error. By virtue of the VT tracking dependent term (i.e.* $\bar{\boldsymbol{\alpha}}(\boldsymbol{e}_p)$*), if for example one vehicle is away from the desired position (*$||\boldsymbol{e}_p|| \neq 0$ *), its assigned virtual target speeds up or slows down in order to reduce the VT tracking error; then, as a direct consequence, also the other vehicles involved in the cooperative mission adjust their speed to maintain coordination. This point will become clear in Section 6.3, where the simulation results are presented.*

◆

Remark 15 *The formulation of the coordination problem described above assumes only relative temporal constraints in the execution of a given mission. Absolute temporal constraints, such as specifications in the desired final time of the mission, are not considered. Nevertheless, such constraints can be easily incorporated in the problem formulation, and enforced by judiciously modifying the coordination control laws presented in this chapter.*

◆

6.3 Simulation results

In this section we present simulation results for the scenario introduced in Figure 6.1, where eight quadrotor UASs, initially positioned along the perimeter of a $40m \times 40m$ square area, have to exchange their positions and arrive at their final destinations at the same time. Before the mission starts, a set of trajectories are generated which ensure temporal deconfliction ($E = 1m$) of the UASs, i.e.

$$\min_{\substack{j,k=1,\dots,8 \\ j \neq k}} \|\boldsymbol{p}_{d,j}(t) - \boldsymbol{p}_{d,k}(t)\|^2 \;\geq\; E^2\,, \qquad\qquad \text{for all } t \in [0, t^f]\,.$$

Figure 6.1 depicts the 2D projection of these trajectories.

In the remainder of this section, we analyze and validate the theoretical findings through three different simulations. In the first simulation we consider the case of ideal all-to-all communication between the vehicles, and assume that the UASs' states coincide with their desired positions, velocities, and orientations for all time, i.e. $||\boldsymbol{e}_p(t)|| = 0\,, \quad \forall t \geq 0$. In the second simulation, a similar experiment is performed, but considering non-ideal communication. In the third simulation, we introduce VT tracking errors by implementing the VT tracking controller introduced in Chapter 5 with a simple inner-loop autopilot for angular rate and total

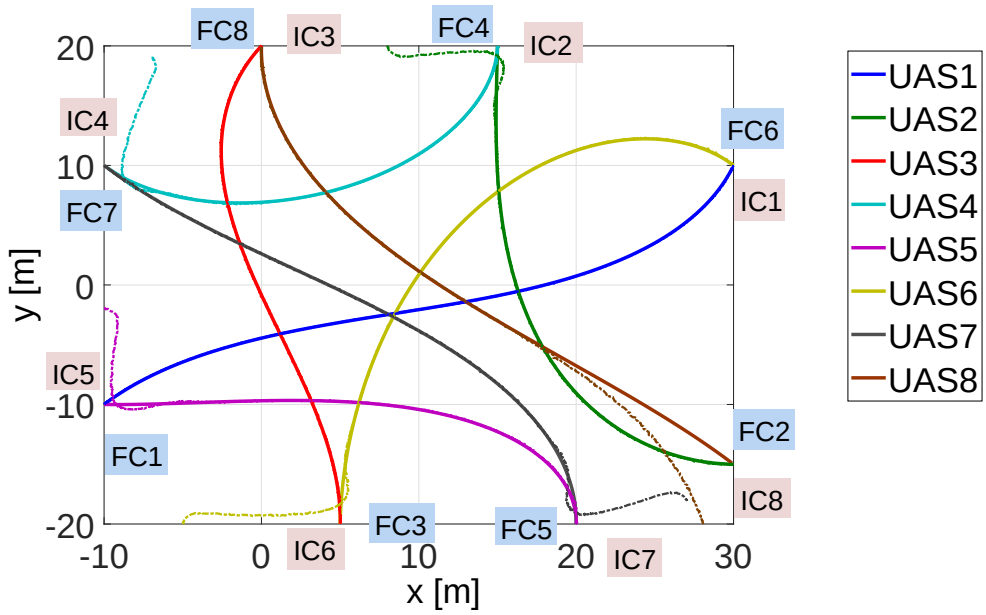


Figure 6.1: Simulation results with eight quadrotor UASs - desired (solid lines) and UASs (dashed lines) trajectories.

thrust commands tracking. In all the experiments, the control gains are chosen to be

$$a = 1.5\,, \qquad b = 3.6\,, \qquad \delta = 3\,.$$

To illustrate the convergence properties of the solution, the virtual times are initialized as follows:

$$\gamma_1(0) = 2\,, \quad \gamma_4(0) = 3\,, \quad \gamma_6(0) = 1\,, \quad \gamma_8(0) = 1.5\,, \quad \gamma_2(0) = \gamma_3(0) = \gamma_5(0) = \gamma_7(0) = 0\,.$$

6.3.1 Ideal communication - ideal virtual target tracking

In this simulation, all the vehicles communicate with each other all the time, i.e. $l_{ii}(t) = 7\,, l_{ij}(t) = -1\,, \quad \forall t \geq 0\,, \quad \forall i,j \in \{1\,,\ldots,8\}\,, i \neq j$, where $l_{ij}(t)'s$ are the entries of the Laplacian matrix $\boldsymbol{L}(t)$. Moroever, we let $||\boldsymbol{e}_p(t)|| = 0\,, \forall t \geq 0$, i.e. the VT tracking algorithm exhibits ideal performance.

At time $t = 0$ the vehicles start navigating the room and follow the predefined trajectories until they reach their final destination, at time $t \approx 8.8s$. In Figure 6.1, the solid lines indicate the trajectories of each UAS, while ICi and FCi indicate, respectively, initial and final position of UASi.

In Figure 6.3 the coordination variables are illustrated. At the beginning of the mission vehicles $1, 4, 6$ and 8 speed up, while vehicles $2, 3, 5$ and 7 slow down (see Figure 6.3b and 6.3c) until, at time $t \approx 2s$, coordination is achieved. Figure 6.3a shows convergence of the virtual times to the same increasing value.

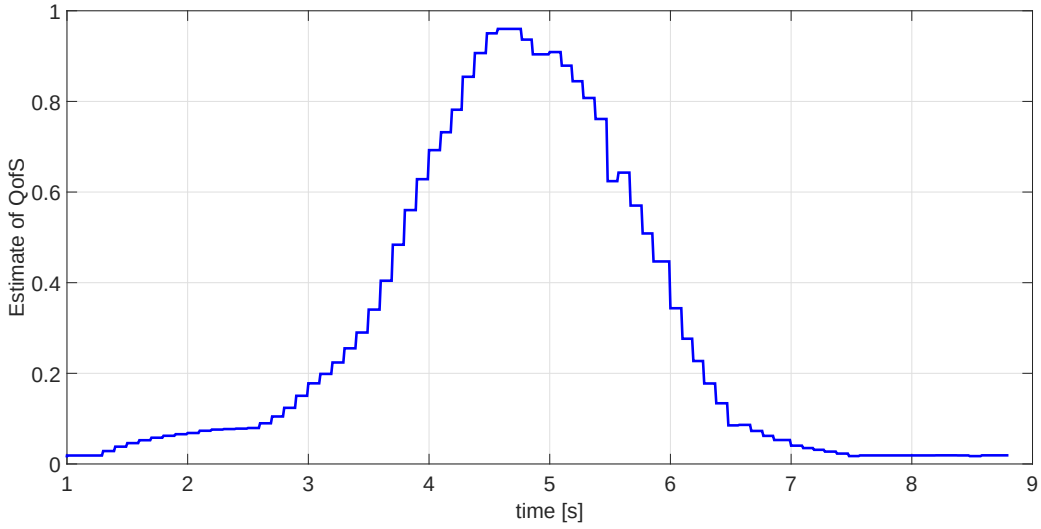


Figure 6.2: Estimate of Quality of Service.

6.3.2 Range-based communication - ideal virtual target tracking

The same experiment is repeated, but in this case we let UASi and UASj communicate with each other at time $t \geq 0$ only if $||\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)|| \leq 20m$.

Figure 6.2 depicts an estimate of the Quality of Service of the network computed as

$$\hat{\mu}(t) = \lambda_{\min}\left(\frac{1}{n}\frac{1}{T}\int_{t-T}^{t} \boldsymbol{Q}\boldsymbol{L}(\tau)\boldsymbol{Q}^{\top} d\tau\right), \quad t \geq T,$$

(see Assumption 1 in Section 2.2.2) with $n = 8$ and $T = 1s$. As can be seen from the figure, the estimate of the Quality of Service is greater at $t \approx 4 - 5s$, when the vehicles are positioned near the center of the area, thus all closer to each other. On the other hand, such a value is smaller at the beginning and the end of the mission, when the vehicles communicate only with few neighbours. Figure 6.4 depicts the performance of the coordination algorithm. It can be noted that the coordination variables converge to the desired values at time $t \approx 4s$ slower than the case with ideal-communication.

6.3.3 Range-based communication - non-ideal virtual target tracking

In this last experiment, to illustrate impact of the non-zero VT tracking error, we implemented the VT tracking controller described in Chapter 5 together with a simple inner loop autopilot to track angular rates and total thrust commands. According to the result obtained in Lemma 3, the VT tracking error is ultimately bounded, and the coordination error satisfies (6.7). The communication topology is the one used

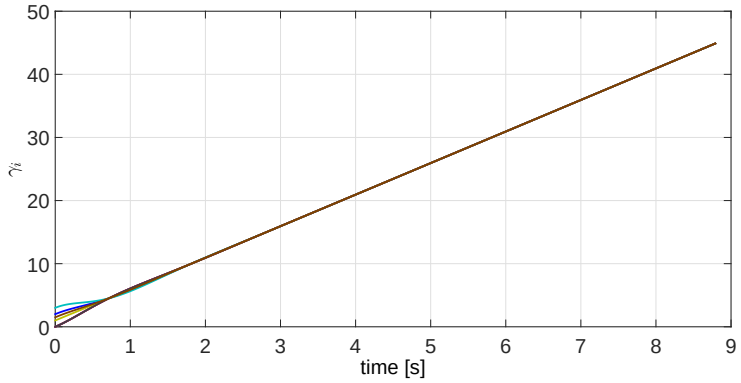
in the previous experiment.

The vehicles start, at $t = 0$, with an initial displacement from the desired positions, and track the desired paths. In Figure 6.1 the dashed lines indicate the actual trajectories of the UASs. Figure 6.5 shows the time history of the coordination variables. Figure 6.6 depicts the time history of the norm of the coordination error state $||\boldsymbol{x_{cd}}(t)||$ (green line), and compares it with the two cases described above (blue and red lines). As expected, the coordination error converges to a neighborhood of the origin, and remains bounded.

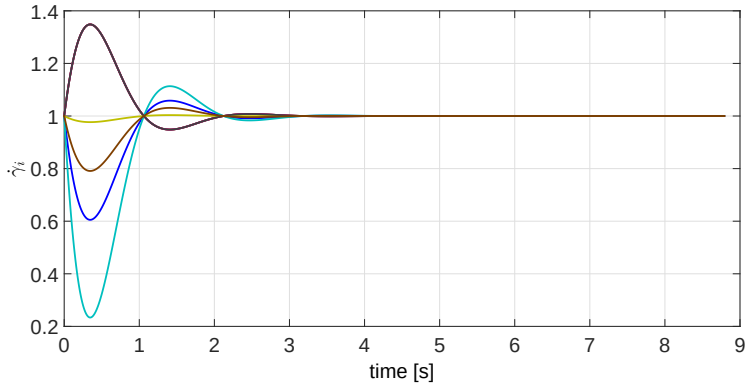
Finally, Figure 6.7 shows the minimum distance between the vehicles throughout the mission, which is

$$\min_{i,j}\{||\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t)||\}\,, \tag{6.11}$$

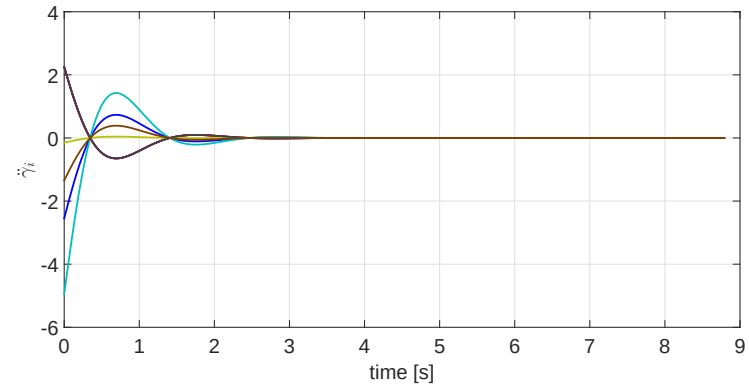
in three different cases: (i) blue line - ideal VT tracking performance; (ii) green line - the VT tracking error is introduced, and the coordination control law given in (6.5) is employed; (iii) red line - the VT tracking error is introduced, and the coordination law employed does not depend on the VT tracking error (i.e. Equation (6.5) without the third term $\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}(\boldsymbol{e}_p)$). While in case (i) temporal separation is guaranteed at the trajectory generation level, when the UASs are away from the desired position, the coordination algorithm must take into account the VT tracking error in order to ensure that the actual UASs' positions are separated. As it was pointed out in Remark 14, the third term in Equation (6.5) enables the UASs to maintain coordination even in the presence of VT tracking errors, which in turn implies that a minimum separation between the vehicles is guaranteed. As it can be seen from Figure 6.5b and 6.5c, since UAS8 is initially displaced by a considerable distance from its desired position, when the mission starts the virtual time associated with UAS8 (i.e. γ_8) decelerates significantly ($\dot{\gamma}_8 < 1$ and $\ddot{\gamma}_8 < 0$) by virtue of $\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}(\boldsymbol{e}_p)$, to allow the vehicle to approach the desired point faster. As a consequence, also γ_1 decelerates to coordinate with γ_8, thus allowing the actual vehicles to synchronize with each other along the paths and maintain a desired separation. In absence of the term $\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}(\boldsymbol{e}_p)$, the virtual times associated with the vehicles would keep coordinating with each other without accounting for the actual position of the UASs, thus leading to potential collisions (red line in Figure 6.7).



(a) Virtual Time.

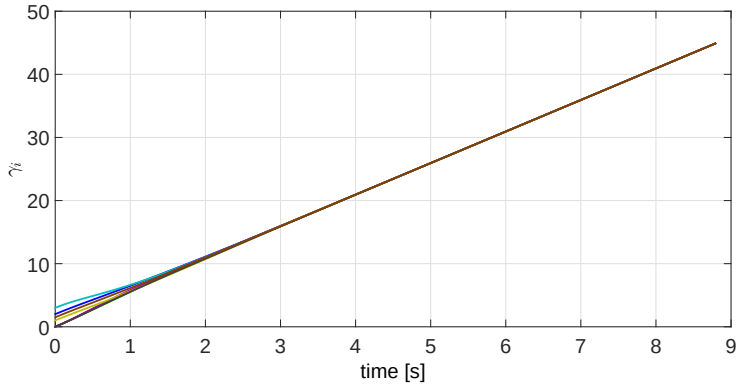


(b) Derivative of Virtual Time.

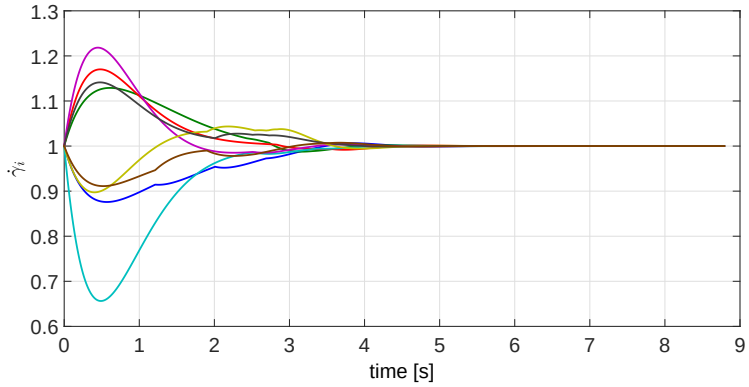


(c) Control input.

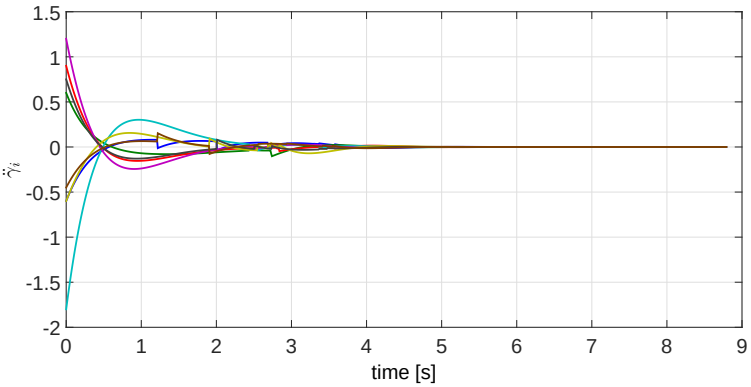
Figure 6.3: Coordination in the case of ideal communication and ideal VT tracking performance.



(a) Virtual Time.

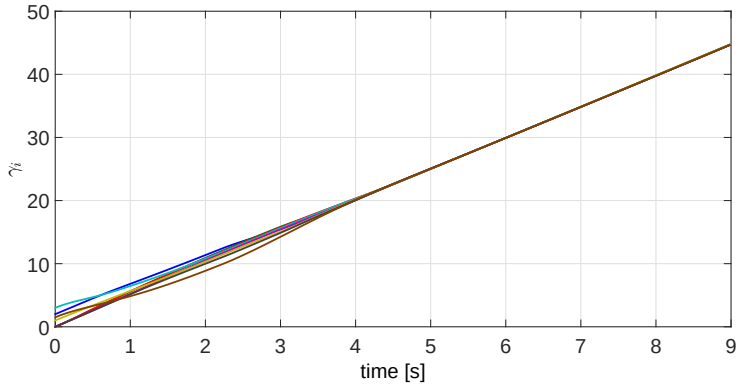


(b) Derivative of Virtual Time.

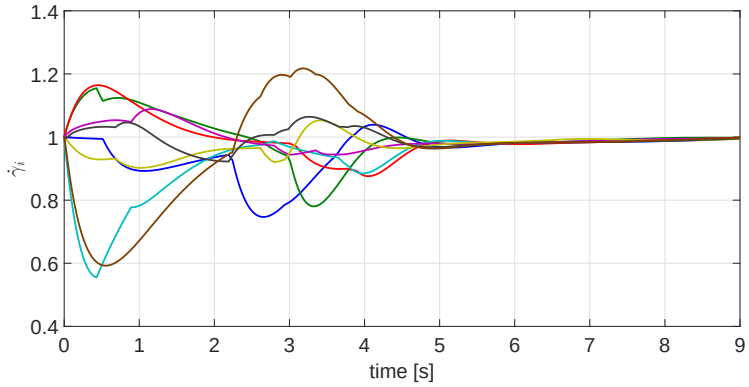


(c) Control input.

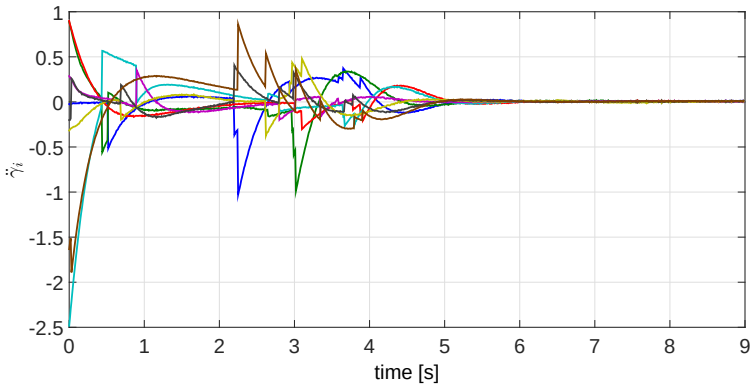
Figure 6.4: Coordination in the case of range-based communication and ideal VT tracking performance.



(a) Virtual Time.



(b) Derivative of Virtual Time.



(c) Control input.

Figure 6.5: Coordination in the case of range-based communication and non-ideal VT tracking performance.

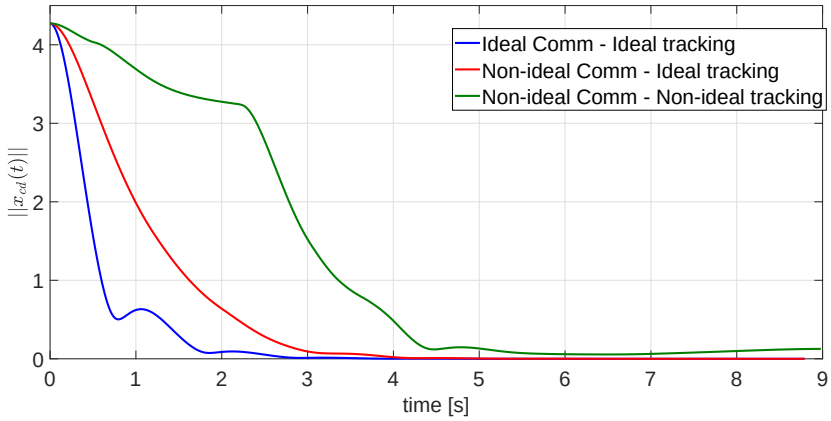


Figure 6.6: Coordination error vector.

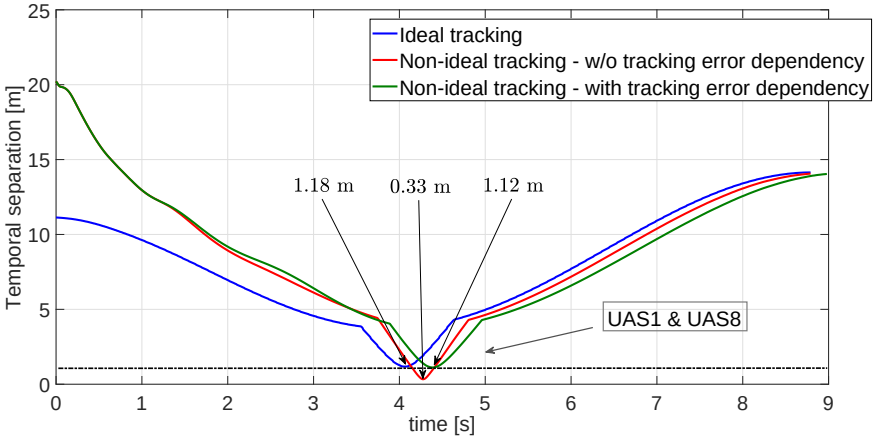


Figure 6.7: Temporal separation between vehicles.

Chapter 7

Flight test results

This chapter presents the results of flight tests with two quadrotor UASs aimed at verifying experimentally the stability and convergence properties associated with the coordination controller presented in Chapter 6. We consider two operational scenarios in which the quadrotors are required to execute simple cooperative tasks. Namely, phase on orbit coordination and spatial coordination along one axis. The reader is referred to [148], where videos of additional experiments can be found. In what follows, we first describe the system architecture and the indoor facility used to conduct the experiments, after which we discuss the flight test results in detail.

7.1 System architecture and indoor facility

The flight tests presented in this chapter were performed at the Center for Autonomous Vehicle Research (CAVR), Naval Postgraduate School, Monterey, CA [149] (see Figure 7.1). The facility is equipped with eight VICON T-160 cameras [150] connected into one network to provide precise synthetic geopositioning with resolution in the order of 1mm in the volume of $30 \times 30 \times 20$ft[1]. The motion capture system data are transmitted to MATLAB\Simulink, running on a Linux OS (ground station). The update rate of the position, velocity, and attitude signals from VICON is available at the frequency of up to 200Hz. The quadrotor UASs employed in these flight tests are the Parrot AR.Drones [151] depicted in Figure 7.2. This platform, commercially available to the general public, is equipped with a sonar height sensor, an Inertial Measurement Unit (IMU) capable of collecting linear acceleration, angular velocity and orientation of the UAS, two cameras (one pointing forward, and the other one facing downwards), and an on-board computer running proprietary software (see Figure 7.2a). The software includes an inner-loop stabilizer which uses the sonar sensor and IMU data in order to track *roll*, *pitch*, *yaw rate*, and *vertical speed* commands. Therefore, the VT tracking problem discussed in Chapter 5 needs to be reformulated for this particular platform, and a VT tracking algorithm needs to be derived. The VT tracking controller employed in these flight tests

[1]The facility includes sixteen cameras at the time of this writing, providing geopositioning data in the volume of $40 \times 40 \times 30$ft.

uses the aforementioned commands as control inputs. A thorough description of the VT tracking algorithm is given in [138]. The VT tracking controller and the coordination algorithm are implemented on the host machine (i.e. the MATLAB\Simulink code implemented on the ground station). The host machine sends the VT tracking commands to the AR.Drones via a wireless ad-hoc connection with an update rate of up to 50Hz. The coordination variables are exchanged among the UASs at a data transfer rate of 100Hz (imposed via Simulink). A schematic of the indoor facility and the system architecture is illustrated in Figure 7.1a.

7.2 Flight test results

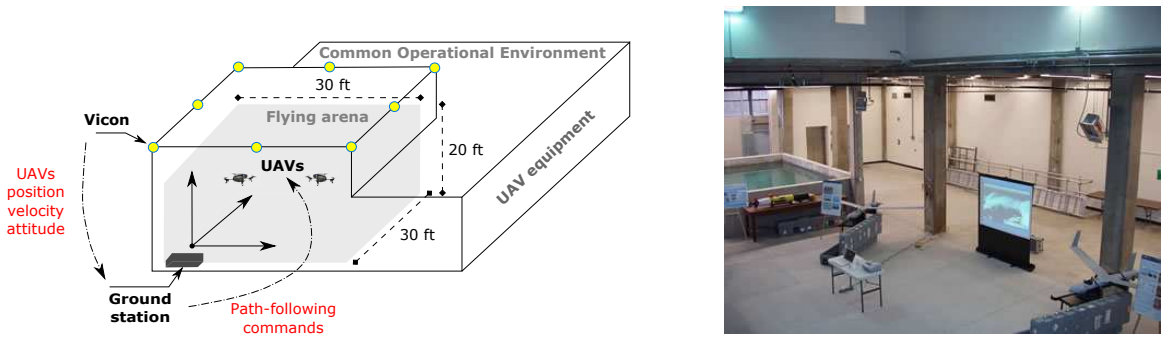
This section describes two flight test scenarios that validate the efficacy of the coordination algorithm described in Chapter 6. The control law introduced in Equation (6.5) is implemented, with control gains selected as follows:

$$a = 3\,, \quad b = 5\,, \quad \delta = 5\,.$$

In the first scenario, i.e. *phase on orbit coordination*, two AR.Drones are required to fly along a circular planar path while maintaining a desired time-varying phase shift. In the second scenario, namely *spatial coordination along one axis*, the UASs are prescribed to follow paths of different lengths while adjusting their speeds in order to maintain coordination along one direction. In what follows, we discuss the results in details.

7.2.1 Phase on orbit coordination

In this scenario two quadrotor UASs are required to follow a circular reference of radius 2m. Figure 7.3 shows the desired orbit and the actual trajectories of the two quadrotors. Since the two UASs are tasked to follow the same orbit, a phase-on-orbit separation is required between the two vehicles to avoid collision. This separation is specified on-line from the ground station, and it varies according to mission requirements. The UASs are initially required to keep a 180-deg phase separation (face-to-face); at approximately $t = 94$ s, the required phase separation switches to 90 deg; the two quadrotors keep this configuration for about 14 s, when the required phase separation switches back to 180 deg; finally, in the last part of the experiment, the UASs are required to keep a phase separation of 270 deg. The scenario at hand is depicted in Figure 7.4, which shows the execution of the mission at six different times. A video of the flight test can be found at `http://naira.mechse.illinois.edu/quadrotor-uavs/`. The desired phase-on-orbit separation, along with the actual phase separation between the two UASs, is shown in Figure 7.5. Finally, Figure 7.6 shows the convergence of $\dot{\gamma}_1$ and $\dot{\gamma}_2$ to the desired rate 1, as well as the convergence of the coordination errors to



(a) Schematic of the indoor facility.

(b) CAVR, Naval Postgraduate School.

Figure 7.1: Indoor facility.

(a) AR.Drone quadrotor.

(b) AR.Drone quadrotor with shell.

Figure 7.2: Quadrotor UAS employed in the flight test experiments.

a neighborhood of zero. From these figures it can be noticed that when the desired phase shift changes, the coordination variables slightly deviate from their equilibria, and then converge again after a small transient by virtue of the time-coordination controller.

7.2.2 Spatial coordination along one axis

In this scenario, the AR.Drones are tasked to follow two paths of different lengths, and coordinate along one direction. In particular, UAS_1 is required to follow a straight line of length 5m along the y-axis, while UAS_2 has to follow a semicircle of radius 2.5m. Figure 7.7 depicts the desired and actual paths of UAS_1 and UAS_2. While the mission unfolds, the UASs adjust their speed profiles in order to always face at each other, i.e. coordinate along the y-axis. Figure 7.8 shows the execution of the mission at hand at four different times, verifying the efficacy of the coordination algorithm. The speed profiles are illustrated in Figure 7.9. It can be noticed that the speed of UAS_2 is always greater than the one of UAS_1, since the desired path assigned to the second quadrotor is longer than the one assigned to the first vehicle. Finally, Figure 7.10 depicts the time history of the coordination variables, which remain within a neighborhood of their equilibria as expected.

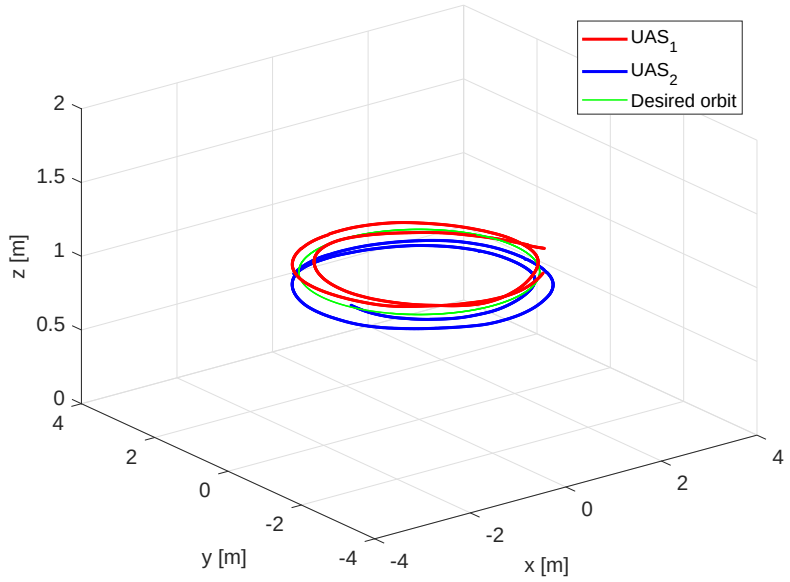


Figure 7.3: Phase on orbit coordination; 3D Paths.

(a) 180 deg separation.

(b) From 180 deg to 90 deg separation.

(c) 90 deg separation.

(d) From 90 deg to 180 deg separation.

(e) 180 deg separation.

(f) 270 deg separation.

Figure 7.4: Phase on orbit coordination; mission execution.

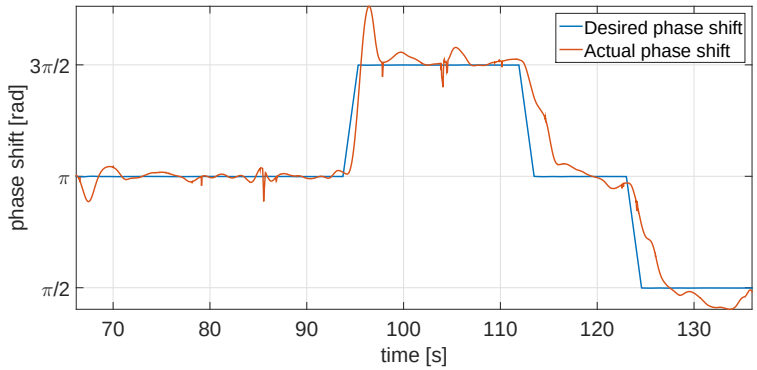


Figure 7.5: Phase shift between two quadrotor UASs.

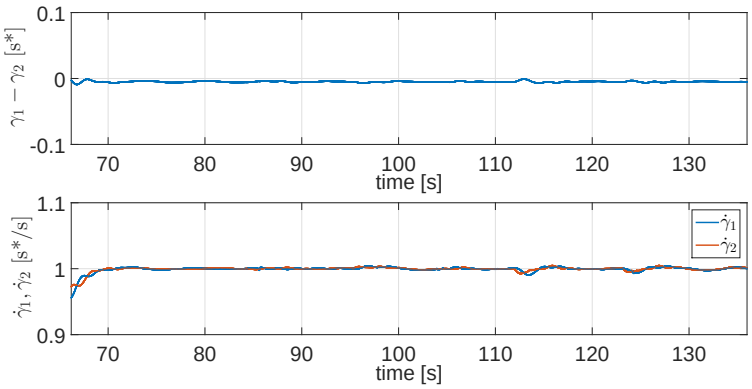


Figure 7.6: Phase on orbit coordination; errors.

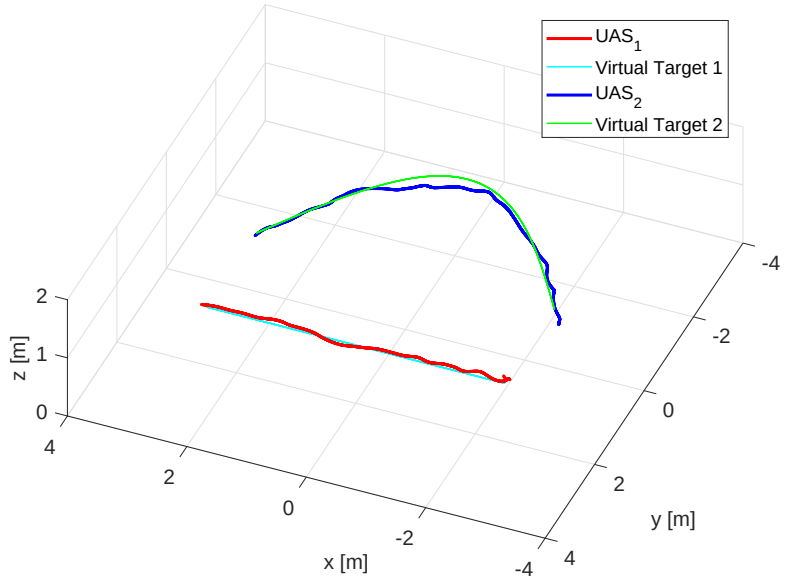
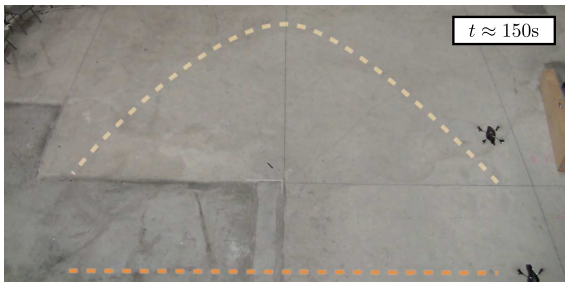
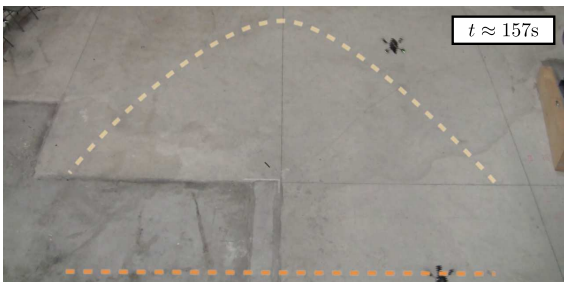


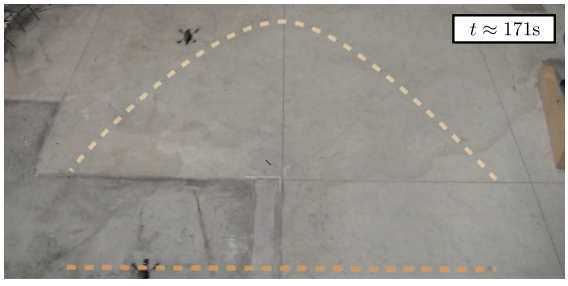
Figure 7.7: Spatial coordination along one axis; 3D Paths.



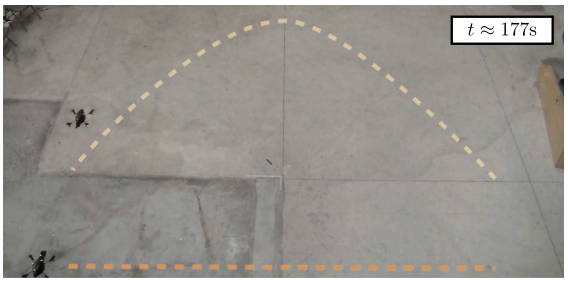
(a)



(b)



(c)



(d)

Figure 7.8: Spatial coordination along one axis; mission execution.

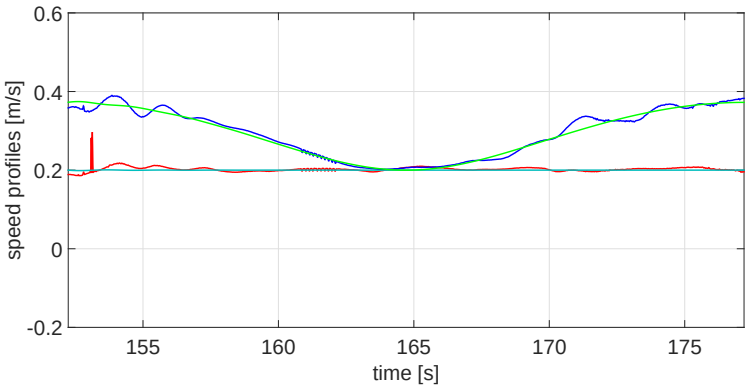


Figure 7.9: Speed profiles of the quadrotor UASs.

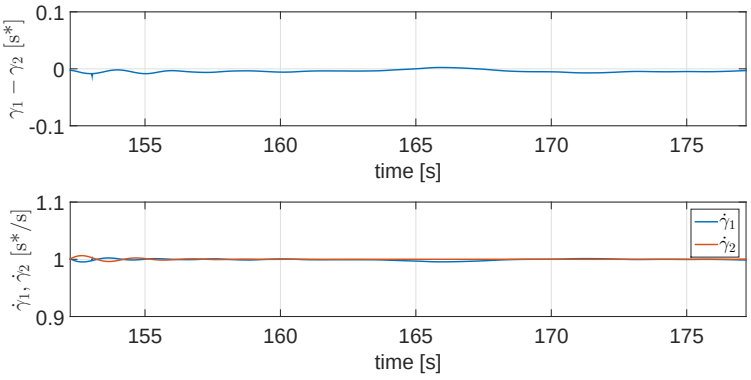


Figure 7.10: Spatial coordination along one axis; errors.

Chapter 8

Conclusions

In this thesis a general framework for control of cooperative autonomous systems has been presented, which allows a group of Unmanned Vehicle Systems (UxSs) to follow desired trajectories while coordinating along them in order to satisfy relative temporal constraints. The described methodology is based on three key steps. Initially, as a result of an optimal motion planning algorithm, each vehicle is assigned a desired trajectory. The desired trajectory satisfies specific mission requirements, boundary conditions, flyability constraints, and feasibility constraints, while minimizing a given cost function. Secondly, virtual target (VT) tracking algorithms implemented on-board the UxSs enable the vehicles to follow their assigned virtual targets running along the desired trajectories. Finally, distributed coordination algorithms adjust the progression of the virtual targets along the desired trajectories to ensure that the vehicles coordinate in order to arrive at the final destination at the same time, or with a predefined temporal separation, according to the mission requirements.

The methodology derived integrates various concepts and tools from a broad spectrum of disciplines, including nonlinear analysis and control, graph theory, optimal control, and numerical analysis, and yields a streamlined design procedure for the control of cooperative autonomous systems. The approach presented applies to teams of heterogeneous systems and departs considerably from well known algorithms used to obtain swarming behaviour, which is unsuitable for most of the mission scenarios envisioned in this thesis.

To solve the optimal motion planning problem, this thesis proposed a numerical method able to efficiently generate feasible and safe trajectories for multiple UxSs. The method is based on direct approximation of a continuous-time optimal control problem into a discrete-time formulation using Bernstein polynomial approximation. These polynomials have favorable geometric properties which allow to efficiently compute minimum distance between curves, saturation constraints, etc., along the entire trajectory. Thus, the proposed approach is particularly convenient for generating trajectories for safe operation of multiple autonomous vehicles in complex environments. Furthermore, Bernstein polynomials possess convergence properties which were used in this thesis to derive convergence results of the discrete solution to the solution of the continuous-time problem.

The coordination problem was formulated as a consensus problem, with the objective of regulating a suitably defined set of coordination variables to zero. Using results from nonlinear systems and graph theory, conditions were derived under which the consensus algorithm proposed solves the coordination problem in the presence of switching communications topologies and communication dropouts. Lower bounds were also derived on the performance of the algorithm as a function of the quality of service (QoS) of the supporting communications network, which in the context of this thesis represents a measure of the level of connectivity of the communications graph. It is relevant to emphasize that the coordination result is independent on the dynamics of the vehicles considered, and on the solution to the VT tracking problem adopted, as long as the VT tracking algorithm satisfies ultimate boundedness properties. Nevertheless, for the sake of completeness, a VT tracking algorithm for multirotor UASs was proposed.

In order to bridge the gap between theory and practice, this thesis included the results of flight tests with multiple multirotor UASs executing cooperative missions. The results illustrated the efficacy of the algorithms proposed. They also demonstrated the validity of the general theoretical framework adopted for control of cooperative autonomous systems in realistic applications as well as the feasibility of the onboard implementation of the algorithms derived.

8.1 Future work

The research developments carried out in this thesis hold promise for the use of UxSs in real world environments. Nevertheless, there is ample room for improvement and intensive research efforts are warranted in a number of topics. A few representative examples are outlined below.

8.1.1 Optimal motion planning

The optimal motion planning algorithm presented in this dissertation requires execution in a centralized fashion. However, for large scale multiple UxSs missions, a centralized motion planning approach has several drawbacks, and a decentralized motion planning architecture is desired or even necessary. Future research will formulate the decentralized optimal motion planning problem as a distributed constrained optimal control problem. Particular emphasis should be placed on the development of a computational framework, based on Bernstein polynomials, and communication protocols that (i) minimize the exchange of information among the UxSs, and (ii) guarantee robustness against partial communication failures and variations of the network topology. Future developments must also account for the presence of uncertainties in the motion planning problem. The framework proposed in this thesis considers multiple vehicles cooperative missions in

deterministic environments, in which the optimal motion planning problem can be formulated as an optimal control problem. However, there may be situations in which it is necessary to solve a more general optimal motion planning problem that cannot be cast into Problem 2, Section 2.2.1. To this end, immediate future research on the approximation method proposed in Chapters 3 and 4 should address generalized stochastic optimal control problems.

8.1.2 Coordinated tracking control

The coordination control problem addressed in Chapter 6 focuses on cooperative missions that require satisfaction of relative temporal specifications. Future developments will address a broader range of missions in which additional absolute temporal requirements are enforced, e.g. arrival within a prescribed time range, or time-varying desired rate of progression of the mission that can be specified on-line by an operator. This extended coordination problem can be addressed by appropriately reformulating the consensus problem introduced in Chapter 2 as a collective tracking problem, or by explicitly controlling the desired rate of change of the coordination states. Additional relevant avenues of research include the derivation of stability and performance guarantees for the proposed coordination algorithm in the presence of quantized information exchange, directed communication graphs, and time-varying switching topologies that do not satisfy the persistency of excitation requirement given in Assumption 1, Section 2.2.2.

8.1.3 Artificial intelligence and optimal decision making

One of the biggest challenges for the integration of autonomous vehicles in real-world environments is the inability of UxSs to autonomously make decisions and take actions in uncertain and unpredicted circumstances. It is therefore important to develop solutions that will allow empowering the UxSs as highly trained systems capable of handling complex problems under high level of uncertainty. To this end, new breakthroughs will be required at the intersection of artificial intelligence and control theory. Of particular interest is the design of an artificial intelligence based software platform that enables multiple UxS decision making based on a smart management of: commanded tasks, UxS own resources, external resources available (other UxSs/sensors or humans), general action constraints, and behavioral rules. The research approach should leverage efforts in deep learning, numerical and variational analysis, and robust and adaptive control in order to provide the autonomous systems with both learning from big data and qualitative model-based decision-making capabilities.

8.1.4 Human-UxS interaction

The interaction between humans and robots (HRI) has been an active area of research during the last few decades [152–154]. In social robotics, HRI has been shown to be a possibility at a social level [155], and robots have been successfully designed in order to take into account humans' perception and comfort level (by, for example, perceiving humans' emotions from their facial expressions, and also conveying emotions by animating a robotic face [156]). Although these results are key for efficient cooperation between humans and robots, it is not possible to use them in the context of UxSs, which are designed from functional point of view and do not have a humanoid aspect. On the other hand, UxSs offer a novel robotic platform thanks to a number of benefits, such as size, agility, payload capacity, flying capabilities, to mention a few. Future work in the broad area of autonomous systems should address the problem of designing UxSs that are perceived as safe by humans. This will pose formidable challenges from human factor, psychology, social science, and engineering perspectives, and will yield new concepts and tools for the design of systems that will enable effective human-UxS interaction. Particular emphasis is to be placed on the derivation of models of human perception as a function of UxSs motion features. The achievement of this goal will not only contribute to the conceptual understanding of human perception of technology and autonomous systems, but will also promote the use of this knowledge for safe operation of UxSs in a human populated environment, thereby extending and combining research from both HRI and autonomous systems.

Appendix A

Mathematical background

A.1 The *Hat* and *Vee* Maps

The *hat map* $(\cdot)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ is defined as

$$(\boldsymbol{x})^\wedge = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

for $\boldsymbol{x} = [x_1,\ x_2,\ x_3]^\top \in \mathbb{R}^3$. The inverse of the hat map is referred to as the *vee map* $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$. A property of the hat and vee maps used in this book is given below:

$$\operatorname{tr}\left[\boldsymbol{M}(\boldsymbol{x})^\wedge\right] = -\boldsymbol{x} \cdot \left(\boldsymbol{M} - \boldsymbol{M}^\top\right)^\vee , \tag{A.1}$$

which holds for any $\boldsymbol{x} \in \mathbb{R}^3$, and $\boldsymbol{M} \in \mathbb{R}^{3\times 3}$. We refer to [145] for further details on the hat and vee maps.

A.2 Bernstein polynomials

The Bernstein basis polynomials of degree N are defined as

$$b_{j,N}(t) = \binom{N}{j} \zeta^j (1-\zeta)^{N-j} , \qquad \zeta \in [0,1] ,$$

for $j = 0, \ldots, N$, with

$$\binom{N}{j} = \frac{N!}{j!(N-j)!} .$$

They were originally introduced by the mathematician Sergei Natanovich Bernstein in 1912 to facilitate a constructive proof of the Weierstrass approximation theorem [157]. An Nth order Bernstein polynomial

$p_N : [0,1] \to \mathbb{R}$ is a linear combination of $N+1$ Bernstein basis polynomials of order N, i.e.

$$p_N(\zeta) = \sum_{j=0}^{N} c_j b_{j,N}(\zeta)\,, \qquad \zeta \in [0,1]\,,$$

where $c_j \in \mathbb{R}$, $j = 0, \ldots, N$, are referred to as Bernstein coefficients (also known as control points). For the sake of generality, and with a slight abuse of terminology, in this thesis we extend the definition of a Bernstein polynomial given above to a polynomial $\boldsymbol{p}_N : [t_0, t_f] \to \mathbb{R}^d$ that is expressed in the following form

$$\boldsymbol{p}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_j b_{j,N}(t)\,, \tag{A.2}$$

defined over the domain $t \in [t_0, t_f]$, with $\boldsymbol{c}_j \in \mathbb{R}^d$, and

$$b_{j,N}(t) = \binom{N}{j} \frac{(t-t_0)^j (t_f - t)^{N-j}}{(t_f - t_0)^N}\,.$$

Bernstein polynomials were popularized by Pierre Bézier in the early 1960s as useful tools for geometric design (Bézier used Bernstein polynomials to design the shape of the cars at the Renault company in France), and are now widely used in computer graphics, animations and type fonts such as postscript fonts and true type fonts. For this reason, the Bernstein polynomial introduced in Equation (A.2) is often referred to as a Bézier curve, especially when used to describe a spatial curve.

Bernstein polynomials possess favorable geometric and numerical properties and computational procedures, which can be exploited in many application domains, including motion planning. In what follows, we provide a review of the properties that are used throughout this thesis. For an extensive review on Bernstein polynomials and their properties the reader is referred to [111].

A.2.1 Properties of Bernstein polynomials

Property 1 (End point values) *The Bernstein polynomial given by Equation* (A.2) *satisfies* $\boldsymbol{p}_N(t_0) = \boldsymbol{c}_0$ *and* $\boldsymbol{p}_N(t_f) = \boldsymbol{c}_N$*. Moreover, the tangent of a Bernstein polynomial at the initial and final points lies on the vectors* $\boldsymbol{c}_1 - \boldsymbol{c}_0$ *and* $\boldsymbol{c}_N - \boldsymbol{c}_{N-1}$*, respectively. A graphical depiction of this property is provided by Figure A.1a, which shows a 2D spatial curve defined by a* 5*th order Bernstein polynomial.* □

Property 2 (Convex hull) *A Bernstein polynomial is completely contained in the convex hull of its Bernstein coefficients (see Figure A.1b).*

□

Property 3 (de Casteljau Algorithm) *The de Casteljau algorithm is an efficient and numerically stable recursive method to evaluate a Bernstein polynomial at any given point. The de Casteljau algorithm is also used to split a Bernstein polynomial into two independent ones. Given an Nth order Bernstein polynomial $\boldsymbol{p}_N : [t_0, t_f] \to \mathbb{R}^d$, and a scalar $t_{\text{div}} \in [t_0, t_f]$, the Bernstein polynomial at t_{div} can be computed using the following recursive relation*

$$\boldsymbol{c}_i^{(0)} = \boldsymbol{c}_i \,, \quad i = 0, \ldots, N$$

$$\boldsymbol{c}_i^{(j)} = \boldsymbol{c}_i^{(j-1)} \frac{t_f - t_{\text{div}}}{t_f - t_0} + \boldsymbol{c}_{i+1}^{(j-1)} \frac{t_{\text{div}} - t_0}{t_f - t_0} \,, \quad i = 0, \ldots, N-j \,, \quad j = 1, \ldots, N \,.$$

Then, the Bernstein polynomial evaluated at t_{div} is given by

$$\boldsymbol{p}_N(t_{\text{div}}) = \boldsymbol{c}_0^{(N)} \,.$$

Moreover, the Bernstein polynomial can be subdivided at t_{div} into two Nth order Bernstein polynomials with Bernstein coefficients

$$\boldsymbol{c}_0^{(0)}, \boldsymbol{c}_0^{(1)}, \ldots, \boldsymbol{c}_0^{(N)} \,, \qquad \textit{and} \qquad \boldsymbol{c}_0^{(N)}, \boldsymbol{c}_1^{(N-1)}, \ldots, \boldsymbol{c}_N^{(0)} \,.$$

Figure A.1c depicts a 2D curve defined by an 5th order Bernstein polynomial (with Bernstein coefficients described by blue circles). The curve is subdivided into two 5th order Bernstein polynomials, each with Bernstein coefficients described by black and red circles.

□

Property 4 (Degree Elevation) *The N th order Bernstein polynomial $\boldsymbol{p}_N(t)$ given by Equation* (A.2) *can be rewritten as a Bernstein polynomial of order $N + r$, for all $r \in \mathbb{Z}^+$, with Bernstein coefficients $\boldsymbol{c}_0^{(r)}, \ldots, \boldsymbol{c}_{N+r}^{(r)}$ given by*

$$\boldsymbol{c}_j^{(r)} = \sum_{i=\max(0,j-r)}^{\min(N,j)} \frac{\binom{r}{j-1}\binom{N}{i}}{\binom{N+r}{j}} \boldsymbol{c}_i \,, \qquad j = 0, \ldots, N+r.$$

□

Property 5 (Product) *The product of an Mth order and an Nth order Bernstein polynomials with Bernstein coefficients $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_M$ and $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_N$ is an $(M+N)$th order Bernstein polynomial with Bernstein coefficients*

$$\boldsymbol{c}_j = \sum_{k=\max(0,j-N)}^{\min(M,j)} \frac{\binom{M}{k}\binom{N}{j-k}}{\binom{M+N}{j}} \boldsymbol{a}_k \boldsymbol{b}_{j-k} \,, \qquad j = 0, \ldots, M+N.$$

□

Property 6 (Derivative of a Bernstein polynomial) *Consider the Bernstein polynomial given by Equation* (A.2)*. Its r-th derivative w.r.t t is a Bernstein polynomial of order $N-r$, which can be easily computed as follows:*

$$\boldsymbol{p}_N^{(r)}(t) = \sum_{j=0}^{N-r} b_{j,N-r}(t)\Delta_N^r \boldsymbol{c}_j \,, \tag{A.3}$$

where Δ_N^r is the weighted forward difference operator

$$\Delta_N^r \boldsymbol{c}_j = \frac{N!}{(N-r)!(t_f-t_0)^r}\Delta^r \boldsymbol{c}_j \,, \tag{A.4}$$

and Δ^r is the forward difference operator w.r.t j, i.e.

$$\Delta^r \boldsymbol{c}_j = \Delta^{r-1}\boldsymbol{c}_{j+1} - \Delta^{r-1}\boldsymbol{c}_j \,.$$

□

Remark 16 *By combining Properties 4 and 6 the derivative of a Nth order Bernstein polynomial can be also expressed as a Nth order Bernstein polynomial*

♦

Property 7 (Integral of a Bernstein polynomial:) *The definite integral of the Bernstein polynomial $\boldsymbol{p}_N(t)$ given by Equation* (A.2) *can be computed as follows:*

$$\int_{t_0}^{t_f} \boldsymbol{p}_N(t)dt = \frac{(t_f-t_0)}{N+1}\sum_{j=0}^{N} \boldsymbol{c}_j \,. \tag{A.5}$$

□

Property 8 (Minimum distance) *The minimum distance between two Bernstein polynomials $\boldsymbol{f}_N(t)$ and $\boldsymbol{g}_N(t)$, with $t \in [t_0, t_f]$, namely*

$$\min_{t_a,t_b\in[t_0,t_f]} ||\boldsymbol{f}_N(t_a) - \boldsymbol{g}_N(t_b)|| \,, \qquad \operatorname*{arg\,min}_{t_a,t_b\in[t_0,t_f]} ||\boldsymbol{f}_N(t_a) - \boldsymbol{g}_N(t_b)|| \,. \tag{A.6}$$

can be efficiently computed by exploiting Properties 2 (convex hull) and 3 (de Casteljau algorithm), in combination with the Gilbert-Johnson-Keerthi (GJK) distance algorithm [158]. The latter is widely used in computer graphics and video games to compute the minimum distance between convex shapes. In [112] the authors propose an iterative procedure that uses the above tools to compute (A.6) *within a desired tolerance.*

This procedure is extremely useful for motion planning applications to efficiently compute the spatial clearance between two paths, or between a path and an obstacle. For example, the minimum distance between the 2D Bernstein polynomial and the point depicted in Figure A.1d is computed in less than 5 ms using an implementation in MATLAB, while the minimum distance between the 3D Bernstein polynomials depicted in Figure A.1e is computed in less than 30 ms. The same procedure can also be employed to compute the extrema (maximum and minimum) of a Bernstein polynomial [113].

□

A.2.2 Bernstein polynomial approximation

Bernstein polynomials can be used to approximate functions. Consider a vector valued function $\boldsymbol{p} : [t_0, t_f] \to \mathbb{R}^d$. The Nth order *Bernstein polynomial approximation* of $\boldsymbol{p}(t)$ is a Bernstein polynomial $\boldsymbol{p}_N(t)$ computed as in (A.2) with $\boldsymbol{c}_j = \boldsymbol{p}(t_j)$, $t_j = \left(t_0 + j\frac{t_f - t_0}{N}\right)$, and $j = 0, \ldots, N$. Namely,

$$\boldsymbol{p}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_j b_{j,N}(t)\,, \qquad \boldsymbol{c}_j = \boldsymbol{p}\left(t_0 + j\frac{t_f - t_0}{N}\right)\,. \tag{A.7}$$

The following results hold for Bernstein polynomial approximations.

Lemma 4 *Let $\boldsymbol{p}(t) \in \mathcal{C}_d^2$. Then, the Bernstein polynomial approximation given by Equation* (A.7) *satisfies*

$$\|\boldsymbol{p}_N(t) - \boldsymbol{p}(t)\| \leq \frac{A}{N}\,,$$

for all $t \in [t_0, t_f]$, where A is independent of N.

■

Proof: We start noticing that the following equality holds:

$$\frac{(t - t_0)(t_f - t)}{N}\dot{b}_{j,N}(t) = \left(t_0 + j\frac{t_f - t_0}{N} - t\right) b_{j,N}(t)\,.$$

Let $\boldsymbol{p}(t) = [p_1(t), \ldots, p_d(t)]$. Consider the ith scalar function of $\boldsymbol{p}(t)$, i.e. $p_i(t)$, and its Bernstein polynomial approximation $p_{N,i}(t)$. Then, one can derive the *generalized* Stancu's remainder formula as follows:

$$\begin{aligned}
p_{N,i}(t) - p_i(t) &= \sum_{j=0}^{N} b_{j,N}(t) \left(p_i \left(t_0 + j\frac{t_f - t_0}{N} \right) - p_i(t) \right) \\
&= \sum_{j=0}^{N} b_{j,N}(t) \left[t_0 + j\frac{t_f - t_0}{N}, t; p_i \right] \left(t_0 + j\frac{t_f - t_0}{N} - t \right) \\
&= \frac{(t - t_0)(t_f - t)}{N} \sum_{j=0}^{N} \dot{b}_{j,N}(t) \left[t_0 + j\frac{t_f - t_0}{N}, t; p_i \right] \\
&= \frac{(t - t_0)(t_f - t)}{(t_f - t_0)} \sum_{j=0}^{N} \left[t_0 + j\frac{t_f - t_0}{N}, t; p_i \right] \left(b_{j-1,N-1}(t) - b_{j,N-1}(t) \right) \\
&= \frac{(t - t_0)(t_f - t)}{(t_f - t_0)} \sum_{j=0}^{N-1} \left(\left[t_0 + j\frac{t_f - t_0}{N}, t; p_i \right] - \left[t_0 + (j+1)\frac{t_f - t_0}{N}, t; p_i \right] \right) b_{j,N-1}(t) \\
&= \frac{(t - t_0)(t_f - t)}{N} \sum_{j=0}^{N-1} \left[t_0 + j\frac{t_f - t_0}{N}, t_0 + (j+1)\frac{t_f - t_0}{N}, t; p_i \right] b_{j,N-1}(t) \,,
\end{aligned} \tag{A.8}$$

where $[x_0, \ldots, x_k; p_i]$ denotes the k-th order divided difference of $p_i(t)$ at the points $x_0, \ldots, x_k$. Thus, the following result holds:

$$|p_{N,i}(t) - p_i(t)| \leq \frac{(t - t_0)(t_f - t)}{2N} \max_{t \in [t_0\,,\, t_f]} |\ddot{p}_i(t)| \,,$$

which proves Lemma 4 with

$$A = \frac{d(t - t_0)(t_f - t)}{2} \max_{t \in [t_0, t_f],\, i = 1, \ldots, d} |\ddot{p}_i(t)|.$$

♠

Lemma 5 *Let $\boldsymbol{p}_N(t)$ be the Bernstein polynomial approximation of the vector valued function $\boldsymbol{p}(t) \in \mathcal{C}_d^{r+2}$ for some $r \in \mathbb{Z}^+$. Let $\boldsymbol{p}^{(r)}(t)$ denote the rth derivative of $\boldsymbol{p}(t)$. The following bound holds:*

$$||\boldsymbol{p}_N^{(r)}(t) - \boldsymbol{p}^{(r)}(t)|| \leq \frac{B}{N} \,,$$

where B is independent of N.

■

Proof: This proof generalizes the proof given in [159, Section 3], where a result similar to the one presented in Lemma 5 with $p : [0, 1] \to \mathbb{R}$, $p \in \mathcal{C}^{r+2}$ is demonstrated.

Let $\boldsymbol{p}(t) = [p_1(t), \ldots, p_d(t)]$, and consider the scalar valued function $p_i(t)$ and its Bernstein polynomial approximation $p_{N,i}(t)$. Let us define the following operator [159]:

$$B_{n,s,m}p_i(t) = \sum_{j=0}^{n-s} \left[t_0 + j\frac{t_f - t_0}{N}, \ldots, t_0 + (j+s)\frac{t_f - t_0}{N}, \underbrace{t, \ldots, t}_{m}; p_i \right] b_{j,n-s}\,. \tag{A.9}$$

Then, Equation (A.8) can be rewritten as follows:

$$p_{N,i}(t) - p_i(t) = \frac{(t-t_0)(t_f - t)}{N} B_{N,1,1}p_i(t)\,.$$

Differentiation of the above equation using the Leibniz rule gives

$$\begin{aligned} p_{N,i}^{(r)}(t) - p_i^{(r)}(t) &= \sum_{k=0}^{r} \binom{r}{k} \frac{d^k}{dt^k} \left(\frac{(t-t_0)(t_f-t)}{N} \right) B_{N,1,1}^{(r-k)} p_i(t) \\ &= \frac{(t-t_0)(t_f-t)}{N} B_{N,1,1}^{(r)} p_i(t) + \frac{r(t_f - 2t + t_0)}{N} B_{N,1,1}^{(r-1)} p_i(t) - \frac{r(r-1)}{N} B_{N,1,1}^{(r-2)} p_i(t)\,. \end{aligned} \tag{A.10}$$

Now we investigate the derivatives of $B_{N,1,1}p_i(t)$. By using the following relationship [160, Chapter 2]

$$\frac{d^r}{dt^r} \left[t_0 + j\frac{t_f - t_0}{N}, t_0 + (j+1)\frac{t_f - t_0}{N}, t; p_i \right] = r! \left[t_0 + j\frac{t_f - t_0}{N}, t_0 + (j+1)\frac{t_f - t_0}{N}, \underbrace{t, \ldots, t}_{r+1}; p_i \right],$$

differentiation of Equation (A.9) with $s = m = 1$ gives

$$\begin{aligned} &(B_{N,1,1}p_i)^{(r)}(t) \\ &= \sum_{j=0}^{N-1} \sum_{k=0}^{r} \binom{r}{k} (r-k)! \left(\left[t_0 + j\frac{t_f - t_0}{N}, t_0 + (j+1)\frac{t_f - t_0}{N}, \underbrace{t, \ldots, t}_{r-k+1}; p_i \right] \right) b_{j,N-1}^{(k)}(t) \\ &= r! \sum_{k=0}^{r} \frac{(N-1)\cdots(N-k)}{k!(t_f - t_0)^k} \sum_{j=0}^{N-k-1} \left(\Delta^k \left[t_0 + j\frac{t_f - t_0}{N}, t_0 + (j+1)\frac{t_f - t_0}{N}, \underbrace{t, \ldots, t}_{r-k+1}; p_i \right] \right) \times b_{j,N-k-1}(t)\,. \end{aligned} \tag{A.11}$$

Notice that

$$\begin{aligned}
&\Delta\left[t_0+j\frac{t_f-t_0}{N},t_0+(j+1)\frac{t_f-t_0}{N},t,\ldots,t;p_i\right]\\
&=\left[t_0+(j+1)\frac{t_f-t_0}{N},t_0+(j+2)\frac{t_f-t_0}{N},t,\ldots,t;p_i\right]\\
&\qquad-\left[t_0+(j+1)\frac{t_f-t_0}{N},t_0+(j+2)\frac{t_f-t_0}{N},t,\ldots,t;p_i\right]\\
&=\frac{2(t_f-t_0)}{N}\left[t_0+j\frac{t_f-t_0}{N},t_0+(j+1)\frac{t_f-t_0}{N},t_0+(j+2)\frac{t_f-t_0}{N},t,\ldots,t;p_i\right],
\end{aligned}\tag{A.12}$$

and continuing to apply Δ implies

$$\begin{aligned}
&\Delta^k\left[t_0+j\frac{t_f-t_0}{N},t_0+(j+1)\frac{t_f-t_0}{N},t,\ldots,t;p_i\right]\\
&\quad=\frac{(k+1)!(t_f-t_0)^k}{N^k}\left[t_0+j\frac{t_f-t_0}{N},\ldots,t_0+(j+k+1)\frac{t_f-t_0}{N},t,\ldots,t;p_i\right].
\end{aligned}$$

Substituting the last result into Equation (A.11) and replacing k by $k-1$ gives

$$\begin{aligned}
(B_{N,1,1}p_i)^{(r)}(t)&=r!\sum_{k=1}^{r+1}k\frac{(N-1)\cdots(N-k+1)}{N^{k-1}}\\
&\qquad\times\sum_{j=0}^{N-k-1}\left(\left[t_0+j\frac{t_f-t_0}{N},\ldots,t_0+(j+k)\frac{t_f-t_0}{N},\underbrace{t,\ldots,t}_{r-k+2};p_i\right]\right)b_{j,N-k}(t)\\
&=r!\sum_{k=1}^{r+1}k\frac{(N-1)\cdots(N-k+1)}{N^{k-1}}B_{N,k,r-k+2}\,.
\end{aligned}$$

From the previous equation we can conclude the following

$$||(B_{N,1,1}p_i)^{(r)}(t)||\le r!\sum_{i=1}^{r+1}k\frac{||p_i^{(r+2)}||}{(r+2)!}\le\frac{||p_i^{(r+2)}||}{2}\,.\tag{A.13}$$

Recalling Equation (A.10), we get

$$|p_{N,i}^{(r)}(t)-p_i^{(r)}(t)|\le\frac{1}{2N}\left((t-t_0)(t_f-t)||p_i^{(r+2)}(t)||+r|t_f-2t+t_0|||p_i^{(r+1)}(t)||+r(r-1)||p_i^{(r)}(t)||\right).$$

Then, Lemma 5 follows with

$$B=\frac{d}{2}\max_{t\in[t_0,t_f],\,i=1,\ldots,n}\left((t-t_0)(t_f-t)||p_i^{(r+2)}||+r|t_f-2t+t_0|||p_i^{(r+1)}||+r(r-1)||p_i^{(r)}||\right).$$

♠

Lemma 6 *Let $\boldsymbol{p}_N(t)$ be the Bernstein polynomial approximation of a vector valued function $\boldsymbol{p}(t) \in \mathcal{C}_d^2$. The following bound holds*

$$\left\| \int_{t_0}^{t_f} \boldsymbol{p}(t)dt - \int_{t_0}^{t_f} \boldsymbol{p}_N(t)dt \right\| \leq \frac{C}{N}\,.$$

where C is independent of N.

■

Proof: We note that

$$\left\| \int_{t_0}^{t_f} \boldsymbol{p}(t)dt - \int_{t_0}^{t_f} \boldsymbol{p}_N(t)dt \right\| \leq \int_{t_0}^{t_f} ||\boldsymbol{p}(t) - \boldsymbol{p}_N(t)||dt \leq \frac{d}{2N} \max_{t\in[t_0,t_f],\, i=1,\ldots,d} |\ddot{p}_i(t)| \int_{t_0}^{t_f} (t-t_0)(t_f-t)dt\,,$$

where we used Lemma 4. Then, Lemma 6 follows by noticing that $\int_{t_0}^{t_f}(t-t_0)(t_f-t)dt = \frac{(t_f-t_0)^3}{6}$.

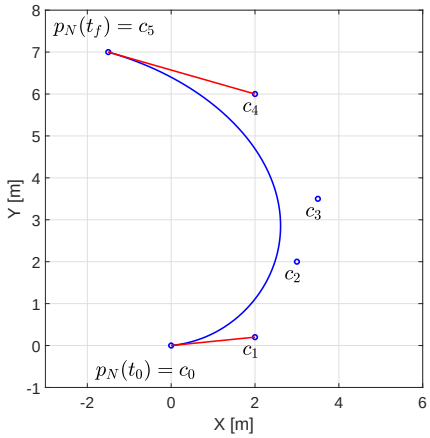
♠

Remark 17 *By using Property 7 with $\boldsymbol{c}_j$ given by Equation* (A.7)*, Lemma 6 can be rewritten as follows:*

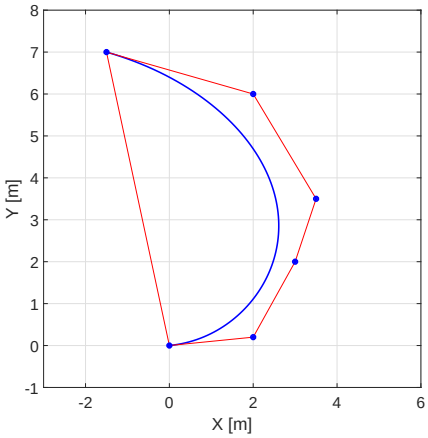
$$\left\| \int_{t_0}^{t_f} \boldsymbol{p}(t)dt - \frac{(t_f-t_0)}{N+1} \sum_{j=0}^{N} \boldsymbol{p}\left(t_0 + j\frac{t_f-t_0}{N}\right) \right\| \leq \frac{C}{N}\,.$$

where C is independent of N.

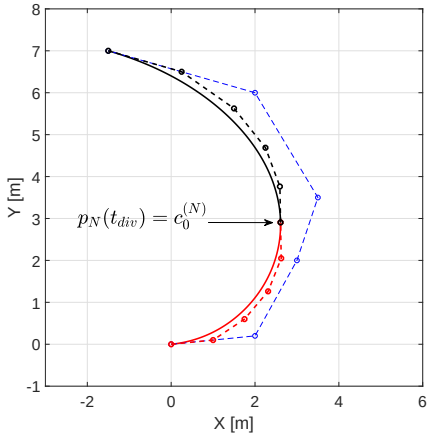
♦



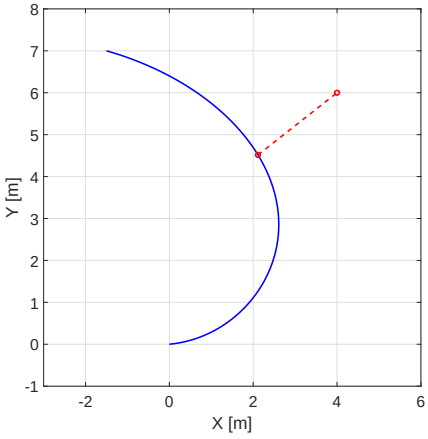
(a) End point values property.



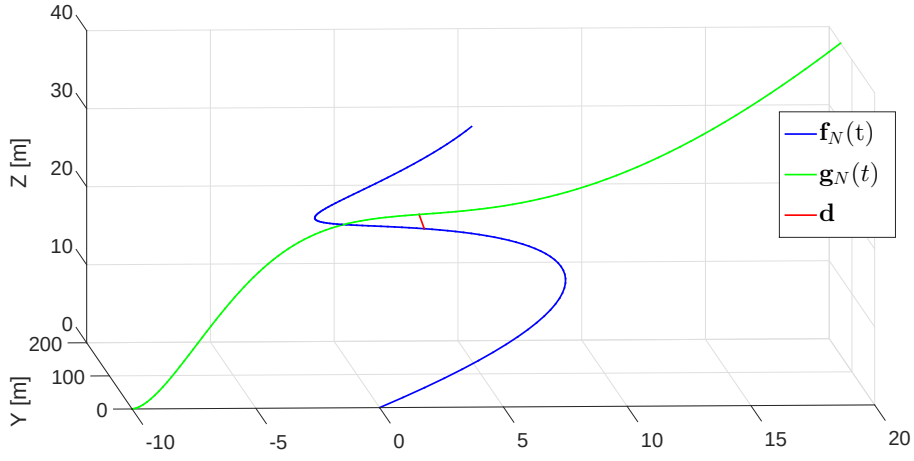
(b) Convex hull property.



(c) de Casteljau algorithm.



(d) Minimum distance to a point.



(e) Distance between two 3D Bernstein polynomials

Figure A.1: 2D and 3D spatial curves defined by Bernstein polynomials, i.e. Bézier curves.

Appendix B

Proofs and derivations

B.1 Proofs and derivations of Chapter 3

B.1.1 Proof of Theorem 1

To prove Theorem 1 it suffices to show that there exists $\boldsymbol{c} = [\boldsymbol{c}_0, \ldots, \boldsymbol{c}_N]$ that satisfies the constraints of Problem P_N^{CV}, namely Equations (3.12) and (3.13). Let $\boldsymbol{y}(t) \in \mathcal{C}_{n_y}^{r+2}$ be a feasible solution to Problem P^{CV}, which exists by assumption (see Assumption 3 in Section 3.1), and define $\boldsymbol{c}_j = \boldsymbol{y}(t_j)$, $t_j = j\frac{t_f}{N}$, $j = 0, \ldots, N$. Then, let

$$\boldsymbol{y}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_j b_{j,N}(t)\,.$$

From Lemmas 4 and 5 in Appendix A.2.2, and Assumption 3 in Section 3.1 it follows that

$$||\boldsymbol{z}_N(t) - \boldsymbol{z}(t)|| \leq \frac{C}{N}\,,$$

where $\boldsymbol{z}(t) = [\boldsymbol{y}(t)^\top, \ldots, \boldsymbol{y}^{(r)}(t)^\top]^\top$, and $\boldsymbol{z}_N(t) = [\boldsymbol{y}_N(t)^\top, \ldots, \boldsymbol{y}_N^{(r)}(t)^\top]^\top$, for some C independent of N. Now consider the inequality constraint given by (3.13). We have

$$\tilde{\boldsymbol{h}}(\boldsymbol{z}_N(t_j)) \leq \tilde{\boldsymbol{h}}(\boldsymbol{z}(t_j)) + ||\tilde{\boldsymbol{h}}(\boldsymbol{z}_N(t_j)) - \tilde{\boldsymbol{h}}(\boldsymbol{z}(t_j))|| \leq L_h \frac{C}{N}\,,$$

where L_h is the Lipschitz constant of $\tilde{\boldsymbol{h}}(\cdot)$ (see Assumption 2 in Section 3.1). Thus, using the properties of exponential growth, there exists N_1 such that for all $N \geq N_1$ the inequality in (3.13) holds. Following a similar argument it can be shown that the equality constraint given by Equation (3.12) is also satisfied, thus proving Theorem 1.

♠

B.1.2 Proof of Theorem 2

This proof is divided into three steps: (1) we show that $\boldsymbol{y}^{\infty}(t)$ is a feasible solution to Problem P^{CV}; (2) we prove that

$$\lim_{N\in V} \tilde{I}_N(\boldsymbol{c}^*) = \tilde{I}(\boldsymbol{y}^{\infty}(t))\,; \tag{B.1}$$

(3) finally, we show that $\tilde{I}(\boldsymbol{y}^{\infty}(t)) = \tilde{I}(\boldsymbol{y}^{*}(t))\,.$

Step (1): We need to show that $\boldsymbol{y}^{\infty}(t)$ satisfies the constraints of Problem P^{CV}, namely Equations (3.8) and (3.9). We start by demonstrating that Equation (3.9) holds, and we do so in a proof by contradiction. Assume that $\boldsymbol{y}^{\infty}(t)$ does not satisfy (3.9). Then, there exists $t' \in [0, t_f]$ such that

$$\tilde{\boldsymbol{h}}(\boldsymbol{z}^{\infty}(t')) > 0\,. \tag{B.2}$$

Since the nodes $\{t_k\}_{k=0}^{N}$ are dense in $[0, t_f]$, for any infinite set V there exists a sequence of indices $\{k_N\}_{N\in V}$ such that

$$\lim_{N\in V} ||\boldsymbol{z}_N^*(t') - \boldsymbol{z}_N^*(t_{k_N})|| = 0\,.$$

Then, we have

$$\begin{aligned}
\tilde{\boldsymbol{h}}(\boldsymbol{z}^{\infty}(t')) &\leq \lim_{N\in V} ||\tilde{\boldsymbol{h}}(\boldsymbol{z}_N^*(t')) - \tilde{\boldsymbol{h}}(\boldsymbol{z}_N^*(t_{k_N}))|| \\
&\quad + \lim_{N\in V} \tilde{\boldsymbol{h}}(\boldsymbol{z}_N^*(t_{k_N})) \\
&\leq \lim_{N\in V} L_h ||\boldsymbol{z}_N^*(t') - \boldsymbol{z}_N^*(t_{k_N})|| + \lim_{N\in V} N^{-\delta_P} = 0\,,
\end{aligned}$$

where we used the fact that $\boldsymbol{z}_N^*(t_{k_N})$ satisfies the constraints in (3.13), and $\tilde{\boldsymbol{h}}(\cdot)$ is Lipschitz. This contradicts (B.2), and in doing so proves that $\boldsymbol{y}^{\infty}(t)$ satisfies the inequality constraint in (3.9). By using an identical argument it can be shown that $\boldsymbol{y}^{\infty}(t)$ satisfies also the equality constraint in (3.8).

Step (2): We need to show that the following equalities hold

$$\begin{aligned}
\tilde{E}(\boldsymbol{z}^{\infty}(0), \boldsymbol{z}^{\infty}(t_f)) &= \lim_{N\in V} \tilde{E}(\boldsymbol{z}_N^*(0), \boldsymbol{z}_N^*(t_N))\,, \\
\int_0^{t_f} \tilde{F}(\boldsymbol{z}^{\infty}(t))dt &= \lim_{N\in V} w \sum_{j=0}^{N} \tilde{F}(\boldsymbol{z}_N^*(t_j))\,.
\end{aligned}$$

The first relationship above follows easily from $\boldsymbol{z}^{\infty}(0) = \lim_{N\in V} \boldsymbol{z}_N^*(0)$ and $\boldsymbol{z}^{\infty}(t_f) = \lim_{N\in V} \boldsymbol{z}_N^*(t_N)$. To prove the second equality, we notice that from Lemma 6 in Appendix A.2.2 (see also Remark 17 in the same

appendix) we have

$$\int_0^{t_f} \tilde{F}(\boldsymbol{z}^{\infty}(t))dt = \lim_{N\in V} w \sum_{j=0}^{N} \tilde{F}(\boldsymbol{z}^{\infty}(t_j))\,,$$

with $w = \frac{t_f}{N+1}$, which combined with the following result

$$\lim_{N\in V} w \sum_{j=0}^{N} \tilde{F}(\boldsymbol{z}^{\infty}(t_j)) = \lim_{N\in V} w \sum_{j=0}^{N} \tilde{F}(\boldsymbol{z}_N^{*}(t_j))\,,$$

proves Equation (B.1).

Step (3): Finally, we need to demonstrate that $\tilde{I}(\boldsymbol{y}^{\infty}(t)) = \tilde{I}(\boldsymbol{y}^{*}(t))$. First, define

$$\tilde{\boldsymbol{y}}_N(t) = \sum_{j=0}^{N} \tilde{\boldsymbol{c}}_j b_{j,N}(t)\,,$$

with $\tilde{\boldsymbol{c}}_j = \boldsymbol{y}^{*}(t_j)$, $j = 0,\ldots,N$, $t_j = j\frac{t_f}{N}$. Similarly to the proof of Theorem 1, one can show that $\tilde{\boldsymbol{c}}$ is a feasible solution of Problem P_N^{CV}. Furthermore, Lemma 6 in Appendix A.2.2 and an argument similar to the one presented in *Step* (2) of this proof yield

$$\tilde{I}(\boldsymbol{y}^{*}(t)) = \lim_{N\in V} \tilde{I}_N(\tilde{\boldsymbol{c}})\,. \tag{B.3}$$

Recall that $\boldsymbol{c}^{*}$ is an optimal solution of Problem P_N^{CV}. Then, we can write

$$\tilde{I}(\boldsymbol{y}^{*}(t)) \leq \tilde{I}(\boldsymbol{y}^{\infty}(t)) = \lim_{N\in V} \tilde{I}_N(\boldsymbol{c}^{*}) \leq \lim_{N\in V} \tilde{I}_N(\tilde{\boldsymbol{c}})\,.$$

The combination of the above expression with Equation (B.3) completes the proof of Theorem 2.

B.2 Proofs and derivations of Chapter 4

B.2.1 Proof of Theorem 3

Let $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ be a feasible solution to Problem P^{OC}, which exists by Assumption 5 in Section 4.1. The goal is to show that there exist $\boldsymbol{c}_x$ and $\boldsymbol{c}_u$ such that the Bernstein polynomials given by

$$\boldsymbol{x}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_{j,x} b_{j,N}(t)\,, \quad \boldsymbol{u}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_{j,u} b_{j,N}(t)\,.$$

satisfy the constraints in (4.7), (4.8), and (4.9). To this end, let us define $\boldsymbol{c}_{k,x} = \boldsymbol{x}(t_k)$ and $\boldsymbol{c}_{k,u} = \boldsymbol{u}(t_k)$, $\forall k \in \{1, \ldots, N\}$. Using Lemmas 4 and 5 in Appendix A.2.2 we have

$$\begin{aligned} ||\boldsymbol{x}_N(t) - \boldsymbol{x}(t)|| &\leq \frac{C_x}{N}\,, \quad ||\boldsymbol{u}_N(t) - \boldsymbol{u}(t)|| \leq \frac{C_u}{N}\,, \\ ||\dot{\boldsymbol{x}}_N(t) - \dot{\boldsymbol{x}}(t)|| &\leq \frac{C_1}{N}\,, \end{aligned} \tag{B.4}$$

for all $t \in [0, t_f]$, where C_x, C_u, C_1 are independent of N. To prove that the dynamic constraint is satisfied, we add and subtract the term $\dot{\boldsymbol{x}}(t_k) - \boldsymbol{f}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k))$ from the left hand side of Equation (4.7), which yields

$$\begin{aligned} ||\dot{\boldsymbol{x}}_N(t_k) - \boldsymbol{f}(\boldsymbol{x}_N(t_k), \boldsymbol{u}_N(t_k))|| \leq\ & ||\dot{\boldsymbol{x}}_N(t_k) - \dot{\boldsymbol{x}}(t_k)|| \\ & + ||\boldsymbol{f}(\boldsymbol{x}_N(t_k), \boldsymbol{u}_N(t_k)) - \boldsymbol{f}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k))|| \\ & + ||\dot{\boldsymbol{x}}(t_k) - \boldsymbol{f}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k))||\,. \end{aligned}$$

The third term in the right hand side of the inequality above is zero (see Equation (4.2)). Moreover, using Equation (B.4) and the fact that $\boldsymbol{f}$ is Lipschitz (see Assumption 4), we get

$$||\dot{\boldsymbol{x}}_N(t_k) - \boldsymbol{f}(\boldsymbol{x}_N(t_k), \boldsymbol{u}_N(t_k))|| \leq \frac{1}{N}(C_1 + L_f(C_x + C_u))\,,$$

where L_f is the Lipschitz constant of $\boldsymbol{f}$. Using the properties of exponential growth, for any $0 < \delta_P < 1$ there exists N_1 such that $\forall N \geq N_1$ we have

$$(C_1 + L_f(C_x + C_u))N^{-1} \leq N^{-\delta_P},$$

which proves that the constraint in Equation (4.7) is satisfied.

The constraint in Equation (4.9) follows easily from the proof of Theorem 1, and by noticing that $\boldsymbol{h}$ is Lipschitz. Finally, using the *end point values* property of Bernstein polynomials (see Property 1 in Appendix A.2.1) we have $\boldsymbol{x}_N(0) = \boldsymbol{c}_{0,x}$ and $\boldsymbol{x}_N(t_f) = \boldsymbol{c}_{N,x}$, which by definition implies that $\boldsymbol{e}(\boldsymbol{x}_N(0), \boldsymbol{x}_N(t_N)) = \boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{0}$, thus proving Equation (4.8). This completes the proof of Theorem

Remark 18 *chp4.thm:existence.*

♠

B.2.2 Proof of Theorem 4

Similarly to the proof of Theorem 2, this proof is divided in three steps.

(1) *we prove that* $(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t))$ *is a feasible solution to Problem* P^{OC}*;*

(2) *we show that*

$$\lim_{N \in V} I^N(\boldsymbol{c}_x^*, \boldsymbol{c}_u^*) = I(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t))\,;$$

(3) *we prove that* $(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t))$ *is an optimal solution to Problem* P^{OC}*, i.e.*

$$I(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t)) = I(\boldsymbol{x}^{*}(t), \boldsymbol{u}^{*}(t))\,.$$

Step (1)***:*** *First, we show that* $(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t))$ *satisfies the dynamic constraint of Problem* P^{OC}*, that is*

$$\dot{\boldsymbol{x}}^{\infty}(t) - \boldsymbol{f}(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t)) = \boldsymbol{0}\,.$$

We show this by contradiction. Assume that the above equality does not hold. Then, there exists t' *such that*

$$||\dot{\boldsymbol{x}}^{\infty}(t') - \boldsymbol{f}(\boldsymbol{x}^{\infty}(t'), \boldsymbol{u}^{\infty}(t'))|| > 0\,. \tag{B.5}$$

Since the time nodes $t_j = \frac{jt_f}{N}$*,* $j = 0, \ldots, N$*, are dense in* $[0, t_f]$*, and* $\dot{\boldsymbol{x}}^{\infty}(t)$*,* $\boldsymbol{x}^{\infty}(t)$*, and* $\boldsymbol{u}^{\infty}(t)$ *are continuous functions by assumption, for some* $k = 0, \ldots, N$ *the left hand side of the above inequality satisfies*

$$||\dot{\boldsymbol{x}}^{\infty}(t') - \boldsymbol{f}(\boldsymbol{x}^{\infty}(t'), \boldsymbol{u}^{\infty}(t'))|| = \lim_{N \in V} ||\dot{\boldsymbol{x}}_N^*(t_k) - \boldsymbol{f}(\boldsymbol{x}_N^*(t_k), \boldsymbol{u}_N^*(t_k))||.$$

However, the dynamic constraint in Problem P_N^{OC} *is*

$$||\dot{\boldsymbol{x}}_N^*(t_k) - \boldsymbol{f}(\boldsymbol{x}_N^*(t_k), \boldsymbol{u}_N^*(t_k))|| \leq N^{-\delta_P}, \quad 0 < \delta_P < 1,$$

which implies

$$\lim_{N \in V} ||\dot{\boldsymbol{x}}_N^*(t_k) - \boldsymbol{f}(\boldsymbol{x}_N^*(t_k), \boldsymbol{u}_N^*(t_k))|| = \lim_{N \in V} N^{-\delta_P} = 0.$$

The above result contradicts Equation (B.5)*, and in doing so proves that* $(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t))$ *satisfies the dynamic constraint in Equation* (4.2)*. The equality and inequality constraints in* (4.8) *and* (4.9) *follow easily by a similar argument (see also the proof of Theorem 2).*

Step (2)***:*** *In this part we need to prove the following:*

$$\lim_{N \in V} \left(E(\boldsymbol{x}_N^*(0), \boldsymbol{x}_N^*(t_N)) + w \sum_{j=0}^{N} F(\boldsymbol{x}_N^*(t_j), \boldsymbol{u}_N^*(t_j)) \right) = E(\boldsymbol{x}^{\infty}(0), \boldsymbol{x}^{\infty}(t_f)) + \int_0^{t_f} F(\boldsymbol{x}^{\infty}(t), \boldsymbol{u}^{\infty}(t))dt\,. \tag{B.6}$$

Let us focus on the integral terms in the equation above. Notice that from Lemma 6 in Appendix A.2.2 (see also Remark 17 in the same appendix) and the fact that $F \in \mathcal{C}^2$ (see Assumption 4) we get

$$\int_0^{t_f} F(\boldsymbol{x}^\infty(t), \boldsymbol{u}^\infty(t))dt = \lim_{N\in V} \sum_{j=0}^{N} wF(\boldsymbol{x}^\infty(t_j), \boldsymbol{u}^\infty(t_j)) = \lim_{N\in V} \sum_{j=0}^{N} wF(\boldsymbol{x}_N^*(t_j), \boldsymbol{u}_N^*(t_j))\,.$$

Similarly, using the Lipschitz assumption on E, one can conclude that

$$\lim_{N\in V} E(\boldsymbol{x}_N^*(0), \boldsymbol{x}_N^*(t_N)) = E(\boldsymbol{x}^\infty(0), \boldsymbol{x}^\infty(t_f))\,,$$

which, in turn, demonstrates Equation (B.6)*.*

Step (3)***:*** *Finally, it remains to be shown that*

$$I(\boldsymbol{x}^\infty(t), \boldsymbol{u}^\infty(t)) = I(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t))\,.$$

Let us define $\tilde{\boldsymbol{c}}_{k,x} = \boldsymbol{x}^*(t_k)$ and $\tilde{\boldsymbol{c}}_{k,u} = \boldsymbol{u}^*(t_k)$, $\forall k \in \{1, \ldots, N\}$. Then, following an argument similar to the one in the proof of Theorem 3, one can show that there exists N_1 such that for any $N \geq N_1$ the pair $(\tilde{\boldsymbol{c}}_x, \tilde{\boldsymbol{c}}_u)$ is a feasible solution to Problem P_N^{OC}. Moreover, Lemma 6 in Appendix A.2.2 and an argument similar to Step (2) *yield*

$$\lim_{N\in V} I^N(\tilde{\boldsymbol{c}}_x, \tilde{\boldsymbol{c}}_u) = I(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t))\,. \tag{B.7}$$

Recall that, by definition, $\boldsymbol{c}_x^*$ and $\boldsymbol{c}_u^*$ are optimal solutions to Problem P_N^{OC}, which implies that

$$I(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)) \leq I(\boldsymbol{x}^\infty(t), \boldsymbol{u}^\infty(t)) = \lim_{N\in V} I^N(\boldsymbol{c}_x^*, \boldsymbol{c}_u^*) \leq \lim_{N\in V} I^N(\tilde{\boldsymbol{c}}_x, \tilde{\boldsymbol{c}}_u)\,. \tag{B.8}$$

The last inequality, combined with (B.7)*, gives*

$$I(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)) = I(\boldsymbol{x}^\infty(t), \boldsymbol{u}^\infty(t))\,,$$

which completes the proof of Theorem 4.

♠

B.3 Proofs and derivations of Chapter 5

B.3.1 Proof of Lemma 1

Consider the vector $\hat{\mathbf{b}}_{\mathbf{3D}}$*, defined as follows:*

$$\hat{\mathbf{b}}_{\mathbf{3D}} = \frac{(k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)}{||(k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)||}\,, \tag{B.9}$$

where k_p*,* $k_v > 1$*, and*

$$s_p = \begin{cases} sign(\boldsymbol{e}_p^\top \hat{\mathbf{e}}_{\mathbf{3}})\,, & if \quad ||k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)|| = 0 \\ 0 & otherwise \end{cases},$$

$$s_v = \begin{cases} sign(\boldsymbol{e}_v^\top \hat{\mathbf{e}}_{\mathbf{3}})\,, & if \quad ||k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_{\boldsymbol{d}}(\gamma)|| = 0 \\ 0 & otherwise \end{cases}.$$

The vector in (B.9) *is not defined if*

$$(k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma) = \mathbf{0}\,. \tag{B.10}$$

Therefore, we need to show that Equation (B.10) *is never verified. Let us write the previous vector equation as three different scalar equations as follows:*

$$\begin{cases} ((k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + m\ddot{\boldsymbol{p}}_d(\gamma))^\top \hat{\mathbf{e}}_{\mathbf{1}} = 0 \\ ((k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + m\ddot{\boldsymbol{p}}_d(\gamma))^\top \hat{\mathbf{e}}_{\mathbf{2}} = 0 \\ ((k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma))^\top \hat{\mathbf{e}}_{\mathbf{3}} = 0\,. \end{cases} \tag{B.11}$$

From the definition of s_p *and* s_v*, the last equation above can be written as follows, depending on the value of* $||k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)||$*:*

$$\begin{cases} |\boldsymbol{e}_p^\top \hat{\mathbf{e}}_{\mathbf{3}}| + |\boldsymbol{e}_v^\top \hat{\mathbf{e}}_{\mathbf{3}}| = 0, & if\ ||k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)|| = 0 \\ (k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma))^\top \hat{\mathbf{e}}_{\mathbf{3}} = 0, & if\ ||k_p\boldsymbol{e}_p + k_v\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)|| \neq 0\,. \end{cases}$$

Since we assumed that $||\ddot{\boldsymbol{p}}_d(\gamma)|| < g$ *(see Equation* (2.18)*), the first condition above is never verified. Moreover, we notice that the second condition cannot hold without violating either the first or second equation in*

(B.11), *which completes the proof.*

B.3.2 Proof of Lemma 2

We start noticing that in Ω_c *the following bounds hold:*

$$\Psi(\tilde{\boldsymbol{R}}) \leq c^2 < \frac{1}{2}\,, \tag{B.12}$$

$$||\boldsymbol{e}_{\tilde{R}}||^2 \leq \Psi(\tilde{\boldsymbol{R}}) \leq \frac{1}{1-c^2}||\boldsymbol{e}_{\tilde{R}}||^2\,, \tag{B.13}$$

$$||\boldsymbol{e}_p|| \leq e_{p\max}\,. \tag{B.14}$$

Then, we note that if (B.12) *is verified, the following inequality holds:*

$$||\boldsymbol{e}_{\tilde{R}}|| \leq c^2\,. \tag{B.15}$$

Let us choose the following Lyapunov candidate function:

$$V = \frac{k_p}{2}||\boldsymbol{e}_p||^2 + \frac{m}{2}||\boldsymbol{e}_v||^2 + \Psi(\tilde{\boldsymbol{R}}) + c_1\boldsymbol{e}_p^\top\boldsymbol{e}_v\,, \tag{B.16}$$

where $c_1, k_p, m > 0$ *were defined in Chapter 5. The bound in* (B.13) *allows us to write the Lyapunov candidate function as follow:*

$$\boldsymbol{x_{pf}}^\top\boldsymbol{W_1}\boldsymbol{x_{pf}} \leq V(\boldsymbol{x_{pf}}) \leq \boldsymbol{x_{pf}}^\top\boldsymbol{W_2}\boldsymbol{x_{pf}}\,, \tag{B.17}$$

where $\boldsymbol{x_{pf}} = [\boldsymbol{e}_p^\top\,,\, \boldsymbol{e}_v^\top\,,\, \boldsymbol{e}_{\tilde{R}}^\top]^\top$, *and*

$$\boldsymbol{W_1} = \begin{bmatrix} \frac{k_p}{2} & -\frac{c_1}{2} & 0 \\ -\frac{c_1}{2} & \frac{m}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{W_2} = \begin{bmatrix} \frac{k_p}{2} & -\frac{c_1}{2} & 0 \\ -\frac{c_1}{2} & \frac{m}{2} & 0 \\ 0 & 0 & \frac{1}{1-c^2} \end{bmatrix}.$$

The derivative of the Lyapunov function follows:

$$\dot{V} = k_p\boldsymbol{e}_p^\top\dot{\boldsymbol{e}}_p + m\boldsymbol{e}_v^\top\dot{\boldsymbol{e}}_v + \boldsymbol{e}_{\tilde{R}}^\top\tilde{\boldsymbol{\omega}} + c_1\boldsymbol{e}_v^\top\dot{\boldsymbol{e}}_p + c_1\boldsymbol{e}_p^\top\dot{\boldsymbol{e}}_v\,. \tag{B.18}$$

Next, consider

$$m\dot{\boldsymbol{e}}_v = m\ddot{\boldsymbol{p}}_d(\gamma) + mg\hat{\mathbf{e}}_{\mathbf{3}} - T\hat{\mathbf{b}}_{\mathbf{3}}\,. \tag{B.19}$$

Adding and subtracting the term $\frac{T\hat{\mathbf{b}}_{\mathbf{3D}}}{\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}}$ *we get:*

$$m\dot{\boldsymbol{e}}_v = m\ddot{\boldsymbol{p}}_d(\gamma) + mg\hat{\mathbf{e}}_{\mathbf{3}} - \frac{T\hat{\mathbf{b}}_{\mathbf{3D}}}{\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}} - \frac{T}{\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}}\left((\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}})\hat{\mathbf{b}}_{\mathbf{3}} - \hat{\mathbf{b}}_{\mathbf{3D}}\right). \tag{B.20}$$

Let $\boldsymbol{X} = \frac{T}{\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}}\left((\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}})\hat{\mathbf{b}}_{\mathbf{3}} - \hat{\mathbf{b}}_{\mathbf{3D}}\right)$*. Notice that* $\boldsymbol{X}$ *is bounded as follows [145]:*

$$||\boldsymbol{X}|| < ((k_p + s_p)||\boldsymbol{e}_p|| + (k_v + s_v)||\boldsymbol{e}_v|| + 2mg)||\boldsymbol{e}_{\tilde{R}}||\,. \tag{B.21}$$

Let also

$$\boldsymbol{A} = -\left((k_p + s_p)\boldsymbol{e}_p + (k_v + s_v)\boldsymbol{e}_v + mg\hat{\mathbf{e}}_{\mathbf{3}} + m\ddot{\boldsymbol{p}}_d(\gamma)\right).$$

Then, from the definition of $\hat{\mathbf{b}}_{\mathbf{3D}}$ *and* T *in* (5.5) *and* (5.11)*, we note that* $T = -\boldsymbol{A}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}$*,* $\hat{\mathbf{b}}_{\mathbf{3D}} = -\frac{\boldsymbol{A}}{||\boldsymbol{A}||}$ *and* $-\boldsymbol{A} = ||\boldsymbol{A}||\hat{\mathbf{b}}_{\mathbf{3D}}$*. Therefore, we get*

$$\frac{T\hat{\mathbf{b}}_{\mathbf{3D}}}{\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}} = \frac{(-\boldsymbol{A}^{\top}\hat{\mathbf{b}}_{\mathbf{3}})\hat{\mathbf{b}}_{\mathbf{3D}}}{\hat{\mathbf{b}}_{\mathbf{3D}}{}^{\top}\hat{\mathbf{b}}_{\mathbf{3}}} = ||\boldsymbol{A}||\hat{\mathbf{b}}_{\mathbf{3D}} = -\boldsymbol{A}.$$

Then, equation (B.20) *becomes:*

$$m\dot{\boldsymbol{e}}_v = -(k_p + s_p)\boldsymbol{e}_p - (k_v + s_v)\boldsymbol{e}_v - \boldsymbol{X}\,, \tag{B.22}$$

which allows us to rewrite the derivative of the Lyapunov function as follows:

$$\begin{aligned}\dot{V} = \big(&k_p\boldsymbol{e}_x^{\top}\boldsymbol{e}_v + \boldsymbol{e}_v^{\top}\left(-(k_p + s_p)\boldsymbol{e}_p - (k_v + s_v)\boldsymbol{e}_v - \boldsymbol{X}\right) + \frac{1}{2}\boldsymbol{e}_{\tilde{R}}^{\top}\tilde{\boldsymbol{\omega}} + c_1\boldsymbol{e}_v^{\top}\boldsymbol{e}_v \\ &+ \frac{c_1}{m}\boldsymbol{e}_p^{\top}\left(-(k_p + s_p)\boldsymbol{e}_p - (k_v + s_v)\boldsymbol{e}_v - \boldsymbol{X}\right)\big)\,.\end{aligned}$$

Substituting the control law for the angular rate introduced in (5.12)*, straightforward computations lead to*

$$\begin{aligned}\dot{V} \leq \Bigg(&-\frac{c_1(k_p + s_p)}{m}||\boldsymbol{e}_p||^2 - (k_v + s_v - c_1)||\boldsymbol{e}_v||^2 - k_{\tilde{R}}||\boldsymbol{e}_{\tilde{R}}||^2 \\ &+ \left(\frac{c_1(k_v + s_v)}{m} + s_p\right)||\boldsymbol{e}_p||||\boldsymbol{e}_v|| + ||\boldsymbol{X}||\left(\frac{c_1}{m}||\boldsymbol{e}_p|| + ||\boldsymbol{e}_v||\right)\Bigg)\,.\end{aligned}$$

Finally, the bounds in (B.15) *and* (B.21) *give*

$$\begin{aligned}\dot{V} \leq & -\frac{c_1(k_p-1)}{m}(1-c^2)||\boldsymbol{e}_p||^2 - ((k_v-1)(1-c^2)-c_1)||\boldsymbol{e}_v||^2 \\ & - k_{\tilde{R}}||\boldsymbol{e}_{\tilde{R}}||^2 + \left(\frac{c_1(k_v+1)}{m}(1+c^2)+1\right)||\boldsymbol{e}_p||||\boldsymbol{e}_v|| \\ & + ((k_p+1)e_{p\max}+2mg)||\boldsymbol{e}_{\tilde{R}}||||\boldsymbol{e}_v|| + 2gc_1||\boldsymbol{e}_{\tilde{R}}||||\boldsymbol{e}_p|| \\ = & - x_{pf}^\top Q x_{pf}\,,\end{aligned} \tag{B.23}$$

with Q being defined as follows:

$$\boldsymbol{Q} = \begin{bmatrix} \frac{c_1(k_p-1)}{m}(1-c^2) & -\frac{1}{2}\left[\frac{c_1(k_v+1)}{m}(1+c^2)+1\right] & -gc_1 \\ -\frac{1}{2}\left[\frac{c_1(k_v+1)}{m}(1+c^2)+1\right] & (k_v-1)(1-c^2)-c_1 & -\frac{k_p+1}{2}e_{p\max}-mg \\ -gc_1 & -\frac{k_p+1}{2}e_{p\max}-mg & k_{\tilde{R}} \end{bmatrix}. \tag{B.24}$$

By properly choosing k_p, k_v, and $k_{\tilde{R}}$, one can show that the following inequality holds

$$\boldsymbol{Q} - 2\lambda_{pf}\boldsymbol{W_2} \geq 0\,, \tag{B.25}$$

where λ_{pf} and $\boldsymbol{W_2}$ were defined in (5.14) *and* (5.17) *respectively, and the matrix $\boldsymbol{Q}$ is defined in* (B.24)

A proof of the existence of k_p, k_v, and $k_{\tilde{R}}$ that satisfy inequality (B.25) *is given in Appendix B.3.3.*

We can therefore upper bound the derivative of the Lyapunov function as follows:

$$\dot{V}(t) \leq -2\lambda_{pf}V(t)\,.$$

Using the Comparison Lemma, the following result holds:

$$V(t) \leq V(0)e^{-2\lambda_{pf}t}\,,$$

which implies that the following inequality

$$||\boldsymbol{x_{pf}}(t)|| \leq \sqrt{\frac{\lambda_{\max}(\boldsymbol{W_2})}{\lambda_{\min}(\boldsymbol{W_1})}}||\boldsymbol{x_{pf}}(0)||e^{-\lambda_{pf}t} \tag{B.26}$$

is verified for any $t \geq 0$, which completes the proof.

B.3.3 Proof of inequality (B.25)

We need to prove that the following inequality holds:

$$\boldsymbol{Q} - 2\lambda_{pf}\boldsymbol{W_2} = \begin{bmatrix} \frac{c_1(k_p-1)}{m}(1-c^2) - \lambda_{pf}k_p & -\frac{c_1(k_v+1)}{2m}(1+c^2) - \frac{1}{2} - \lambda_{pf}c_1 & -gc_1 \\ -\frac{c_1(k_v+1)}{2m}(1+c^2) - \frac{1}{2} - \lambda_{pf}c_1 & (k_v-1)(1-c^2) - c_1 - \lambda_{pf}m & -\frac{k_p+1}{2}e_{p\max} - mg \\ -gc_1 & -\frac{k_p+1}{2}e_{p\max} - mg & k_{\tilde{R}} - 2\frac{\lambda_{pf}}{1-c^2} \end{bmatrix} \geq 0\,. \tag{B.27}$$

We notice that since $\lambda_{pf} < c_1(1-c^2)/m$, by letting $k_v > 1 + \frac{c_1+\lambda_{pf}m}{1-c^2}$, there exists some k_p such that the upper left 2×2 corner of $(\boldsymbol{Q}-2\lambda_{pf}\boldsymbol{W_2})$ is positive definite. Finally, since the only element of $\boldsymbol{Q}-2\lambda_{pf}\boldsymbol{W_2}$ that depends on $k_{\tilde{R}}$ is its $(3,3)$ entry, then it can be shown that there is some $k_{\tilde{R}}$ such that $\det(\boldsymbol{Q}-2\lambda_{pf}\boldsymbol{W_2}) > 0$.

B.3.4 Proof of Lemma 3

Start by considering the Lyapunov function in (B.16), with time derivative given by

$$\dot{V} = k_p\boldsymbol{e}_p^\top\dot{\boldsymbol{e}}_p + m\boldsymbol{e}_v^\top\dot{\boldsymbol{e}}_v + \boldsymbol{e}_{\tilde{R}}^\top\tilde{\boldsymbol{\omega}} + c_1\boldsymbol{e}_v^\top\dot{\boldsymbol{e}}_p + c_1\boldsymbol{e}_p^\top\dot{\boldsymbol{e}}_v\,. \tag{B.28}$$

At this point, we notice that

$$\boldsymbol{e}_{\tilde{R}}^\top\tilde{\boldsymbol{\omega}} = \boldsymbol{e}_{\tilde{R}}^\top\left(\begin{bmatrix}p\\q\\r\end{bmatrix} - \tilde{\boldsymbol{R}}^\top\{\boldsymbol{\omega_{D/I}}\}_D\right) = \boldsymbol{e}_{\tilde{R}}^\top\left(\begin{bmatrix}p-p_c\\q-q_c\\r-r_c\end{bmatrix} + \begin{bmatrix}p_c\\q_c\\r_c\end{bmatrix} - \tilde{\boldsymbol{R}}^\top\{\boldsymbol{\omega_{D/I}}\}_D\right)\,. \tag{B.29}$$

Letting p_c, q_c, r_c be governed by the control law given in (5.12), the previous equation becomes:

$$\boldsymbol{e}_{\tilde{R}}^\top\tilde{\boldsymbol{\omega}} = \boldsymbol{e}_{\tilde{R}}^\top\left(\begin{bmatrix}p-p_c\\q-q_c\\r-r_c\end{bmatrix} - 2k_{\tilde{R}}\boldsymbol{e}_{\tilde{\boldsymbol{R}}}\right)\,.$$

Similarly, consider the term

$$\begin{aligned} m\dot{\boldsymbol{e}}_v &= m\ddot{\boldsymbol{p}}_d(\gamma) + mg\hat{\mathbf{e}}_{\mathbf{3}} - T\hat{\mathbf{b}}_{\mathbf{3}} + T_c\hat{\mathbf{b}}_{\mathbf{3}} - T_c\hat{\mathbf{b}}_{\mathbf{3}} \\ &= m\ddot{\boldsymbol{p}}_d(\gamma) + mg\hat{\mathbf{e}}_{\mathbf{3}} - T_c\hat{\mathbf{b}}_{\mathbf{3}} - (T - T_c)\hat{\mathbf{b}}_{\mathbf{3}} \\ &= -(k_p + s_p)\boldsymbol{e}_p - (k_v + s_v)\boldsymbol{e}_v - \boldsymbol{X} - (T - T_c)\hat{\mathbf{b}}_{\mathbf{3}}\,. \end{aligned} \tag{B.30}$$

Substituting (B.29) *and* (B.30) *in* (B.28)*, and after some algebraic manipulations similar to the ones outlined in Appendices B.3.2 and B.3.3, we get:*

$$\dot{V} \leq -2\lambda_{pf}V + \gamma_\omega||\boldsymbol{e}_{\tilde{R}}|| + \gamma_T(||\boldsymbol{e}_v|| + \frac{c_1}{m}||\boldsymbol{e}_p||)\,,$$

which can be rewritten as

$$\begin{aligned} \dot{V} \leq\ & -2\lambda_{pf}(1-\delta_\lambda)V \\ & - 2\lambda_{pf}\delta_\lambda\left(\frac{k_p}{2}||\boldsymbol{e}_p||^2 + \frac{m}{2}||\boldsymbol{e}_v||^2 + \frac{1}{1-c^2}||\boldsymbol{e}_{\tilde{R}}||^2 + c_1\boldsymbol{e}_p^\top\boldsymbol{e}_v\right) + \gamma_\omega||\boldsymbol{e}_{\tilde{R}}|| + \gamma_T\left(||\boldsymbol{e}_v|| + \frac{c_1}{m}||\boldsymbol{e}_p||\right)\,, \end{aligned}$$

where δ_λ *satisfies* $0 < \delta_\lambda < 1$*. If follows that, for all* $\boldsymbol{x_{pf}}$ *satisfying*

$$-2\lambda_{pf}\delta_\lambda\left(\frac{k_p}{2}||\boldsymbol{e}_p||^2 + \frac{m}{2}||\boldsymbol{e}_v||^2 + \frac{1}{1-c^2}||\boldsymbol{e}_{\tilde{R}}||^2 + c_1\boldsymbol{e}_p^\top\boldsymbol{e}_v\right) + \gamma_\omega||\boldsymbol{e}_{\tilde{R}}|| + \gamma_T\left(||\boldsymbol{e}_v|| + \frac{c_1}{m}||\boldsymbol{e}_p||\right) \leq 0\,, \tag{B.31}$$

the derivative of the Lyapunov function is lower bounded as follows

$$\dot{V} \leq -2\lambda_{pf}(1-\delta_\lambda)V\,.$$

We notice that inequality (B.31) *can be rewritten as*

$$\begin{aligned} 2\lambda_{pf}\delta_\lambda\Bigg(&\frac{k_p}{2}||\boldsymbol{e}_p||\left(||\boldsymbol{e}_p|| - \frac{\gamma_T c_1}{k_p\lambda_{pf}\delta_\lambda}\right) + \frac{m}{2}||\boldsymbol{e}_v||\left(||\boldsymbol{e}_v|| - \frac{\gamma_T}{m\lambda_{pf}\delta_\lambda}\right) \\ &+||\boldsymbol{e}_{\tilde{R}}||\left(\frac{1}{1-c^2}||\boldsymbol{e}_{\tilde{R}}|| - \frac{\gamma_\omega}{2\lambda_{pf}\delta_\lambda}\right) + c_1\boldsymbol{e}_p^\top\boldsymbol{e}_v\Bigg) \geq 0\,, \end{aligned}$$

which is satisfied outside the closed set D *defined by*

$$D := \left\{(\boldsymbol{x_{pf}} \in \mathbb{R}^9 \;\;|\;\; \|\boldsymbol{e}_p\| \leq \frac{\gamma_T c_1}{k_x\lambda_{pf}\delta_\lambda}\,, \quad \|\boldsymbol{e_v}\| \leq \frac{\gamma_T}{m\lambda_{pf}\delta_\lambda}\,, \quad \|\boldsymbol{e}_{\tilde{R}}\| \leq \frac{\gamma_\omega(1-c^2)}{2\lambda_{pf}\delta_\lambda}\right\}\,,$$

which is contained inside the compact set

$$\Omega_b := \left\{ (\boldsymbol{x_{pf}} \in \mathbb{R}^9 \;\; | \;\; \|\boldsymbol{x_{pf}}\| \leq \frac{(c_1/m+1)\gamma_T + \gamma_\omega}{\lambda_{pf}\lambda_{\min}(\boldsymbol{W_2})\delta_\lambda} \right\} .$$

Then, the design constraints for the performance bounds γ_ω and γ_T given in (5.18) *imply that the set Ω_b is contained in Ω_{pf}. Finally, using a proof similar to that of [161, Theorem 4.18], it can be shown that for any initial state $\boldsymbol{x_{pf}}(0) \in \Omega_{pf}$, there is a time $T_b \geq 0$ such that the following bounds are satisfied*

$$\begin{aligned} &||\boldsymbol{x_{pf}}(t)|| \leq k_{pf}||\boldsymbol{x_{pf}}(0)||\mathrm{e}^{-\lambda_{pf}(1-\delta_\lambda)t}, && \text{for all } 0 \leq t < T_b, \\ &||\boldsymbol{x_{pf}}(t)|| \leq \rho, && \text{for all } t \geq T_b, \end{aligned}$$

with

$$k_{pf} \triangleq \sqrt{\frac{\lambda_{\max}(\boldsymbol{W_2})}{\lambda_{\min}(\boldsymbol{W_1})}},$$

and

$$\rho \triangleq \sqrt{\frac{\lambda_{\max}(\boldsymbol{W_2})}{\lambda_{\min}(\boldsymbol{W_1})}} \left(\frac{(c_1/m+1)\gamma_T + \gamma_\omega}{\lambda_{pf}\lambda_{\min}(\boldsymbol{W_2})\delta_\lambda} \right),$$

which completes the proof.

B.4 Proofs and derivations of Chapter 6

B.4.1 Proof of Theorem 5

Consider the following system

$$\dot{\boldsymbol{\phi}}(t) = -\frac{a}{b}\bar{\boldsymbol{L}}\boldsymbol{\phi}(t), \tag{B.32}$$

where the matrix $\bar{\boldsymbol{L}}(t) = \boldsymbol{Q}\boldsymbol{L}(t)\boldsymbol{Q}^\top$ satisfies the (PE)-like condition in Assumption 1, Section 2.2.2. Then, using the result reported in [162, Lemma 5], we conclude that the system in (B.32) *is GUES (globally uniformly exponentially stable), and that the following bound holds:*

$$||\boldsymbol{\phi}(t)|| \leq k_\lambda ||\boldsymbol{\phi}(0)|| e^{-\gamma_\lambda t},$$

with $k_\lambda = 1$ and $\gamma_\lambda \geq \bar{\lambda}_{cd} = \frac{a}{b}\frac{n\mu}{T(1+\frac{a}{b}nT)^2}$. This, together with [162, Lemma 1] or a similar argument as the one in [161, Theorem 4.14], implies that there exists a continuously differentiable, symmetric, positive

definite matrix $\boldsymbol{P}(t)$ *that satisfies the inequalities*

$$\begin{aligned} &0 < \bar{c}_1 \boldsymbol{I} \triangleq \frac{\bar{c}_3}{2n}\boldsymbol{I} \leq \boldsymbol{P}(t) \leq \frac{\bar{c}_4}{2\gamma_\lambda}\boldsymbol{I} \triangleq \bar{c}_2 \boldsymbol{I} \\ &\dot{\boldsymbol{P}} - \frac{a}{b}\bar{\boldsymbol{L}}\boldsymbol{P} - \frac{a}{b}\boldsymbol{P}\boldsymbol{L} \leq -\bar{c}_3 \boldsymbol{I}\,. \end{aligned} \tag{B.33}$$

Next, introducing the vector

$$\boldsymbol{\chi}(t) = b\boldsymbol{\zeta_1}(t) + \boldsymbol{Q}\boldsymbol{\zeta_2}(t)\,,$$

the auxiliary coordination state can be defined as $\bar{\boldsymbol{x}}_{\boldsymbol{TC}} = [\boldsymbol{\chi}^\top, \boldsymbol{\zeta_2}^\top]^\top$*, with dynamics*

$$\begin{cases} \dot{\boldsymbol{\chi}} = -\frac{a}{b}\bar{\boldsymbol{L}}\boldsymbol{\chi} + \frac{a}{b}\boldsymbol{Q}\boldsymbol{L}\boldsymbol{\zeta_2} - \boldsymbol{Q}\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}(\boldsymbol{e}_p) \\ \dot{\boldsymbol{\zeta}}_{\boldsymbol{2}} = -(b\boldsymbol{I} - \frac{a}{b}\boldsymbol{L})\boldsymbol{\zeta_2} - \frac{a}{b}\boldsymbol{L}\boldsymbol{Q}^\top\boldsymbol{\chi} - \bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}(\boldsymbol{e}_p)\,. \end{cases} \tag{B.34}$$

Consider the following Lyapunov candidate function

$$V = \boldsymbol{\chi}^\top \boldsymbol{P}\boldsymbol{\chi} + \frac{\beta_1}{2}||\boldsymbol{\zeta_2}||^2 = \bar{\boldsymbol{x}}_{\boldsymbol{TC}}^\top \boldsymbol{W}\bar{\boldsymbol{x}}_{\boldsymbol{TC}}\,, \tag{B.35}$$

where $\beta_1 > 0$*,* $\boldsymbol{P}$ *was introduced above, and*

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\beta_1}{2}\boldsymbol{I} \end{bmatrix}.$$

Using (B.34)*, ,the time-derivative of* (B.35) *can be computed to yield*

$$\begin{aligned} \dot{V} = \boldsymbol{\chi}^\top \boldsymbol{P}\left(-\frac{a}{b}\bar{\boldsymbol{L}}\boldsymbol{\chi} + \frac{a}{b}\boldsymbol{Q}\boldsymbol{L}\boldsymbol{\zeta_2} - \boldsymbol{Q}\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}\right) + \left(-\frac{a}{b}\boldsymbol{\chi}^\top\bar{\boldsymbol{L}} + \frac{a}{b}\boldsymbol{\zeta_2}^\top\boldsymbol{L}\boldsymbol{Q}^\top - \bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}^\top\boldsymbol{Q}^\top\right)\boldsymbol{P}\boldsymbol{\chi} \\ + \boldsymbol{\chi}^\top\dot{\boldsymbol{P}}\boldsymbol{\chi} + \beta_1\boldsymbol{\zeta_2}^\top\left(-\left(b\boldsymbol{I} - \frac{a}{b}\boldsymbol{L}\right)\boldsymbol{\zeta_2} - \frac{a}{b}\boldsymbol{L}\boldsymbol{Q}^\top\boldsymbol{\chi} - \bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}\right), \end{aligned}$$

which leads to

$$\begin{aligned} \dot{V} \leq &\boldsymbol{\chi}^\top\left(\dot{\boldsymbol{P}} - \frac{a}{b}\boldsymbol{P}\bar{\boldsymbol{L}} - \frac{a}{b}\bar{\boldsymbol{L}}\boldsymbol{P}\right)\boldsymbol{\chi} - \beta_1\boldsymbol{\zeta_2}^\top\left(b\boldsymbol{I} - \frac{a}{b}\boldsymbol{L}\right)\boldsymbol{\zeta_2} \\ &+ 2\frac{a}{b}n||\boldsymbol{P}||||\boldsymbol{\chi}||||\boldsymbol{\zeta_2}|| + 2||\boldsymbol{P}||||\boldsymbol{\chi}||||\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}|| \\ &+ \beta_1\frac{a}{b}n||\boldsymbol{\zeta_2}||||\boldsymbol{\chi}|| + \beta_1||\boldsymbol{\zeta_2}||||\bar{\boldsymbol{\alpha}}_{\boldsymbol{pf}}||\,, \end{aligned}$$

where we used the fact that $||\bar{\boldsymbol{L}}|| \leq n$. *Using* (B.33), *and after straightforward computations, we obtain:*

$$\begin{aligned}\dot{V} \leq & -\bar{c}_3||\boldsymbol{\chi}||^2 - \beta_1\left(b - \frac{a}{b}n\right)||\boldsymbol{\zeta_2}||^2 + \left(2\frac{a}{b}n\bar{c}_2 + \beta_1\frac{a}{b}n\right)||\boldsymbol{\zeta_2}||||\boldsymbol{\chi}||+ \\ & + 2\left(2\bar{c}_2||\boldsymbol{\chi}|| + \beta_1||\boldsymbol{\zeta_2}||\right)\frac{v_{\max}}{v_{\min}+\delta}||\boldsymbol{e}_p||\,,\end{aligned}$$

where $v_{\max} = \max_i\{v_{i,\max}\}\,, \quad v_{\min} = \min_i\{v_{i,\min}\}$.

Finally, using $\bar{c}_2 = \frac{\bar{c}_4}{2\gamma_\lambda}$, *letting* $\bar{c}_4 = \bar{c}_3$, *and choosing* $\delta > v_{\max} - v_{\min}$, *we get*

$$\begin{aligned}\dot{V} \leq & -\bar{c}_3||\boldsymbol{\chi}||^2 - \beta_1\left(b - \frac{a}{b}n\right)||\boldsymbol{\zeta_2}||^2 + \left(\frac{a}{b}\frac{n\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\frac{a}{b}n\right)||\boldsymbol{\zeta_2}||||\boldsymbol{\chi}||+ \\ & + 2\left(\frac{\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\right)||\bar{\boldsymbol{x}}_{\boldsymbol{TC}}||||\boldsymbol{e}_p||\,,\end{aligned}$$

that can be written in matrix form as

$$\dot{V} \leq -\bar{\boldsymbol{x}}_{\boldsymbol{TC}}^\top \boldsymbol{M}\bar{\boldsymbol{x}}_{\boldsymbol{TC}} + 2\left(\frac{\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\right)||\bar{\boldsymbol{x}}_{\boldsymbol{TC}}||||\boldsymbol{e}_p||\,,$$

with

$$\boldsymbol{M} = \begin{bmatrix} \bar{c}_3 & -\left(\frac{a}{b}\frac{n\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\frac{a}{b}n\right) \\ -\left(\frac{a}{b}\frac{n\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\frac{a}{b}n\right) & \beta_1\left(b - \frac{a}{b}n\right)\end{bmatrix}.$$

Now, for any $\delta_{\bar{\lambda}}$ *satisfying* $0 < \delta_{\bar{\lambda}} < 1$, *define* $\lambda_{cd} \triangleq \delta_{\bar{\lambda}}\bar{\lambda}_{cd}$. *Then, by choosing* b *large enough, the following matrix inequality holds:*

$$\boldsymbol{M} - 2\lambda_{cd}\boldsymbol{W} \geq \begin{bmatrix} \bar{c}_3 - \frac{\bar{c}_3\lambda_{cd}}{\bar{\lambda}_{cd}} & -\left(\frac{a}{b}\frac{n\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\frac{a}{b}n\right) \\ -\left(\frac{a}{b}\frac{n\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\frac{a}{b}n\right) & \beta_1\left(b - \frac{a}{b}n\right) - \beta_1\lambda_{cd}\end{bmatrix} \geq 0\,. \tag{B.36}$$

Thus, the derivative of the Lyapunov function is bounded as follows

$$\dot{V} \leq -2\lambda_{cd}V + 2\left(\frac{\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1\right)||\bar{\boldsymbol{x}}_{\boldsymbol{TC}}||||\boldsymbol{e}_p||\,.$$

Using [161, Lemma 4.6], one can conclude that the system (B.34) *is input to state stable, with input* $\boldsymbol{e}_p$, *and the following bound holds:*

$$||\bar{\boldsymbol{x}}_{\boldsymbol{TC}}(t)|| \leq \sqrt{\frac{\max\left(\bar{c}_2, \beta_1/2\right)}{\min\left(\bar{c}_1, \beta_1/2\right)}}||\bar{\boldsymbol{x}}_{\boldsymbol{TC}}(0)||e^{-\lambda_{cd}t} + \sqrt{\frac{\max\left(\bar{c}_2, \beta_1/2\right)}{\min\left(\bar{c}_1, \beta_1/2\right)}}\frac{\frac{\bar{c}_3}{\bar{\lambda}_{cd}} + \beta_1}{\lambda_{cd}\min\left(\bar{c}_1, \beta_1/2\right)}\sup_{t\geq 0}(||\boldsymbol{e}_p(t)||)\,. \tag{B.37}$$

Finally, from the definition

$$\bar{\boldsymbol{x}}_{\boldsymbol{TC}} \triangleq \boldsymbol{S}\boldsymbol{x}_{\boldsymbol{cd}}\,, \quad \boldsymbol{S} = \begin{bmatrix} b\boldsymbol{I}_{\boldsymbol{n-1}} & \boldsymbol{Q} \\ 0 & \boldsymbol{I}_{\boldsymbol{n}} \end{bmatrix},$$

we can conclude that

$$||\boldsymbol{x}_{\boldsymbol{cd}}(t)|| \leq \kappa_1 ||\boldsymbol{x}_{\boldsymbol{cd}}(0)|| e^{-\lambda_{cd} t} + \kappa_2 \sup_{t \geq 0}(||\boldsymbol{e}_p(t)||)\,, \tag{B.38}$$

with

$$\kappa_1 = ||\boldsymbol{S}^{-1}|| \sqrt{\frac{\max(\bar{c}_2, \beta_1/2)}{\min(\bar{c}_1, \beta_1/2)}} ||\boldsymbol{S}||\,, \tag{B.39}$$

and

$$\kappa_2 = ||\boldsymbol{S}^{-1}|| \sqrt{\frac{\max(\bar{c}_2, \beta_1/2)}{\min(\bar{c}_1, \beta_1/2)}} \frac{\frac{\bar{c}_3}{\lambda_{cd}} + \beta_1}{\lambda_{cd} \min(\bar{c}_1, \beta_1/2)}\,. \tag{B.40}$$

As a last step to complete the proof, we need to demonstrate that $\dot{\gamma}_i$ *and* $\ddot{\gamma}_i$ $\forall i \in \{1 \ldots, n\}$ *satisfy the bounds given in* (2.16)*. To this end, notice that*

$$\ddot{\gamma}_i \leq b||\boldsymbol{\zeta_2}|| + an||\boldsymbol{\zeta_1}|| + ||\boldsymbol{e}_p||\,.$$

For simplicity, let $b > an$*. Using the bound in* (B.38)*, and recalling the bound on the path-following error given in Lemma 3, the above inequality reduces to*

$$\ddot{\gamma}_i \leq (2b\kappa_1 + 2b\kappa_2 + 1) \max(||\boldsymbol{x}_{\boldsymbol{cd}}(0)||, \rho)\,.$$

Moreover, using the fact that

$$||\boldsymbol{\zeta_2}(t)|| \leq \kappa_1 ||\boldsymbol{x}_{\boldsymbol{cd}}(0)|| e^{-\lambda_{cd} t} + \kappa_2 \sup_{t \geq 0}(||\boldsymbol{e}_p(t)||)\,,$$

one can show

$$\dot{\gamma}_i \leq 1 + (\kappa_1 + \kappa_2) \max(||\boldsymbol{x}_{\boldsymbol{cd}}(0)||, \rho)\,,$$
$$\dot{\gamma}_i \geq 1 - (\kappa_1 + \kappa_2) \max(||\boldsymbol{x}_{\boldsymbol{cd}}(0)||, \rho)\,.$$

Finally, since by assumption inequality (6.6) *holds, then* (2.16) *is satisfied, and one can show that the bound in* (B.38) *holds* $\forall t \geq 0$*.*

B.4.2 Proof of Corollary 1

Assume that the given VT tracking algorithm satisfies

$$||\boldsymbol{e}_p(t)|| \leq k_{pf}||\boldsymbol{e}_p(0)||e^{-\lambda_{pf}t}\,. \tag{B.41}$$

Now, rewrite inequality (B.38) *as follows:*

$$||\boldsymbol{x_{cd}}(t)|| \leq \kappa_1||\boldsymbol{x_{cd}}(s)||e^{-\lambda_{cd}(t-s)} + \kappa_2 \sup_{s\leq\tau\leq t}(||\boldsymbol{e}_p(\tau)||)\,, \tag{B.42}$$

where $t \geq s \geq 0$. *Apply* (B.42) *with* $s = t/2$ *to obtain*

$$||\boldsymbol{x_{cd}}(t)|| \leq \kappa_1||\boldsymbol{x_{cd}}(t/2)||e^{-\lambda_{cd}(t/2)} + \kappa_2 \sup_{t/2\leq\tau\leq t}(||\boldsymbol{e}_p(\tau)||)\,. \tag{B.43}$$

Apply (B.42) *with* $s = 0$ *and* t *replaced by* $t/2$ *to obtain the estimate of* $\boldsymbol{x_{cd}}(t/2)$ *as*

$$||\boldsymbol{x_{cd}}(t/2)|| \leq \kappa_1||\boldsymbol{x_{cd}}(0)||e^{-\lambda_{cd}(t/2)} + \kappa_2 \sup_{0\leq\tau\leq t/2}(||\boldsymbol{e}_p(\tau)||)\,. \tag{B.44}$$

Combining (B.43) *and* (B.44) *we get*

$$||\boldsymbol{x_{cd}}(t)|| \leq \kappa_1 e^{-\lambda_{cd}t/2}\left(\kappa_1||\boldsymbol{x_{cd}}(0)||e^{-\lambda_{cd}t/2} + \kappa_2 \sup_{0\leq\tau\leq t/2}(||\boldsymbol{e}_p(\tau)||)\right) + \kappa_2 \sup_{t/2\leq\tau\leq t}(||\boldsymbol{e}_p(\tau)||)\,. \tag{B.45}$$

Notice that using (B.41) *we can write*

$$\sup_{0\leq\tau\leq t/2}(||\boldsymbol{e}_p(\tau)||) \leq k_{pf}||\boldsymbol{e}_p(0)||\,,$$

$$\sup_{t/2\leq\tau\leq t}(||\boldsymbol{e}_p(\tau)||) \leq k_{pf}||\boldsymbol{e}_p(0)||e^{-\lambda_{pf}t/2}$$

Therefore, combining (B.45) *with the previous two inequalities, and letting*

$$\bar{\kappa}_1 \triangleq \kappa_1^2\,, \quad \bar{\kappa}_2 \triangleq (1+\kappa_1)\kappa_2 k_{pf}\,, \tag{B.46}$$

we get

$$||\boldsymbol{x_{cd}}(t)|| \leq \bar{\kappa}_1||\boldsymbol{x_{cd}}(0)||e^{-\lambda_{cd}t} + \bar{\kappa}_2||\boldsymbol{e}_p(0)||e^{-\frac{\lambda_{pf}+\lambda_{cd}}{2}t}\,,$$

thus proving Corollary 1.

References

[1] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches*. London. UK: Springer, 2014.

[2] D. Stevenson, M. Wheeler, M. E. Campbell, W. W. Whitacre, R. T. Rysdyk, and R. Wise, "Cooperative tracking flight test," in *AIAA Guidance, Navigation and Control Conference*, Hilton Head, SC, August 2007, AIAA-2007-6756.

[3] Y. Kuwata and J. P. How, "Cooperative distributed robust trajectory optimization using receding horizon MILP," *IEEE Transactions on Control System Technology*, vol. 19, no. 2, pp. 423–431, March 2011.

[4] M. Mesbahi, "On state-dependent dynamic graphs and their controllability properties," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 387–392, March 2005.

[5] M. Cao, D. A. Spielman, and A. S. Morse, "A lower bound on convergence of a distributed network consensus algorithm," in *IEEE Conference on Decision and Control*, Seville, Spain, December 2005, pp. 2356–2361.

[6] Q.-C. Pham and J.-J. Slotine, "Stable concurrent synchronization in dynamic system networks," *Neural Networks*, vol. 20, no. 1, pp. 62–77, January 2007.

[7] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge, UK: Cambridge University Press, 2001.

[8] R. Sepulchre, D. Paley, and N. Leonard, *Collective Motion and Oscillator Synchronization*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin, 2005, vol. 309, pp. 189–206.

[9] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *IEEE Conference on Decision and Control*, Seville, Spain, December 2005, pp. 2996–3000.

[10] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[11] Z. Qu, *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. London, UK: Springer-Verlag London, 2009.

[12] J. Shamma, Ed., *Cooperative Control of Distributed Multi-Agent Systems*. Chichester, UK: John Wiley & Sons, 2007.

[13] Z. Lin, B. A. Francis, and M. Maggiore, "State agreement for continuous-time coupled nonlinear systems," *SIAM Journal on Control and Optimization*, vol. 46, no. 1, pp. 288–307, 2007.

[14] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, April 2006.

[15] F. Liao, R. Teo, J. L. Wang, X. Dong, F. Lin, and K. Peng, "Distributed formation and reconfiguration control of vtol UAVs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 270–277, 2017.

[16] J. H. van Schuppen and T. Villa, *Coordination control of distributed systems*. Springer, 2015.

[17] K. P. Bollino and L. R. Lewis, "Collision-free multi-UAV optimal path planning and cooperative control for tactical applications," in *AIAA Guidance, Navigation and Control Conference*, Honolulu, HI, August 2008, AIAA 2008-7134.

[18] N. Mathew, S. L. Smith, and S. L. Waslander, "Optimal path planning in cooperative heterogeneous multi-robot delivery systems," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 407–423.

[19] A. Tsourdos, B. A. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*. Chichester, UK: John Wiley & Sons, 2010.

[20] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1029–1043, 2014.

[21] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, April 2006.

[22] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, December 2001.

[23] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, September 2004.

[24] R. Olfati Saber, W. B. Dunbar, and R. M. Murray, "Cooperative control of multi-vehicle systems using cost graphs and optimization," in *American Control Conference*, Denver, CO, June 2003, pp. 2217–2222.

[25] R. Olfati Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[26] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205–2213, 2009.

[27] D. M. Stipanović, A. Melikyan, and N. Hovakimyan, "Some sufficient conditions for multi-player pursuit-evasion games with continuous and discrete observations," in *Advances in Dynamic Games and Their Applications*. Springer, 2009, pp. 1–13.

[28] T. Mylvaganam, M. Sassano, and A. Astolfi, "A constructive differential game approach to collision avoidance in multi-agent systems," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 311–316.

[29] J. Hu, M. P. Wellman et al., "Multiagent reinforcement learning: theoretical framework and an algorithm." in *ICML*, vol. 98. Citeseer, 1998, pp. 242–250.

[30] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 157–163.

[31] H. Zhang, J. Zhang, G.-H. Yang, and Y. Luo, "Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 152–163, 2015.

[32] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.

[33] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 2137–2145.

[34] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 464–473.

[35] K. Bollino and L. R. Lewis, "Collision-free multi-UAV optimal path planning and cooperative control for tactical applications," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, p. 7134.

[36] T. W. McLain and R. W. Beard, "Coordination variables, coordination functions, and cooperative timing missions," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 150–161, 2005.

[37] J. Ousingsawat and M. E. Campbell, "Optimal cooperative reconnaissance using multiple vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 122–132, 2007.

[38] E. Scholte and M. E. Campbell, "Robust nonlinear model predictive control with partial state information," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 636–651, 2008.

[39] T. Schouwenaars, J. How, and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 5141.

[40] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative path planning of unmanned aerial vehicles*. John Wiley & Sons, 2010, vol. 32.

[41] Y. Zhang, J. Yang, S. Chen, and J. Chen, "Decentralized cooperative trajectory planning for multiple UAVs in dynamic and uncertain environments," in *2015 IEEE 7th International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE, 2015, pp. 377–382.

[42] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, 2007.

[43] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, 2008.

[44] W. Ren and Y. Cao, *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010.

[45] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[46] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[47] D. J. Stilwell, E. M. Bollt, and D. G. Roberson, "Sufficient conditions for fast switching synchronization in time-varying network topologies," *SIAM Journal on Applied Dynamical Systems*, vol. 5, no. 1, pp. 140–156, 2006.

[48] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[49] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[50] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, July 2006.

[51] I. Kaminer, O. A. Yakimenko, A. M. Pascoal, and R. Ghabcheloo, "Path generation, path following and coordinated control for time-critical missions of multiple UAVs," in *American Control Conference*, Mineapolis, MN, June 2006, pp. 4906–4913.

[52] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Tokyo, Japan: Springer Tokyo, 2010.

[53] T. Shima and S. Rasmussen, Eds., *UAV Cooperative Decision and Control: Challenges and Practical Approaches*, ser. Advances in Design and Control. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2009.

[54] Y. D. Song, Y. Li, and X. H. Liao, "Orthogonal transformation based robust adaptive close formation control of multi-UAVs," in *American Control Conference*, vol. 5, Portland, OR, June 2005, pp. 2983–2988.

[55] D. M. Stipanović, G. İnalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, August 2004.

[56] R. Ritz, M. W. Müller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4972–4978.

[57] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 545–558.

[58] K. P. Valavanis and G. J. Vachtsevanos, Eds., *Handbook of Unmanned Aerial Vehicles*. Dordrecht, The Netherlands: Springer Dordrecht, 2015.

[59] R. Ghabcheloo, A. M. Pascoal, C. Silvestre, and I. Kaminer, "Coordinated path following control of multiple wheeled robots using linearization techniques," *International Journal of Systems Science*, vol. 37, no. 6, pp. 399–414, May 2006.

[60] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 324–333, April 2008.

[61] A. Stubbs, V. Vladimerou, A. T. Fulford, D. King, J. Strick, and G. E. Dullerud, "Multivehicle systems control over networks," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 56–69, June 2006.

[62] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016.

[63] R. Doriya, K. M. Srivastava, and P. Buwe, "An architecture for detection of land mines using swarm robots," in *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2016, pp. 311–323.

[64] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on Control System Technology*, vol. 9, no. 6, pp. 777–790, November 2001.

[65] M. Mesbahi and F. Y. Hadaegh, "Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching," *AIAA Journal of Guidance, Control and Dynamics*, vol. 24, no. 2, pp. 369–377, March–April 2001.

[66] W. Ren, "Formation keeping and attitude alignment for multiple spacecraft through local interactions," *AIAA Journal of Guidance, Control and Dynamics*, vol. 30, no. 2, pp. 633–638, March–April 2007.

[67] A. Matos, A. Martins, A. Dias, B. Ferreira, J. M. Almeida, H. Ferreira, G. Amaral, A. Figueiredo, R. Almeida, and F. Silva, "Multiple robot operations for maritime search and rescue in eurathlon 2015 competition," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–7.

[68] C. Hu, L. Fu, and Y. Yang, "Cooperative navigation and control for surface-underwater autonomous marine vehicles," in *Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017 IEEE 2nd Information*. IEEE, 2017, pp. 589–592.

[69] F. Arrichiello, S. Chiaverini, and T. I. Fossen, "Formation control of marine surface vessels using the null-space-based behavioral control," in *Group Coordination and Cooperative Control*, ser. Lecture Notes in Control and Information Sciences, K. Y. Pettersen, J. T. Gravdahl, and H. Nijmeijer, Eds. Springer-Verlag Berlin Heidelberg, 2006, vol. 336, pp. 1–19.

[70] M. Breivik, V. E. Hovstein, and T. I. Fossen, "Ship formation control: A guided leader-follower approach," in *IFAC World Congress*, Seoul, South Korea, July 2008.

[71] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Trondheim, Norway: Marine Cybernetics, 2002.

[72] I.-A. F. Ihle, "Coordinated control of marine craft," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway, September 2006.

[73] A. Marino, G. Antonelli, A. P. Aguiar, A. Pascoal, and S. Chiaverini, "A decentralized strategy for multirobot sampling/patrolling: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 313–322, 2015.

[74] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "Design and implementation of a range-based formation controller for marine robots," in *ROBOT2013: First Iberian Robotics Conference*. Springer, 2014, pp. 55–67.

[75] T. Petrovic, T. Haus, B. Arbanas, M. Orsag, and S. Bogdan, "Can UAV and UGV be best buddies? towards heterogeneous aerial-ground cooperative robot system for complex aerial manipulation tasks," in *Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on*, vol. 1. IEEE, 2015, pp. 238–245.

[76] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6105–6110.

[77] D. Doroftei, A. Matos, E. Silva, V. Lobo, R. Wagemans, and G. De Cubber, "Operational validation of robots for risky environments," in *8th IARP Workshop on Robotics for Risky Environments*, 2015.

[78] E. Xargay, V. Dobrokhodov, I. Kaminer, A. M. Pascoal, N. Hovakimyan, and C. Cao, "Time-critical cooperative control of multiple autonomous vehicles: Robust distributed strategies for path-following control and time-coordination over dynamic communications networks," *IEEE Control Systems Magazine*, vol. 32, no. 5, pp. 49–73, October 2012.

[79] V. Cichella, I. Kaminer, V. Dobrokhodov, E. Xargay, R. Choe, N. Hovakimyan, A. P. Aguiar, and M. A. Pascoal, "Cooperative Path-Following of Multiple Multirotors over Time-Varying Networks," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 945–957, July 2015.

[80] V. Cichella, R. Choe, S. B. Mehdi, E. Xargay, N. Hovakimyan, V. Dobrokhodov, I. Kaminer, M. A. Pascoal, and A. P. Aguiar, "Safe coordinated maneuvering of teams of multirotor unmanned aerial vehicles: A cooperative control framework for multivehicle, time-critical missions," *IEEE Control Systems Magazine*, vol. 36, no. 4, pp. 59–82, 2016.

[81] I. Kaminer, M. A. Pascoal, E. Xargay, N. Hovakimyan, V. Cichella, and V. Dobrokhodov, *Time-Critical Cooperative Control of Autonomous Air Vehicles*. Elsevier, 2017.

[82] R. Olfati Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[83] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1859–1864.

[84] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[85] Y. Cao, W. Ren, F. Chen, and G. Zong, "Finite-time consensus of multi-agent networks with inherent nonlinear dynamics under an undirected interaction graph," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 4020–4025.

[86] N. Chopra and M. W. Spong, "On exponential synchronization of kuramoto oscillators," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 353–357, 2009.

[87] D. V. Dimarogonas and K. J. Kyriakopoulos, "On the rendezvous problem for multiple nonholonomic agents," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 916–922, 2007.

[88] J. Ng and T. Bräunl, "Performance comparison of bug navigation algorithms," *Journal of Intelligent & Robotic Systems*, vol. 50, no. 1, pp. 73–84, 2007.

[89] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[90] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[91] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[92] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[93] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[94] O. Von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of operations research*, vol. 37, no. 1, pp. 357–373, 1992.

[95] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[96] A. V. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497–528, 2009.

[97] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, pp. 271–306, 2012.

[98] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[99] E. Polak, "Optimization: Algorithms and consistent approximations," 1997, Springer Verlage Publications.

[100] A. Schwartz and E. Polak, "Consistent approximations for optimal control problems based on runge–kutta integration," *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1235–1269, 1996.

[101] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182–197, 2012.

[102] F. Fahroo and I. M. Ross, "On discrete-time optimality conditions for pseudospectral methods," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006, p. 6304.

[103] K. Bollino, L. R. Lewis, P. Sekhavat, and I. M. Ross, "Pseudospectral optimal control: a clear road for autonomous intelligent path planning," in *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*, 2007, p. 2831.

[104] Q. Gong, R. Lewis, and M. Ross, "Pseudospectral motion planning for autonomous vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 1039–1045, 2009.

[105] N. S. Bedrossian, S. Bhatt, W. Kang, and I. M. Ross, "Zero-propellant maneuver guidance," *IEEE Control Systems*, vol. 29, no. 5, 2009.

[106] N. Bedrossian, S. Bhatt, M. Lammers, L. Nguyen, and Y. Zhang, "First ever flight demonstration of zero propellant maneuver (tm) attitute control concept," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, p. 6734.

[107] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5954–5961.

[108] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1917–1922.

[109] G. Farin, *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

[110] R. Farouki and T. Goodman, "On the optimal stability of the Bernstein basis," *Mathematics of Computation of the American Mathematical Society*, vol. 65, no. 216, pp. 1553–1566, 1996.

[111] R. T. Farouki, "The Bernstein polynomial basis: a centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.

[112] J.-W. Chang, Y.-K. Choi, M.-S. Kim, and W. Wang, "Computation of the minimum distance between two Bézier curves/surfaces," *Computers & Graphics*, vol. 35, no. 3, pp. 677–684, 2011.

[113] R. Choe, "Distributed cooperative trajectory generation for multiple autonomous vehicles using pythagorean hodograph Bézier curves," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2017.

[114] R. Choe, J. Puig-Navarro, V. Cichella, E. Xargay, and N. Hovakimyan, "Cooperative trajectory generation using Pythagorean Hodograph Bézier curves," *AIAA Journal of Guidance, Control, and Dynamics*, pp. 1–20, 2016.

[115] L. Yang, D. Song, J. Xiao, J. Han, L. Yang, and Y. Cao, "Generation of dynamically feasible and collision free trajectory by applying six-order Bézier curve and local optimal reshaping," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 643–648.

[116] O. K. Sahingoz, "Generation of Bézier curve-based flyable trajectories for multi-UAV systems with parallel genetic algorithm," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, p. 499, 2014.

[117] B. Park, Y.-C. Lee, and W. Y. Han, "Trajectory generation method using Bézier spiral curves for high-speed on-road autonomous vehicles," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 927–932.

[118] Q. Gong, W. Kang, and I. M. Ross, "A pseudospectral method for the optimal control of constrained feedback linearizable systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 7, pp. 1115–1129, 2006.

[119] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, "Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control," *Computational Optimization and Applications*, vol. 41, no. 3, pp. 307–335, 2008.

[120] Q. Gong, I. M. Ross, and F. Fahroo, "Spectral and pseudospectral optimal control over arbitrary grids," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 759–783, 2016.

[121] P. Encarnação and A. M. Pascoal, "Combined trajectory tracking and path following: An application to the coordinated control of autonomous marine craft," in *IEEE Conference on Decision and Control*, Orlando, FL, December 2001, pp. 964–969.

[122] R. Skjetne, S. Moi, and T. I. Fossen, "Nonlinear formation control of marine craft," in *IEEE Conference on Decision and Control*, vol. 2, Las Vegas, NV, December 2002, pp. 1699–1704.

[123] L. Lapierre, D. Soetanto, and A. M. Pascoal, "Coordinated motion control of marine robots," in *IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September 2003.

[124] R. Skjetne, I.-A. F. Ihle, and T. I. Fossen, "Formation control by synchronizing multiple maneuvering systems," in *IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September 2003.

[125] A. P. Aguiar and A. M. Pascoal, "Coordinated path-following control for nonlinear systems with logic-based communication," in *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007, pp. 1473–1479.

[126] Y. Xu and J. P. Hespanha, "Communication logic design and analysis for networked control systems," in *Current Trends in Nonlinear Systems and Control*, ser. Systems and Control: Foundations & Applications, L. Menini, L. Zaccarian, and C. T. Abdallah, Eds. Cambridge, MA: Birkhäuser Boston, 2006, pp. 495–514.

[127] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation and bandwidth: Reducing communication in distributed control systems using state estimators," *IEEE Transactions on Control System Technology*, vol. 10, no. 4, pp. 503–518, July 2002.

[128] E. Kyrkjebø and K. Y. Pettersen, "Ship replenishment using synchronization control," in *IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September 2003.

[129] E. Kyrkjebø, M. Wondergem, K. Y. Pettersen, and H. Nijmeijer, "Experimental results on synchronization control of ship rendezvous operations," in *IFAC Conference on Control Applications in Marine Systems*, Ancona, Italy, July 2004, pp. 453–458.

[130] D. Soetanto, L. Lapierre, and A. M. Pascoal, "Adaptive, non-singular path-following control of dynamic wheeled robots," in *International Conference on Advanced Robotics*, Coimbra, Portugal, June–July 2003, pp. 1387–1392.

[131] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, "Performance limitations in reference tracking and path following for nonlinear systems," *Automatica*, vol. 44, no. 3, pp. 598–610, March 2008.

[132] R. Ghabcheloo, A. P. Aguiar, A. M. Pascoal, C. Silvestre, I. Kaminer, and J. P. Hespanha, "Coordinated path-following in the presence of communication losses and delays," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 234–265, 2009.

[133] E. Xargay, I. Kaminer, M. A. Pascoal, N. Hovakimyan, V. Dobrokhodov, V. Cichella, A. P. Aguiar, and R. Ghabcheloo, "Time-critical cooperative path following of multiple unmanned aerial vehicles over time-varying networks," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 499–516, 2013.

[134] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International journal of control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[135] M. J. Van Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," *International Journal of Robust and Nonlinear Control*, vol. 8, no. 11, pp. 995–1020, 1998.

[136] V. Cichella, R. Choe, S. B. Mehdi, E. Xargay, N. Hovakimyan, I. Kaminer, and V. Dobrokhodov, "A 3D path-following approach for a multirotor UAV on SO(3)," in *IFAC Research, Education and Development of Unmanned Aerial Systems*, vol. 2, no. 1, 2013, pp. 13–18.

[137] R. Choe, J. Puig-Navarro, V. Cichella, E. Xargay, and N. Hovakimyan, "Trajectory generation using spatial Pythagorean Hodograph Bézier curves," in *AIAA Guidance, Navigation and Control Conference*, Kissimmee, FL, January 2015.

[138] V. Cichella, I. Kaminer, E. Xargay, V. Dobrokhodov, N. Hovakimyan, M. A. Pascoal, and A. P. Aguiar, "A Lyapunov-based approach for time-coordinated 3D path-following of multiple quadrotors," in *2012 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 1776–1781.

[139] V. Cichella, R. Naldi, V. Dobrokhodov, I. Kaminer, and L. Marconi, "On 3D path following control of a ducted-fan UAV on SO(3)," in *2011 50th IEEE Conference on Decision and Control (CDC)*. IEEE, 2011, pp. 3578–3583.

[140] V. Cichella, E. Xargay, V. Dobrokhodov, I. Kaminer, A. M. Pascoal, and N. Hovakimyan, "Geometric 3D path-following control for a fixed-wing UAV on SO(3)," in *AIAA Guidance, Navigation and Control Conference*, Portland, OR, January 2011.

[141] N. Biggs, *Algebraic Graph Theory*. New York, NY: Cambridge University Press, 1993.

[142] R. Diestel, *Graph Theory 3rd. ed.* Springer-Verlag, Heidelberg - New York, 2005.

[143] C. Godsil and G. Royle, *Algebraic graph theory*. Springer New York, 2001, vol. 207.

[144] Q. Gong, W. Kang, N. S. Bedrossian, F. Fahroo, P. Sekhavat, and K. Bollino, "Pseudospectral optimal control for military and industrial applications," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 4128–4142.

[145] T. Lee, M. Leok, and N. H. McClamroch, "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3)," 2011, available online: `arXiv:1003.2005v4`.

[146] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE control systems magazine*, vol. 25, no. 6, pp. 45–55, 2005.

[147] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transactions on control systems technology*, vol. 14, no. 3, pp. 562–571, 2006.

[148] Advanced Controls Research Laboratory at the University of Illinois at Urbana Champaign, "Quadrotor UAVs," http://naira.mechse.illinois.edu/quadrotor-{UAV}s/, Online; accessed April 15, 2018.

[149] Naval Postgraduate School, Monterey, CA, "Center for Autonomous Vehicle Research," http://my.nps.edu/web/cavr, Online; accessed April 15, 2018.

[150] Motion Capture System from Vicon, https://www.vicon.com/, Online; accessed April 15, 2018.

[151] AR.Drone Parrot, http://ardrone.parrot.com, Online; accessed April 15, 2018.

[152] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and autonomous systems*, vol. 42, no. 3, pp. 143–166, 2003.

[153] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[154] P. A. Lasota, T. Fong, J. A. Shah et al., "A survey of methods for safe human-robot interaction," *Foundations and Trends® in Robotics*, vol. 5, no. 4, pp. 261–349, 2017.

[155] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, pp. 143 – 166, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092188900200372X

[156] C. Breazeal, "Emotion and Sociable Humanoid Robots," *International Journal of Human-Computer Studies*, vol. 59, no. 1-2, pp. 119–155, July 2003. [Online]. Available: http://dx.doi.org/10.1016/S1071-5819(03)00018-1

[157] S. N. Bernstein, "Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités (demonstration of a theorem of Weierstrass based on the calculus of probabilities)," *Communications de la Société Mathématique de Kharkov*, vol. 13, no. 1, pp. 1–2, 1912.

[158] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.

[159] M. S. Floater, "On the convergence of derivatives of Bernstein approximation," *Journal of Approximation Theory*, vol. 134, no. 1, pp. 130–135, 2005.

[160] S. D. Conte and C. W. D. Boor, *Elementary numerical analysis: an algorithmic approach*. McGraw-Hill Higher Education, 1980.

[161] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice Hall, 2002.

[162] A. Loría and E. Panteley, "Uniform exponential stability of linear time-varying systems: Revisited," *Systems & Control Letters*, vol. 47, no. 1, pp. 13–24, September 2002.