KTH ROYAL INSTITUTE
OF TECHNOLOGY

Licentiate Thesis in Electrical Engineering

Trajectory Tracking and Prediction-Based Coordination of Underactuated Unmanned Vehicles

DŽENAN LAPANDIĆ

Stockholm, Sweden 2023

Trajectory Tracking and Prediction-Based Coordination of Underactuated Unmanned Vehicles

DŽENAN LAPANDIĆ

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Licentiate of Electrical Engineering on Thursday the 4th May 2023, at 1:00 p.m. in V2, Teknikringen 76, Stockholm.

Licentiate Thesis in Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden 2023

ISBN: 978-91-8040-530-0
TRITA-EECS-AVL-2023:27

Printed by: Universitetsservice US-AB, Sweden 2023

Abstract

In this thesis, we study trajectory tracking and prediction-based control of underactuated unmanned aerial and surface vehicles.

In the first part of the thesis, we examine the trajectory tracking using prescribed performance control (PPC) assuming that the model parameters are unknown. Moreover, due to the underactuation the original PPC is redesigned to accommodate for the specifics of the considered underactuated systems. We prove the stability of the proposed control schemes and support it with numerical simulations on the quadrotor and boat models. Furthermore, we propose enhancements to kinodynamic motion-planning via funnel control (KDF) framework that are based on rapidly-exploring random tree (RRT) algorithm and B-splines to generate the smooth trajectories and track them with PPC. We conducted real-world experiments and tested the advantages of the proposed enhancements to KDF.

The second part of the thesis is devoted to the rendezvous problem of autonomous landing of a quadrotor on a boat based on distributed model predictive control (MPC) algorithms. We propose an algorithm that assumes minimal exchange of information between the agents, which is the rendezvous location, and an update rule to maintain the recursive feasibility of the landing. Moreover, we present a convergence proof without enforcing the terminal set constraints. Finally, we investigated a leader-follower framework and presented an algorithm for multiple follower agents to land autonomously on the landing platform attached to the leader. An agent is equipped with a trajectory predictor to handle the cases of communication loss and avoid the inter-agent collisions. The algorithm is tested in a simulation scenario with the described challenges and the numerical results support the theoretical findings.

Sammanfattning

I denna avhandling studerar vi banspårning och prediktionsbaserad styrning av underaktuerade obemannade luft- och ytfarkoster.

I den första delen av avhandlingen undersöker vi banspårningen med hjälp av föreskriven prestationskontroll (PPC) förutsatt att modellparametrarna är okända. På grund av underaktueringen i systemen vi betraktar har den ursprungliga PPC:n dessutom designats om för att specifikationerna för dessa system. Vi bevisar att de föreslagna regulatorerna stabiliserar systemet och validerar dem med numeriska simuleringar på både quadrotor- och båtmodellen. Dessutom föreslår vi förbättringar av kinodynamisk rörelseplanering via ramverk för trattkontroll (KDF) som är baserade på algoritmen för snabbutforskande slumpmässiga träd (RRT) och B-splines för att generera släta banor och spåra dem med PPC. Vi genomförde fysikaliska experiment och validerade fördelarna med de föreslagna förbättringarna av KDF.

Den andra delen av avhandlingen ägnas åt mötesproblemet med autonom landning av en quadrotor på en båt baserat på algoritmer för distribuerad modellprediktiv styrning (MPC). Vi föreslår en algoritm som förutsätter ett minimalt utbyte av information mellan agenterna, nämligen mötesplatsen, och en uppdateringsregel för att upprätthålla den rekursiva genomförbarheten av landningen. Dessutom presenterar vi ett konvergensbevis utan att upprätthålla begränsningar i slutuppsättningen. Slutligen undersökte vi ett ledare-följare ramverk och presenterade en algoritm där flera följaragenter kan autonomt landa på en plattform som sitter fast i ledaren. En agent är utrustad med en banprediktor för att hantera fall av kommunikationsbortfall samt för att undvika kollision med andra agenter. Algoritmen testas i ett scenario med de beskrivna utmaningarna och de numeriska resultaten överensstämmer med de teoretiska resultaten.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Bo Wahlberg for his guidance and the opportunity to work on such relevant and interesting research topics. I truly appreciate his prompt and insightful feedback and his efforts that enabled to conduct the real-world experiments.

I would also like to give special thanks to my co-supervisor Prof. Dimos Dimarogonas for his support, detailed and mathematically rigor feedback, and time he devoted to our collaboration. I also thank my second co-supervisor Prof. Christos Verginis, who joined this journey one year ago, for the fruitful discussions and very productive meetings we had.

I thank all my colleagues (current and former) at the Division of Decision and Control Systems at KTH for contributing to a nice working atmosphere and our volleyball and other after work activities. Special thank to my colleagues and good friends Rijad Ališić, Mayank Sewlia, Rebecka Winqvist and Pedro Roque for all the fun we have together and our lengthy work and life related discussions. I thank Linnea Persson for the initial collaboration we had on the cooperative MPC, and Pedro Roque and Maria Charitidou for our MPC reading group. I extend my gratitude to my office roommates Javad Parsa, Rijad Ališić, Rebecka Winqvist, Elis Stefansson, Jacob Lindbäck, Rui Oliveira, David Umsonst for a great working atmosphere. Many thanks also to Victor Molnö, Inês Lourenço, Miguel Aguiar, Adrian Wiltz for their contribution to the great atmosphere at the department. I am grateful for the support that WASP Research Arena - Public Safety has provided to my research, and I would like to thank all people involved in WASP PS.

Finally, I would like to send special thanks to my family, my mother Edisa, my father Nermin and my sister Dženita for their unwavering support and encouragement throughout this journey.

Dženan Lapandić
March, 2023.

Contents

Abbreviations

BVLOS	Beyond visul line of sight
DMPC	Distributed Model Predictive Control
DOCP	Distributed Optimal Control Problem
DoF	Degrees of freedom
EKF	Extended Kalman filter
ENU	East-North-Up inertial frame
FHOCP	Finite-horizon Optimal Control Problem
IVP	Initial value problem
KDF	Kinodynamic motion planning with funnel control
LQR	Linear-quadratic regulator
MPC	Model Predictive Control
NED	North-East-Down inertial frame
OCP	Optimal Control Problem
PID	Proportional-integral-differential controller
PPC	Prescribed Performance Control
ROA	Region of attraction
RRT	Rapidly-exploring random tree
SAE	The Society of Automotive Engineers
SAR	Search and rescue
SMC	Sliding-mode control
STL	Signal-temporal logic
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
USV	Unmanned surface vehicle
VLOS	Visual line of sight

List of Figures

Chapter 1

Introduction

The last decade has seen an unprecedented growth in commercially available unmanned aerial vehicles (UAV). By the current regulations, at least in the European Union, their use requires a registration and a remote pilote licence for UAVs weighting less than 25 kg. The operation is restricted within the visual line of sight (VLOS), while the beyond visual line of sight (BVLOS) usage is still legally available only to special categories of users [1].

The commercially available UAVs are remotely operated and can include some basic features that resemble autonomous behaviors such as autonomous take-off and landing, following the remote controller/pilot, collision avoidance, return to the take-off position in case of low battery or performing pre-programmed missions.

Although a clear standardized classification of levels of autonomy for UAVs does not exist yet, based on [2] and analogous to the widely accepted SAE classification for autonomous cars [3], the listed features still satisfy only lower levels or at most the second level of autonomy, while fully autonomous operation or coordination and cooperative missions of several agents that are still subject to development, belong to the higher levels of autonomy. Therefore, there are still open research questions in the control of a single autonomous agent as well as coordination of several of them, which we will refer in the further discussions as a multi-agent system. Developing the control strategies for more autonomous and intelligent decision making will broaden the spectrum of the applicability of these multi-agent systems.

The current applications include video and photography, entertainment, audiovisual inspection, agriculture, delivery, defense, search and rescue, public safety and others. Due to the constrained amount of payload and energy source (battery, fuel) they can carry, the flight time is limited and for long-duration flights refueling or battery exchange is needed. These support tasks can be automated [4] with auxiliary machinery and allow for higher operation autonomy of the unmanned system during a specific task [5]. The continuous operation can also be enabled with several UAVs collaborating together with the support of one or more ground control stations or unmanned surface vehicles (USV) such as a car or a mobile ground robot on the land or a vessel on the water.

Figure 1.1: Examples of cooperative multi-domain operations with manned or unmanned agents. SpaceX rocket landing on the autonomous drone ship (USV), Boeing UAV conducting aerial refueling of an aircraft, Swedish and US marines using a UAV and ship radars for surveillance. Image credit: SpaceX/Boeing/DoD

Some of the notable examples of the cooperative multi-domain (air, land, sea) operations with manned or unmanned agents include SpaceX rocket landings on the ship, Boeing UAV for the aerial refueling, or surveillance for public safety and are depicted on Fig 1.1.

The development of the mentioned technologies has enabled state actors to further strengthen the capabilities in terms of public safety, firefighting and search and rescue (SAR) missions. However, since most of the current unmanned agents are still remotely operated or require active human assistance during the mission, their service drains more resources and is subject to human errors or suboptimal performance. Therefore, increasing the autonomy capabilities of unmanned systems and optimality of their performance has a potential of saving human lives and enhancing the well-being of a society.

In terms of the multi-agent system coordination we can distinguish three approaches based on the location of execution of the control algorithm and shared information i.e. *centralized*, the computation is done at one location and all information is available to the controller, *decentralized*, every agent is executing its part of the control algorithm and controllers are not sharing the information with other agents, and *distributed* control, in which the computation is divided and the information is shared between the agents [6]. The focus of multi-agent system research has been on developing the distributed algorithms to solve general tasks such as *consensus* or *renzdezvous*, in which all agents converge to a certain point in space, and *formation*, in which agents converge to a geometric shape and/or move together in that formation. Most of the distributed approaches considered agents with simple dynamics, with auxiliary low level controllers to ensure the trajectory tracking of the generated trajectories.

In our work we focus on the multi-agent system consisting of a USV, and one or more quadrotor UAVs, that we will refer as UAVs in the further discussions. These agents have nonlinear complex underactuated dynamics with parameters that can change during the execution and are subject to external disturbances such as winds and waves. The motivating problem we consider is the autonomous landing scenario

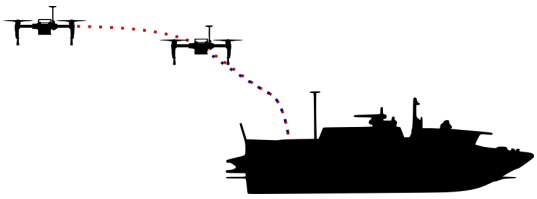

Figure 1.2: The motivating application is a scenario where drones must be able to rendezvous and land on a moving boat.

of a UAV on a moving USV. This particular task allows us to explore almost all aspects of the autonomous cooperative multi-agent behavior, such as trajectory tracking, robustness and disturbance rejection, collision avoidance, and distributed cooperative control. An illustration of the motivating problem is depicted in Fig. 1.2.

Motivated by the above discussion, the approach in the control and coordination of these agents can be twofold. The distributed control algorithms can be developed using the simple dynamics and an auxiliary controller can be added to fulfill the specified performance requirements, and, the distributed control algorithm can use the full model dynamics and prediction-based techniques to enhance the performance. Therefore, this thesis aims to develop trajectory tracking control laws for given trajectories and performance specifications, and prediction-based distributed algorithms for rendezvous.

1.1 Related Work

The related work is divided in two subsections covering: the trajectory tracking and prescribed performance control (PPC), and prediction-based rendezvous using distributed model predictive control.

1.1.1 Trajectory tracking and PPC

Unmanned aerial vehicles (UAV)

UAV systems are highly nonlinear, underactuated and model parameters may vary during the flight. This makes the control design even more demanding, especially in scenarios when UAVs need to meet performance and safety specifications. Such specifications are vital in landing scenarios on other USVs.

There already exists an extensive amount of works in literature concerning stabilization and trajectory tracking control of quadrotors. The early works consider proportional-integral-differential (PID) controllers [7–9], which is designed on simplified model excluding cross-coupling in attitude dynamics and has limited performance in the presence of strong perturbations, and linear-quadratic regulator (LQR) [10–12], whose limitations are due to linearization and requirement of model knowledge. Advanced control methods such as backstepping [13, 14] and

sliding-mode control [15] deliver satisfactory tracking performance but still require model knowledge. Sliding-mode control is known for introducing chattering effect and improvements are made using adjusted boundary-layer sliding control [16, 17]. Adaptive control based on backstepping is derived in [18] and $\mathcal{L}_1$ adaptive control is used for aggressive flight maneuvers in [19]. Flatness-based control is employed in [20, 21], and $\mathcal{H}_\infty$ controller in [22]. With the increase of available computational power in embedded devices, Model predictive control (MPC) became very popular due its ability to handle state and input constraints and optimize the trajectory online [23, 24]. For handling the disturbances, various wind estimation model-based and neural network methods have been applied [25, 26].

However, most of these works focus on model-based approaches or the stability can only be shown around linearized equilibrium points. Furthermore, a significant property that lacks from the related literature on quadrotor control is tracking/stabilization with predefined transient and steady-state specifications, such as overshoot, convergence speed or steady state error. Such specifications can encode time and safety constraints, which are crucial when it comes to physical autonomous systems, and especially UAVs.

Unmanned surface vehicles (USV)

The trajectory tracking problem for surface vehicles or vessels has been extensively studied the last century and a nice overview is given in [27]. We consider an underactuated vessel with 3 degrees of freedom (DoF), i.e. the position in $\mathbb{R}^2$ and orientation, and only two control inputs, for which the trajectory tracking problem cannot be solved using linear systems theory. The classical approach is to steer the vessel along a path using the control of forward velocity and turning [28]. In [29, 30], the authors developed nonlinear Lyapunov-based controllers for path following and trajectory tracking for the considered underactuated vehicles.

Another approach is to introduce a change of frame, and design the nonlinear controller in the new frame. The reference frame is usually chosen as the *Serret-Frenet frame* and using Lyapunov direct method and backstepping the path-following can be ensured [31, 32]. However, this type of controllers assume a constant forward velocity which is a drawback. Another coordinate transformation is utilized in [33] together with backstepping to stabilize the error dynamics and ensure trajectory tracking.

More recently, MPC has been used for the trajectory tracking [34–36] and autonomous docking of USVs [37]. However, most of the controllers in the literature assume the knowledge of dynamical model and parameters. Neural networks have been used in [38] in a rare model-free approach to this problem while prescribed performance has been used for underwater underactuated vehicles in [39]. In the latter, authors consider unicycle-like underwater vehicle that with the neglection of the vertical dimension corresponds to the 3DoF USV. The challenges with this control scheme are control input constraints and obtaining the reference trajectories that the agent is able to track with respect to the given constraints. Therefore, we

focus on developing kinodynamic motion planning and funnel control (KDF) [40] with respect to the input constraints for the considered 3DoF USV model.

1.1.2 Prediction-based rendezvous

Landing scenarios and agents coordination have been explored in [41–44]. The problem of safely landing UAVs on USVs while they are moving at high speeds to ensure agents rendezvous simultaneously has been studied in [45]. The rendezvous problem is challenging due to several reasons, for example, sudden communication losses or strong disturbances acting on the agents can lead to disastrous consequences. Moreover, even the basic tasks to determine if the rendezvous is possible or not and what strategy to employ when the rendezvous location has to be updated can be complex.

MPC has often been used in such applications because of its ability to explicitly include complex system dynamics as well as diverse state and input constraints directly in the computation of the control inputs. A question that has not been directly addressed in previous research is that of efficient communication strategies between the agents. Instead, previous distributed solutions have exchanged all state and trajectory information between the agents at each sample time, [46].

The review [47] gives an overview of several approaches to distributed implementation of model predictive control. Our focus is on dynamically decoupled systems that can be coupled with performance criteria. In [48], the authors assume that each agent knows the system dynamics of all of its neighbors to compute their assumed optimal state trajectories. The stability is established with the requirement that the mismatch from the actual trajectories of the agent's neighbors is small. A similar approach was taken in [49], in which the stability is imposed by requiring that the calculated trajectories of each agent do not deviate from those calculated in the previous time step. Sequential optimization of the local cost functions can, under some assumptions, guarantee stability and convergence to the common cooperative goal, as shown in [50].

However, most of the mentioned research assumes a periodical exchange of information between the agents and recalculation of the control inputs at every sampling time instance. The recalculated control inputs usually do not generate much different state trajectories compared to the ones from the previous time steps, especially if the model is very accurate and disturbances acting on the system are small but are critical for feasibility requirements, see, e.g. [51]. The aperiodic (distributed) MPC can be implemented using the event-triggered or self-triggered strategy [52]. The triggering conditions can be cost-based, then the optimal control problem is recalculated when the cost is not guaranteed to decrease [53]. Moreover, they can be trajectory-based and recalculated when the trajectories deviated significantly compared to the previous ones and the feasibility of the overall problem might be compromised [54], [55]. However, the triggering conditions for nonlinear systems are based on the worst-case trajectory prediction that involves Lipschitz continuity assumption and Lipschitz constant, which for the systems with

fast and agile dynamics, like quadrotors, can lead to very conservative triggering conditions to maintain feasibility and stability.

Another aspect to consider are unpredicted strong disturbances that lead to infeasible landing. The question is then how to proceed and how to formulate the alternative safe strategy. A conventional approach in dealing with the disturbances is to design a strong disturbance rejection controller or, with MPC, to design a tube-based MPC that ensures the trajectories will stay in some safe set given the upper bound of the disturbance [56–59].

In the case of communication loss, we want to have the agent equipped with an alternative strategy to proceed with landing. In [60], the authors propose a vision-based framework with the state estimation for the ground vehicle considered as a moving target. Estimation-based control protocol proposed in [61] uses only local observations of the state of the neighboring agents for rendezvous and flocking control. In another estimation-based protocol, authors in [62] use particle filter to predict aircraft trajectories. Prediction-based navigation in a decentralized event-based scheme is studied in [63].

Moreover, learning-based prediction approaches for multi-agent systems in the recent years became very popular area of research [64–68]. Unfortunately, most of these approaches require extensive amount of data for the training purpose and direct transferability to our particular application is unknown. In [69] authors combine MPC with online learning a stochastic model of a driver or an agent.

1.2 General Problem Formulation

The general objectives of this thesis are to develop trajectory tracking control laws independent of model parameters for underactuated unmanned vehicles and prediction-based rendezvous algorithms with application to the autonomous landing scenario. Underactuated systems are systems that have fewer control inputs than configuration variables. More formally, for the classical mechanical systems obeying to Newtonian mechanics, the underactuated systems can be defined as following.

Definition 1.1. [70] A second-order dynamical control system described by the equations

$$\ddot{\boldsymbol{q}} = f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}, t)$$

where $\boldsymbol{q}$ is the configuration vector of positions, $\dot{\boldsymbol{q}}$ is the vector of velocities, is *fully actuated* in state $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$ and time t if the resulting map f is surjective: for every $\ddot{\boldsymbol{q}}$ there exists a $\boldsymbol{u}$ which produces the desired response. Otherwise it is *underactuated* (at state $\boldsymbol{x}$ at time t).

The dynamics of the considered underactuated unmanned aerial and surface vehicles are heterogeneous and can be modeled separately in state space as a nonlinear system with additive disturbances

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{w}, \tag{1.1}$$

where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ and $\boldsymbol{w} \in \mathcal{W} \subset \mathbb{R}^n$.

In the first part, we address the problem of performance prescribed control for the underactuated systems.

Problem 1.1. Consider the underactuated system (1.1). Given the reference trajectory and prescribed performance functions in time $t \geq 0$, design the control law $\boldsymbol{u}$ that ensures the trajectory tracking and all errors remain inside of the prescribed funnels.

In the second part of the thesis, we focus on rendezvous algorithms with application to the autonomous landing. Agents in the landing scenario can be considered as a multi-agent system. In general, we consider M agents adhering to the nonlinear dynamics without dynamical couplings

$$\dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i, \boldsymbol{u}_i) + \boldsymbol{w}_i, \quad i = 1, ..., M \tag{1.2}$$

Problem 1.2. Consider the system (1.2). Design the distributed control laws $\boldsymbol{u}_i$ that fulfill the rendezvous task, if possible. If this is not possible, find a safe solution online.

The second problem can be further divided into subproblems with respect to the amount of information that can be shared between the agents and the type of failure that can occur during the landing such as communication loss, wind or other type of unpredictable disturbance that prevents the landing to be successful. Therefore, the subproblems include

1. to determine the necessary amount of information to be shared between agents for the successful rendezvous,

2. to detect a failure and utilize an alternative control law online to preserve the safety of the multi-agent system while, potentially, sacrificing the performance.

1.3 Thesis Outline and Contributions

In this section, we provide the thesis outline and state the contributions of the each chapter. In Chapter 2 we introduce the notation, experimental platforms and preliminaries used throughout the thesis. The preliminaries include dynamical control systems, model predictive control, prescribed performance control and distributed control. The thesis is further divided into four main chapters that deal with trajectory tracking for underactuated unmanned aerial and surface vehicles, and rendezvous algorithms based on distributed predictive control.

Chapter 3

We consider the problem of trajectory tracking for underactuated quadrotors with unknown model parameters. The proposed control protocol is based on the prescribed

performance control (PPC) methodology with the necessary design modifications due to the underactuation of the quadrotor system. The stability analysis guarantees that the tracking errors remain inside of designer-specified time-varying functions, achieving prescribed performance independent from the control gains' selection. The results presented in this chapter are based on:

- C1: **Dženan Lapandić**, Christos K. Verginis, Dimos V. Dimarogonas and Bo Wahlberg, "Robust Trajectory Tracking for Underactuated Quadrotors with Prescribed Performance", 61st IEEE Conference on Decision and Control (CDC), Cancún, Mexico, 2022.

Chapter 4

This chapter presents an improved kinodynamic motion planning via funnel control (KDF) with design modifications to accommodate for underactuated unmanned surface vehicle (USV). The considered USV has the input saturation and the proposed control protocol based on prescribed performance guarantees stability. KDF takes into account the desired velocity and acceleration which is determined by the input limitations. The trajectory is generated using KDF-RRT algorithm with the proposed smoothening procedure. This chapter is based on:

- J1: **Dženan Lapandić**, Christos K. Verginis, Dimos V. Dimarogonas and Bo Wahlberg, "Kinodynamic Motion Planning via Funnel Control for Underactuated Unmanned Surface Vehicle", IEEE Transactions on Control Systems Technology, 2023. (To be submitted)

Chapter 5

In this chapter we investigate the rendezvous problem for the autonomous cooperative landing of an unmanned aerial vehicle (UAV) on an unmanned surface vehicle (USV). The agents are heterogeneous, nonlinear and dynamically decoupled but share a common cooperative rendezvous task. The underlying control scheme is based on distributed Model Predictive Control (MPC). The main contribution is a rendezvous algorithm with an online update rule of the rendezvous location. The algorithm only requires the agents to exchange information when they can not guarantee to rendezvous. Hence, the exchange of information occurs aperiodically, which reduces the necessary communication between the agents. We prove that the algorithm guarantees recursive feasibility and asymptotic stability. This chapter is partly based on:

- C2: **Dženan Lapandić**, Linnea Persson and Bo Wahlberg, "Aperiodic Communication for MPC in Autonomous Cooperative Landing" , 7th IFAC Conference on Nonlinear Model Predictive Control (NMPC), Bratislava, Slovakia, 2021.

Chapter 6

We consider the problem of autonomous cooperative landing in a leader-follower framework. We propose a distributed model predictive control (DMPC) based algorithm with a predictor in a scenario of communication loss during the landing of an unmanned aerial vehicle (UAV) on a moving unmanned surface vehicle (USV). The algorithm is executing DMPC with the exchange of predicted trajectories between the agents at a sufficient rate. In the case of loss of communication, and given the sensor setup, agents are predicting the trajectories of other agents based on the output measurements and prior information as the reference input to DMPC. During the landing the follower is tasked with avoidance of obstacles modeled with a nonlinear non-convex function. This chapter is partially based on:

- C3: **Dženan Lapandić**, Dimos V. Dimarogonas and Bo Wahlberg, "Prediction-Based Leader-Follower Rendezvous in Case of Communication Loss", 62nd IEEE Conference on Decision and Control, Singapore, 2023. (Submitted)

Chapter 7 presents the outline of the future work and concludes the thesis.

Chapter 2

Background

In this chapter we introduce the notation and essential background for the contributions presented in the thesis. Furthermore, we present the experimental platforms that are in the focus of our research.

2.1 Notation

The general notation used throughout the thesis is introduced in this section. We may introduce additional notation in the subsequent chapters if necessary.

$\mathbb{R}$ denotes the set of real numbers, while $\mathbb{R}^n$ is the n-dimensional real vector space. $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$ denote the set of non-negative and positive real numbers, respectively. We use $\boldsymbol{P} \succ 0$ to denote that a matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ is positive definite. The notation $\|\boldsymbol{x}\|$ is used as the Euclidean norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ defined by $\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$, where $\boldsymbol{x}^T$ is the transpose of $\boldsymbol{x}$. The one-norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is $\|\boldsymbol{x}\|_1 := \sum_{i=1}^n |x_i|$, where $|x_i|$ is the absolute value of x_i which is the i-th element of the vector $\boldsymbol{x}$. In general, the p-norm, for $p \geq 1$ is defined as $\|\boldsymbol{x}\|_p := (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$. The infinity norm is defined as $\|\boldsymbol{x}\|_\infty := \max(|x_1|, ..., |x_n|)$. We define $\|\boldsymbol{x}\|_{\boldsymbol{P}}$ as a weighted norm of $\boldsymbol{x}$, where $\|\boldsymbol{x}\|_{\boldsymbol{P}} = \sqrt{\boldsymbol{x}^T \boldsymbol{P} \boldsymbol{x}}$.

We define special orthogonal group $SO(n)$ of dimension n as following

$$SO(n) = \{\boldsymbol{R} \in \mathbb{R}^{n \times n} : \boldsymbol{R}\boldsymbol{R}^T = I_n, \det \boldsymbol{R} = 1\}$$

and $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. The notation $\lambda_{\min}(Q)$ and $\lambda_{\max}(Q)$ are used to denote the minimal and maximal eigenvalues of the matrix Q. In some cases we might denote vectors or matrices with not only bold letters but those cases will be noted throughout the thesis.

We denote the system state trajectories with $\boldsymbol{x}(t)$, nominal state trajectories with $\hat{\boldsymbol{x}}(t)$ and optimal state trajectories with $\hat{\boldsymbol{x}}^*(t)$.

2.2 Dynamical Control Systems

In general, we consider the dynamical control systems modeled by ordinary differential equations which together with an initial condition of the system are defined as the initial value problem (IVP)

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.1}$$

where $t \in \mathbb{R}_{\geq 0}$, $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathbb{R}^m$, $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$.

The goal is to design feedback control laws $\boldsymbol{u}(\boldsymbol{x}, t)$ to satisfy some complex requirements. Then the initial value problem reduces to

$$\dot{\boldsymbol{x}} = g(\boldsymbol{x}, t), \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0). \tag{2.2}$$

where $g : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$.

A function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is called Lipschitz continuous in $\mathbb{R}^n$, if there exists a constant $L_f, 0 < L_f < \infty$ such that

$$\|f(\boldsymbol{x}_1, \boldsymbol{u}) - f(\boldsymbol{x}_2, \boldsymbol{u})\| \leq L_f \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n, \forall \boldsymbol{u} \in \mathbb{R}^m.$$

First, we introduce basic concepts of stability and Lyapunov's method.

Definition 2.1. [71, Definition 3.1] Let $f(\boldsymbol{x})$ be a locally Lipschitz function defined over a domain $D \subset \mathbb{R}^n$, which contains the origin, and $f(0) = 0$. The equilibrium point $\boldsymbol{x} = 0$ of $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ is

- stable if for each $\varepsilon > 0$ there is $\delta(\varepsilon) > 0$ such that
$$\|\boldsymbol{x}(0)\| < \delta \implies \|\boldsymbol{x}(t)\| < \varepsilon, \quad \forall t \geq 0$$
- unstable if it is not stable.
- asymptotically stable if it is stable and δ can be chosen such that
$$\|\boldsymbol{x}(0)\| < \delta \implies \lim_{t \to \infty} \boldsymbol{x}(t) = 0.$$

Theorem 2.1. *[71, Theorem 3.3] Let $f(\boldsymbol{x})$ be a locally Lipschitz function defined over a domain $D \subset \mathbb{R}^n$, which contains the origin, and $f(0) = 0$. Let $V(\boldsymbol{x})$ be a continuously differentiable function defined over D such that*

$$V(0) = 0 \ \text{ and } \ V(\boldsymbol{x}) > 0 \ \text{ for all } \ \boldsymbol{x} \in D \ \text{ with } \ \boldsymbol{x} \neq 0 \tag{2.3}$$

$$\dot{V}(\boldsymbol{x}) \leq 0 \ \text{ for all } \ \boldsymbol{x} \in D \tag{2.4}$$

Then, the origin is a stable equilibrium point of $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$. Moreover, if

$$\dot{V}(\boldsymbol{x}) < 0 \ \text{ for all } \ \boldsymbol{x} \in D \ \text{ with } \ x \neq 0 \tag{2.5}$$

then origin is asymptotically stable. Furthermore, if $D = \mathbb{R}^n$, (2.3) and (2.5) hold for all $\boldsymbol{x} \neq 0$, and

$$\|\boldsymbol{x}\| \to \infty \implies V(\boldsymbol{x}) \to \infty$$

then the origin is globally asymptotically stable.

A continuously differentiable function $V(\boldsymbol{x})$ satisfying (2.3) and (2.4) is called a *Lyapunov function*.

We state some preliminaries on dynamical systems from [72] that will be used in Chapters 3 and 4.

Theorem 2.2 (Existence and uniqueness of solutions)**.** *[72, Theorems 2.1.1(i), 2.1.3] Let Ω be an open set in $\mathbb{R}^n \times \mathbb{R}_{\geq 0}$. Consider a function $g : \Omega \to \mathbb{R}^n$ that satisfies the following conditions:*

1. *For every $\boldsymbol{x} \in \mathbb{R}^n$, the function $t \to g(\boldsymbol{x}, t)$ defined on*

$$\Omega_{\boldsymbol{x}} := \{t : (\boldsymbol{x}, t) \in \Omega\} \tag{2.6}$$

is measurable. For every $t \in \mathbb{R}_{\geq 0}$, the function $\boldsymbol{x} \to g(\boldsymbol{x}, t)$ defined on

$$\Omega_t := \{\boldsymbol{x} : (\boldsymbol{x}, t) \in \Omega\} \tag{2.7}$$

is continuous.

2. *For every compact $S \subset \Omega$, there exist constants C_S, L_S such that*

$$\|g(\boldsymbol{x}, t)\| \leq C_S, \quad \|g(\boldsymbol{x}, t) - g(\boldsymbol{y}, t)\| \leq L_S \|\boldsymbol{x} - \boldsymbol{y}\|, \forall (\boldsymbol{x}, t), (\boldsymbol{y}, t) \in S$$

Then the initial value problem (2.2) *for some $(\boldsymbol{x}_0, t_0) \in \Omega$, has a unique and local solution defined in $[t_0, t_{\max})$, with $t_{\max} > t_0$, such that $(\boldsymbol{x}(t), t) \in \Omega, \forall t \in [t_0, t_{\max})$.*

Theorem 2.3 (Maximal solutions)**.** *[72, Theorem 2.1.4] Let the conditions of Theorem 2.2 hold in Ω and let $t_{\max} > t_0$ be the supremum of all times τ such that the initial value problem* (2.2) *has a solution $\boldsymbol{x}(\cdot)$ defined in $[t_0, \tau)$. Then, either $t_{\max} = \infty$ or*

$$\lim_{t \to t_{\max}^-} \left[\|\boldsymbol{x}(t)\| + \frac{1}{d((\boldsymbol{x}(t), t), \partial\Omega)} \right] = \infty, \tag{2.8}$$

where d is the distance of a point $\boldsymbol{p} \in \mathbb{R}^n$ to a set A, defined as $d(\boldsymbol{p}, A) := \inf_{\boldsymbol{y} \in A} \{\|\boldsymbol{p} - \boldsymbol{y}\|\}$.

Furthermore, we introduce concepts of boundedness. Let us consider a system similar to (2.1),

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.9}$$

but where $f : D \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz $\boldsymbol{x}$, and $\boldsymbol{x} \in D$, where $D \subset \mathbb{R}^n$ is a domain that contains the origin.

Definition 2.2. [71, Definition 4.1]

- A scalar continuous function $\alpha(r)$, defined for $r \in [0, a)$, belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. It belongs to class $\mathcal{K}_\infty$ if it is defined for all $r \geq 0$ and $\alpha(r) \to \infty$ as $r \to \infty$.

- A scalar continuous function $\beta(r, s)$, defined for $r \in [0, a)$ and $s \in [0, \infty)$, belongs to class $\mathcal{KL}$ if, for each fixed s, the mapping $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed r, the mapping $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$.

Definition 2.3. [71, Definition 4.3] The solutions of (2.9) are

- uniformly bounded if there exists $c > 0$, independent of t_0, and for every $a \in (0, c)$, there is $\beta > 0$, dependent on a but independent of t_0, such that

$$\|\boldsymbol{x}(t_0)\| \leq a \implies \|\|\boldsymbol{x}(t)\| \leq \beta, \quad \forall t \geq t_0 \tag{2.10}$$

- globally uniformly bounded if (2.10) holds for arbitrarily large a.

- uniformly ultimately bounded with ultimate bound b if there exists a positive constant c, independent of t_0, and for every $a \in (0, c)$, there is $T \geq 0$, dependent on a and b but independent of t_0, such that

$$\|\boldsymbol{x}(t_0)\| \leq a \implies \|\|\boldsymbol{x}(t)\| \leq b, \quad \forall t \geq t_0 + T \tag{2.11}$$

- globally uniformly ultimately bounded if (2.11) holds for arbitrarily large a.

Theorem 2.4. *[71, Theorem 4.4] Let $D \subset \mathbb{R}^n$ be a domain containing $B_\mu = \{\|\boldsymbol{x}\| \leq \mu\}$ and $V(\boldsymbol{x})$ be a continuously differentiable function such that*

$$\alpha_1(\|\boldsymbol{x}\|) \leq V(\boldsymbol{x}) \leq \alpha_2(\|\boldsymbol{x}\|) \tag{2.12}$$

$$\frac{\partial V}{\partial \boldsymbol{x}} f(\boldsymbol{x}, t) \leq -W_3(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in D \text{ with } \|\boldsymbol{x}\| \geq \mu, \ \forall t \geq 0 \tag{2.13}$$

where α_1 and α_2 are class $\mathcal{K}$ functions and $W_3(\boldsymbol{x})$ is a continuous positive definite function. Choose $c > 0$ such that $\Omega_c = \{V(\boldsymbol{x}) \leq c\}$ is compact and contained in D and suppose that $\mu < \alpha_2^{-1}(c)$. Then, Ω_c is positively invariant for the system (2.9) and there exists a class $\mathcal{KL}$ function β such that for every initial state $\boldsymbol{x}(t_0) \in \Omega_c$, the solution of (2.9) satisfies

$$\|\boldsymbol{x}(t)\| \leq \max\{\beta(\|\boldsymbol{x}(t_0)\|, t - t_0), \alpha_1^{-1}(\alpha_2(\mu))\}, \quad t \geq t_0 \tag{2.14}$$

If $D = \mathbb{R}^n$ and $V(x)$ is radially unbounded, then (2.14) holds for any initial state $\boldsymbol{x}(t_0)$, with no restriction on how large μ is.

2.3 Unmanned Vehicles as Dynamical Control Systems

In this section we present the unmanned vehicles of interest and briefly summarize the derivation of their models to the form of (2.1).

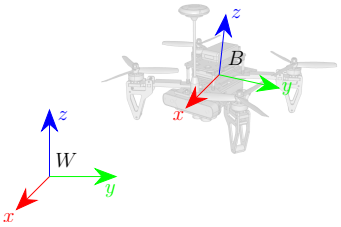

Figure 2.1: The world-fixed inertial frame and body-fixed frame.

2.3.1 Quadrotor

The quadrotor is modeled as a rigid body with 6 degrees of freedom (DoF). Before we start with the derivation of the model equations, it is useful to define the coordinate frames. We define a world-fixed inertial frame W and a body-fixed frame B as depicted on Fig. 2.1.

The rotation matrix $\boldsymbol{R}_{WB} \in SO(3)$ that rotates the coordinates of a point from frame B to frame W is chosen according to the ZYX Tait-Bryan convention, which means that the rotation matrix is a product of three consecutive rotations around the axes Z, Y and X determined by the Euler angles ψ, θ, ϕ known as *yaw-pitch-roll* angles.

$$\boldsymbol{R}_{WB}(\boldsymbol{\eta}) = \boldsymbol{R}_{z,\psi}(\psi)\boldsymbol{R}_{y,\theta}(\theta)\boldsymbol{R}_{x,\phi}(\phi)$$

$$\boldsymbol{R}_{WB}(\boldsymbol{\eta}) = \overset{\text{yaw}}{\begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{pitch}}{\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}} \overset{\text{roll}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}}$$

$$= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

where $\boldsymbol{\eta} = [\phi, \theta, \psi]^T \in \mathbb{T} = (-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\pi, \pi)$ are the Euler angles compactly written as a vector. $\boldsymbol{R}_{z,\psi} : (-\pi, \pi) \to SO(3)$, $\boldsymbol{R}_{y,\theta} : (-\frac{\pi}{2}, \frac{\pi}{2}) \to SO(3)$, $\boldsymbol{R}_{x,\phi} : (-\frac{\pi}{2}, \frac{\pi}{2}) \to SO(3)$ are the respective rotation matrices.

We start the derivation with the Newton-Euler equation for a rigid body [73]

$$\begin{bmatrix} m\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{I}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}^B \\ \dot{\boldsymbol{\omega}}^B \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}^B \times m\boldsymbol{v}^B \\ \boldsymbol{\omega}^B \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega}^B \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^B \\ \boldsymbol{\tau}^B \end{bmatrix} \tag{2.15}$$

given in the body-fixed coordinate frame, where $\boldsymbol{v}^B$ is the velocity in the body frame, $\boldsymbol{\omega}^B$ is the angular velocity in the body frame, $\boldsymbol{F}^B$ is the body force and

$\boldsymbol{\tau}^B$ is the body torque, both applied on the center of mass. m is the mass of the quadrotor, $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{3\times3}$ is the inertia matrix which is constant in the body frame, $\boldsymbol{I}$ and $\boldsymbol{0}$ are identity and zero matrices, respectively, with the appropriate dimensions, in this case in $\mathbb{R}^{3\times3}$.

For the practical reasons, it is more convenient to consider the Newton's law in the world-fixed inertial frame, thus we obtain the following pair of equations of motion

$$\begin{aligned} m\dot{\boldsymbol{v}}^W &= \boldsymbol{F}^W \\ \boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}}^B + \boldsymbol{\omega}^B \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega}^B &= \boldsymbol{\tau}^B \end{aligned} \tag{2.16}$$

where $\boldsymbol{v}^W = \boldsymbol{R}_{WB}\boldsymbol{v}^B$ and $\boldsymbol{F}^W = \boldsymbol{R}_{WB}\boldsymbol{F}^B$ are the velocity and force in the world inertial frame.

The forces acting on the quadrotor are

- the resulting thrust from the propellers $\boldsymbol{T} = \boldsymbol{R}_{WB}\boldsymbol{F}_T$, where $\boldsymbol{F}_T$ is the resulting thrust in the body-fixed frame,
- the gravitational force $\boldsymbol{F}_g = m\boldsymbol{g}$ acting in vertical direction, where $\boldsymbol{g} = [0, 0, -g]^T$, and g is the gravitational acceleration,
- other external forces such as wind, hub and drag forces acting in horizontal direction, ground effects or other unmodeled aerodynamical forces will be considered as disturbances and denoted with $\boldsymbol{F}_d$.

The moments acting on the quadrotor are the resulting torque $\boldsymbol{\tau}$ from propellers and $\boldsymbol{\tau}_d$ that represents other external moments such as rolling, pitching, yawing moments, drag induced moments or other unmodeled aerodynamical moments considered as disturbances. A comprehensive description of aerodynamic forces and moments acting on the quadrotor are given in [74].

Each propeller is generating a thrust T_i, and consequently a torque and a drag moment q_i proportional to the squared angular velocity of the propeller

$$T_i = b\omega_i^2, \qquad q_i = d\omega_i^2. \tag{2.17}$$

The resulting thrust generated by the propellers in the body-fixed frame is

$$\boldsymbol{F}_T = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=0}^{4} T_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=0}^{4} b\omega_i^2 \end{bmatrix} := \begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} \tag{2.18}$$

while the resulting torque is

$$\boldsymbol{\tau} = \begin{bmatrix} l(T_4 - T_2) \\ l(T_1 - T_3) \\ \sum_{i=1}^{4} q_i(-1)^{(i-1)} \end{bmatrix} = \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_1^2 - \omega_3^2) \\ d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix} := \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{2.19}$$

where l is the distance of the propeller from the center of gravity. The control of quadrotor is then achieved by controlling the angular velocities of the each propeller with a low-level controller. However, we assume that we are able to control F_z and $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$ directly.

Now, the system in (2.16) becomes

$$\begin{aligned} \dot{\boldsymbol{v}} &= \frac{1}{m}\left(\boldsymbol{R}_{WB}\boldsymbol{F}_T + \boldsymbol{F}_d\right) - \boldsymbol{g} \\ \dot{\boldsymbol{R}}_{WB} &= \boldsymbol{R}_{WB}\boldsymbol{S}(\boldsymbol{\omega}) \\ \boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega} \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega} + \boldsymbol{\tau} + \boldsymbol{\tau}_d \end{aligned}$$

where we dropped the notation for the coordinate frames, but it must be noted that $\boldsymbol{v}$ is in the world-fixed inertial frame and $\boldsymbol{\omega}$ is in the body-fixed frame and included the relation between the rotation matrix and the body-fixed angular velocities. $\boldsymbol{S}(\boldsymbol{a})$ is a skew-symmetric matrix obtained from a vector $\boldsymbol{a} = [a_1, a_2, a_3]^T$ as

$$\boldsymbol{S}(\boldsymbol{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2.20}$$

Since the matrix $\boldsymbol{R}_{WB}$ depends on the Euler angles $\boldsymbol{\eta}$, it is useful to obtain a direct mapping $\boldsymbol{R}_T : \mathbb{T} \to \mathbb{R}^{3\times 3}$ between the time derivatives of the Euler angles and the angular velocity. In the body frame $\dot{\boldsymbol{\eta}} = \boldsymbol{R}_T^B(\boldsymbol{\eta})\boldsymbol{\omega}$ this transformation is

$$\boldsymbol{R}_T^B(\boldsymbol{\eta}) = \begin{bmatrix} 1 & s_\psi t_\theta & c_\psi t_\theta \\ 0 & c_\psi & -s_\psi \\ 0 & \frac{s_\psi}{c_\theta} & \frac{c_\psi}{c_\theta} \end{bmatrix} \tag{2.21}$$

Note that, in the case one wants to use angular velocities in the world frame then $\dot{\boldsymbol{R}}_{WB} = \boldsymbol{S}(\boldsymbol{\omega}^W)\boldsymbol{R}_{WB}$ and a different transformation matrix is obtained $\dot{\boldsymbol{\eta}} = \boldsymbol{R}_T^W(\boldsymbol{\eta})\boldsymbol{\omega}$,

$$\boldsymbol{R}_T^W(\boldsymbol{\eta}) = \begin{bmatrix} \frac{c_\psi}{c_\theta} & \frac{s_\psi}{c_\theta} & 0 \\ -s_\psi & c_\psi & 0 \\ c_\psi t_\theta & s_\psi t_\theta & 1 \end{bmatrix} \tag{2.22}$$

Unless otherwise specified we will be using $\boldsymbol{R}_T = \boldsymbol{R}_T^B$ as we are considering angular velocities in the body frame. Note that $\boldsymbol{R}_T$ is well-defined for $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, which will be an assumption that we will use throughout the thesis. The simplified version of the model can be used assuming that $\boldsymbol{R}_T(\boldsymbol{\eta}) = I_3$ which is a reasonable assumption if the perturbations from hover flight are small. Here we adopt the shorthand notation for trigonometric functions, i.e., $s_\psi = \sin\psi, c_\psi = \cos\psi, t_\theta = \tan\theta$.

Finally, we can state the quadrotor UAV model:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{2.23a}$$

$$\dot{\boldsymbol{v}} = \frac{1}{m}\left(\boldsymbol{R}_{WB}(\boldsymbol{\eta})\boldsymbol{F}_T + \boldsymbol{F}_d\right) - \boldsymbol{g}, \tag{2.23b}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}_T(\boldsymbol{\eta})\boldsymbol{\omega}, \tag{2.23c}$$

$$\boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega} + \boldsymbol{\tau} + \boldsymbol{\tau}_d \tag{2.23d}$$

where $\boldsymbol{p} \in \mathbb{R}^3$ is the position in the world frame, and as stated before $\boldsymbol{v} \in \mathbb{R}^3$ is the velocity in the world frame, $\boldsymbol{\eta} = [\phi, \theta, \psi]^T \in \mathbb{T}$ is the vector of Euler angles, and $\boldsymbol{\omega} = [p, q, r]^T \in \mathbb{R}^3$ is the angular velocity, expressed in the body frame.

In terms of the state-space representation the state vector is

$$\boldsymbol{x} = [\boldsymbol{p}^T, \boldsymbol{v}^T, \boldsymbol{\eta}^T, \boldsymbol{\omega}^T]^T$$

and belongs to a subset of $\mathbb{R}^{12}$ due to the Euler angle representation, and the input vector is

$$\boldsymbol{u} = [F_z, \boldsymbol{\tau}^T]^T \in \mathbb{R}^4.$$

The underactuation stems from the fact that the quadrotor UAV has higher degrees of freedom (DoF) than control inputs The given model is a 6-DoF model and has only 4 control inputs.

2.3.2 Piraya boat

We consider the model of a boat USV with one rotating thruster at the rear as depicted on Fig. 4.1. The derivation of the model starts from the rigid-body dynamics

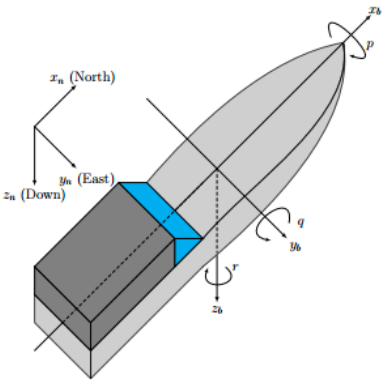

Figure 2.2: USV boat model with world-fixed inertial and body-fixed frame orientations. The illustration is courtesy of [75].

as given in Eq. (2.15). We neglect the motion in the vertical direction, thus the velocity in the body frame is now two-dimensional (forward and lateral velocities). Also, we neglect the rolling and pitching effects on the boat, and the angular velocity is only considered around the z-axis. Therefore, the model is the reduced 3-DoF surface boat model which, with some abuse of the notation used in the previous section but to be consistent with the boat community literature, can be written as in [27, 75] for the motion in surge, sway and yaw.

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{b}} = \boldsymbol{R}_{z,\psi}(\psi_b)\boldsymbol{\nu}_{\boldsymbol{b}} \tag{2.24a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_{\boldsymbol{b}} + \boldsymbol{C}(\boldsymbol{\nu}_{\boldsymbol{b}})\boldsymbol{\nu}_{\boldsymbol{b}} + \boldsymbol{D}(\boldsymbol{\nu}_{\boldsymbol{b}})\boldsymbol{\nu}_{\boldsymbol{b}} = \boldsymbol{\tau}_{act} + \boldsymbol{\tau}_d \tag{2.24b}$$

where $\boldsymbol{\eta}_{\boldsymbol{b}} = [x, y, \psi]$, $(x, y) \in \mathbb{R}^2$ is the position in the world frame, $\psi \in [0, 2\pi)$ is the rotation of the boat in the world frame, $\boldsymbol{\nu}_{\boldsymbol{b}} = [u, v, r]$ is the velocity in the body frame, where u, v, r are forward velocity (surge), lateral velocity (sway) and angular velocity in yaw, respectively. $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu}_{\boldsymbol{b}})$ denotes the Coriolis and centripetal effects, $\boldsymbol{D}(\boldsymbol{\nu}_{\boldsymbol{b}})$ is the drag matrix. We will assume that the drag induced by the relative velocity of ship and surrounding water is considered as a disturbance. This leads to a linear damping matrix that is valid for low relative velocities, while at higher velocities nonlinear term dominates and big discrepancies in the model are expected. After the observed simplifications the matrices reduce to

$$\boldsymbol{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad \boldsymbol{C}(\boldsymbol{\nu}_{\boldsymbol{b}}) = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{bmatrix} \tag{2.25}$$

$$\boldsymbol{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & 0 \\ 0 & 0 & -N_r \end{bmatrix} \tag{2.26}$$

$\boldsymbol{\tau}_d$ are unknown disturbances, and $\boldsymbol{\tau}_{act} = [X, Y, N]^T$ is the control torque. In general, the control torque for N thrusters that generate a force $F_{T,i}$ mounted at a location that is $\Delta_{x,i}, \Delta_{y,i}$ displaced from the center of mass with a rotation $\alpha_{r,i}$ is defined as

$$\boldsymbol{\tau}_{act} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} F_{T,i} \cos(\alpha_{r,i}) \\ \sum_{i=1}^{N} F_{T,i} \sin(\alpha_{r,i}) \\ \sum_{i=1}^{N} F_{T,i} \left(\Delta_{x,i} \sin(\alpha_{r,i}) - \Delta_{y,i} \cos(\alpha_{r,i})\right) \end{bmatrix}. \tag{2.27}$$

For the configuration with a thruster located at the rear $\Delta_{y,1} = 0$ and

$$\boldsymbol{\tau}_{act} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} F_T \cos(\alpha_r) \\ F_T \sin(\alpha_r) \\ \Delta_x F_T \sin(\alpha_r) \end{bmatrix}. \tag{2.28}$$

2.4 Feedback Control Laws

In this section, we confer the basic principles of the feedback control laws that will be used, with modifications, throughout the thesis.

2.4.1 Performance Prescribed Control

We present the basic framework of prescribed performance control (PPC) which is originally introduced in [76, 77]. It belongs to the class of robust adaptive nonlinear controllers. The main idea is to ensure the convergence of the tracking errors $e(t)$ within a predefined set at a specified rate. This is accomplished by enforcing the error to stay within a region bounded by a certain smooth and bounded function of time, i.e.,

$$-\rho(t) < e(t) < \rho(t), \quad \forall t \geq 0, \tag{2.29}$$

where $\rho(t)$ is the performance function that can be defined, similar to [76], as following.

Definition 2.4 (Performance function)**.** A performance function $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ is a smooth, positive, and non-increasing function given by

$$\rho(t) = (\rho_0 - \rho_\infty)e^{-lt} + \rho_\infty, \quad \forall t \geq 0, \tag{2.30}$$

where $\rho_0, \rho_\infty \in \mathbb{R}_{>0}$ with $\rho_0 \geq \rho_\infty$ and $l \in \mathbb{R}_{\geq 0}$.

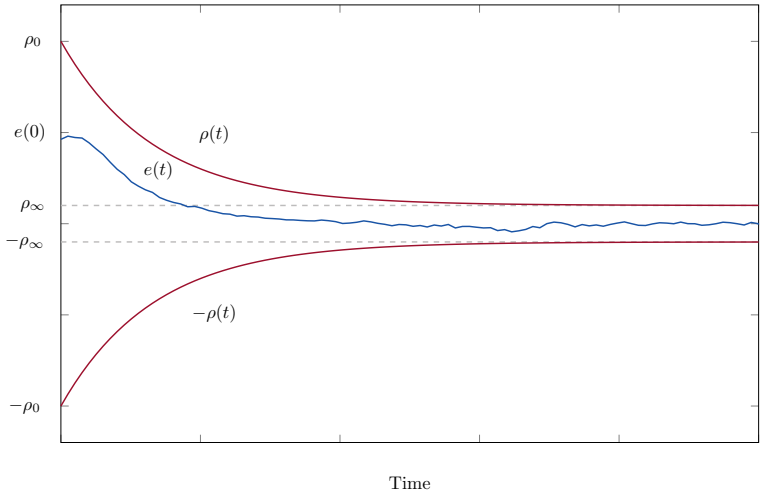

Figure 2.3: The error evolves inside of the prescribed performance funnel.

Practically, ρ_0 is selected such that the error starts inside of the prescribed funnel, i.e. $\rho_0 > |e(0)|$, $\rho_\infty := \lim_{t\to\infty} \rho(t) > 0$ represents the upper bound on the steady-state error and l is the lower bound on the convergence rate of the error.

Therefore, appropriate choice of the discussed parameters determines the transient and steady-state performance of the error $e(t)$, as depicted on Fig. 2.3. Furthermore, let the normalized errors $\xi(t)$ be defined as

$$\xi(t) = \rho(t)^{-1} e(t).$$

The important point in designing PPC is a transformation of the normalized error $\xi(t)$ with a strictly increasing, bijective function $\mathbb{T}$.

Definition 2.5 (Transformation function)**.** A transformation function $\mathbb{T} : (-1, 1) \to \mathbb{R}$ is a smooth and strictly increasing, hence a bijective function. In particular, let

$$\mathbb{T}(\xi(t)) = \operatorname{atanh}(\xi(t)) = \frac{1}{2} \ln \frac{1 + \xi(t)}{1 - \xi(t)}. \tag{2.31}$$

The derivative of this transformation is

$$r(\xi) := \frac{\mathbb{T}(\xi)}{\xi} = \frac{1}{1 - \xi^2}. \tag{2.32}$$

For a vector of normalized errors $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_n]^T \in \mathbb{R}^n$, we define

$$\mathbb{T}(\boldsymbol{\xi}) = \frac{1}{2} \begin{bmatrix} \ln \frac{1+\xi_1}{1-\xi_1} & \ldots & \ln \frac{1+\xi_n}{1-\xi_n} \end{bmatrix}^T \tag{2.33}$$

and the derivative in the matrix form

$$\frac{\mathbb{T}(\boldsymbol{\xi})}{\boldsymbol{\xi}} = \operatorname{diag} \left\{ \left[\frac{1}{1 - \xi_i^2} \right]_{i \in \{1, \ldots, n\}} \right\} \tag{2.34}$$

In order to design a continuous feedback control law and ensure the normalized errors stay within the interval $(-1, 1)$, the transformed errors $\varepsilon(t) = \mathbb{T}(\xi(t))$ need to be bounded. Therefore, if $\varepsilon(t)$ is bounded for all $t \geq 0$, then $e(t)$ satisfies (2.29). Note that the direct application of PPC methodology is not possible for underactuated systems, a topic that we will further investigate in the coming chapters.

2.4.2 Model Predictive Control

Model predictive control (MPC) is a control strategy based on iterative computation of a control input by minimizing a given cost function with respect to all possible control inputs and predicted state trajectories satisfying enforced input and state constraints. At every time step, the controller is solving the finite-horizon optimal control problem (FHOCP) based on the current initial condition. The problem is considering the system during a predefined prediction horizon. The first value of the resulting optimal control input is implemented in a zero-order-hold manner until the next control input is computed. The time between two control inputs is defined as

the control horizon, and the system is evolving under an "*open-loop*" control input. By executing the described procedure iteratively the feedback loop is closed [78].

Consider the initial value problem given by general nonlinear system dynamics with additive disturbances, and the initial condition $\boldsymbol{x}_0$ at time $t_0 = 0$,

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2.35}$$

where $\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, $\boldsymbol{u}(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ and $\boldsymbol{w}(t) \in \mathcal{W} \subset \mathbb{R}^n$, for all $t \geq 0$.

We introduce the MPC as a stabilization problem around a steady-state pair $(\boldsymbol{x}_s, \boldsymbol{u}_s) \in \mathcal{X} \times \mathcal{U}$, that we consider, without loss of generality, to be the origin $(\boldsymbol{x}_s, \boldsymbol{u}_s) = (\mathbf{0}, \mathbf{0})$. The following standard MPC assumptions as in [51] are needed.

Assumption 2.1. It is assumed that

(i) the function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is twice continuously differentiable and $f(\mathbf{0}, \mathbf{0}) = \mathbf{0}$;

(ii) $\mathcal{U} \subseteq \mathbb{R}^m$ is compact, convex and $\mathbf{0} \in \mathbb{R}^m$ is contained in $\mathcal{U}$;

(iii) the system in (2.35) has a unique solution for any initial condition $\boldsymbol{x}_0 \in \mathbb{R}^n$, any piecewise continuous and right-continuous control $\boldsymbol{u} : [t_0, \infty) \to \mathcal{U}$, and any disturbance $\boldsymbol{w} : [t_0, \infty) \to \mathcal{W}$;

(iv) for the linearized system around the origin without disturbances, i.e., $\dot{\boldsymbol{x}} = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)$, where $A = \frac{\partial f}{\partial \boldsymbol{x}}(\mathbf{0}, \mathbf{0})$ and $B = \frac{\partial f}{\partial \boldsymbol{u}}(\mathbf{0}, \mathbf{0})$, the pair (A, B) is stabilizable;

(v) for the linearized dynamics around the origin, there exists a matrix K such that $A_k = A + BK$ is a stable Hurwitz matrix.

The cost function $J(\hat{\boldsymbol{x}}(t_k), \boldsymbol{u}(t_k))$, at the time step t_k, is defined such that it penalizes the deviations of the predicted nominal system trajectories $\hat{\boldsymbol{x}}(\tau; t_k)$ and control input $\boldsymbol{u}(\tau)$ from the origin, for the duration of the prediction horizon $\tau \in [t_k, t_k + T]$, $T > 0$, as

$$J(\hat{\boldsymbol{x}}(\tau; t_k), \boldsymbol{u}(\tau; t_k)) = V(\hat{\boldsymbol{x}}(t_k + T; t_k)) + \int_{t_k}^{t_k+T} l(\hat{\boldsymbol{x}}(\tau; t_k), \boldsymbol{u}(\tau)) d\tau, \tag{2.36}$$

where $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the terminal cost and $l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is the stage cost function.

Now, we formulate the MPC as the finite-horizon optimal control problem that is solved at every time step $t_k = t_0 + k\Delta$, where $k \in \mathbb{N}$ and $\Delta > 0$ is the sampling rate.

Problem 2.1. At time t_k, given the initial state $\boldsymbol{x}(t_k)$, the finite-horizon optimal control problem is to minimize

$$\min_{\boldsymbol{u}} J(\hat{\boldsymbol{x}}(\tau; t_k), \boldsymbol{u}(\tau; t_k)) \tag{2.37a}$$

subject to

$$\dot{\hat{\boldsymbol{x}}}(\tau;t_k) = f(\hat{\boldsymbol{x}}(\tau;t_k), \boldsymbol{u}(\tau;t_k)),\ \ \tau \in [t_k, t_k+T]\,, \tag{2.37b}$$
$$\hat{x}(\tau;t_k) \in \mathcal{X}, \tag{2.37c}$$
$$u(\tau;t_k) \in \mathcal{U}, \tag{2.37d}$$
$$\hat{x}(t_k+T;t_k) \in \mathcal{X}_f. \tag{2.37e}$$

The result is the optimal control input $\boldsymbol{u}^*(\tau;t_k)$, valid for $\tau \in [t_k, t_k+T]$. The first control input is applied $\boldsymbol{u}^*(t_k;t_k)$ for $\tau \in [t_k, t_k+\Delta]$ and the procedure is repeated. As an optimization-based method, MPC takes into account the state and input constraint satisfaction during the prediction horizon (2.37c)-(2.37d). The results on feasibility and stability of MPC are based on recursive properties of the method as well as satisfaction of the terminal constraints (2.37e) and existence a local control law that ensure that the cost is not increasing when the state is in the terminal constraint set, although this local control law is not needed to be implemented. However, there are results to avoid usage of the terminal constraints [79]. Moreover, MPC also posses inherent robustness properties [80, 81].

2.5 Distributed Control

The distributed control of multi-agent systems is a very broad research area. There are two important goal-oriented aspects of the distributed control of multi-agent systems that are consensus (*rendezvous*) and formation control. In this thesis, we consider the rendezvous problem and formulate it for two or more agents. The formulation for two agents is based on the leader-follower framework in which one of the agent is acting as the leader and may have an additional goal in the problem, while the other agent is simply following the leader. These formulations will be additionally specified in the chapters considering the distributed control problems.

The distributed control in the context of this thesis will be used with prediction-based methods such as model predictive control. We consider dynamically decoupled agents in problems that are designed as cooperative problems, though some aspects of non-cooperative control are also examined. The distributed model predictive control has been widely studied as outlined in Section 1.1.2.

Chapter 3

Trajectory Tracking for Unmanned Aerial Vehicles with Prescribed Performance

In this chapter we investigate trajectory tracking with prescribed performance control (PPC) methodology for a quadrotor unmanned aerial vehicle (UAV). UAV systems are highly nonlinear, underactuated and model parameters may vary during the flight as outlined in Subsection 2.3.1. This makes the control design even more demanding, especially in scenarios when UAVs need to meet performance and safety specifications. Such specifications are vital in landing scenarios on other unmanned surface vehicles (USV).

The prescribed performance control traditionally deals with model uncertainties and transient- and steady-state constraints and it is introduced in Subsection 2.4.1. However, the original PPC methodology cannot be directly applied to the quadrotor systems because they belong to the class of underactuated systems. Therefore, it is necessary to introduce design modifications to stabilize the considered system with prescribed performance.

We present a modified Prescribed Performance Control protocol developed in [82] to solve the trajectory-tracking control problem for quadrotors with prescribed performance. Our main contribution is in the extension of the original PPC algorithm to account for the underactuated quadrotor system. At the same time, the proposed control protocol does not use any information on the model parameters and is robust to unknown exogenous disturbances without employing approximation or observer-based schemes. Similarly to the original PPC methodology, the tracking errors evolve within predefined user-specified functions of time, achieving prescribed transient and steady-state performance that is independent from the selection of the control gains. Furthermore, we present the stability theorem and the proof. Finally, simulation results for two scenarios verify the theoretical results.

It should be noted that PPC has been recently used to control quadrotors. In [83, 84] authors use PPC for the attitude subsystem, thus avoiding the underactuated part of the system. Other works [85–88] focus on the complete system but use neural network approximations, partial knowledge of dynamic parameters, observers for

disturbance estimates and exploit these information in the controller. On the contrary, the proposed controller does not use any information on the dynamic parameters or external disturbances. In [89], the authors proposed a similar control design using PPC for an underactuated 3-DOF helicopter. Such a system is, however, significantly different than the one studied in our work and hence the respective control design is not applicable. A comprehensive literature review is available in Subsection 1.1.1.

3.1 Problem Statement

In this chapter, we consider the general trajectory tracking control problem as stated in Problem 1.1, for quadrotor systems. Before we state and formalize the specifics of the problem, let us first introduce the dynamical system of interest for this chapter.

The quadrotor UAV model (2.23) is described in Subsection 2.3.1 but here we consider angular velocities $\boldsymbol{\omega}$ in the inertial world frame. Therefore the matrix $\boldsymbol{R}_T(\boldsymbol{\eta}) = \boldsymbol{R}_T^W(\boldsymbol{\eta})$ as in (2.22). Thus the model is

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{3.1a}$$

$$\dot{\boldsymbol{v}} = \frac{1}{m}\left(\boldsymbol{R}_{WB}(\boldsymbol{\eta})\boldsymbol{F}_T + \boldsymbol{F}_d(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)\right) - \boldsymbol{g}, \tag{3.1b}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}_T(\boldsymbol{\eta})\boldsymbol{\omega}, \tag{3.1c}$$

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\eta})\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{\mathcal{I}}(\boldsymbol{\eta})\boldsymbol{\omega} + \boldsymbol{\tau} + \boldsymbol{\tau}_d(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) \tag{3.1d}$$

where $\boldsymbol{x} = [\boldsymbol{p}^T, \boldsymbol{v}^T, \boldsymbol{\eta}^T, \boldsymbol{\omega}^T]^T$, $\boldsymbol{p} \in \mathbb{R}^3$ is the position in the inertial frame, $\boldsymbol{v} \in \mathbb{R}^3$ is the linear velocity, $\boldsymbol{\eta} = [\phi, \theta, \psi]^T \in \mathbb{T} = (-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\pi, \pi)$ is the vector of Euler angles representing the attitude (roll, pitch, yaw angles), and $\boldsymbol{\omega} = [\omega_\phi, \omega_\theta, \omega_\psi]^T$ is the angular velocity, expressed in the inertial frame; $\boldsymbol{F}_T = [0, 0, F_z]^T$ is the controlled thrust, and $\boldsymbol{\tau}$ is the inertial-frame controlled torque; The functions $\boldsymbol{F}_d(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) = [\boldsymbol{F}_{d,xy}^T, \boldsymbol{F}_{d,z}]^T$, $\boldsymbol{\tau}_d := \boldsymbol{\tau}_d(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)$ represent unmodelled aerodynamic forces and moments like drag, hub forces or ground and gyroscopic effects, and exogenous disturbance. The two functions are continuous in $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$, uniformly bounded in t. The term $\boldsymbol{g} = [0, 0, g]^T \in \mathbb{R}^3$ corresponds to the constant gravity vector. Finally, $m \in \mathbb{R}$ and $\boldsymbol{\mathcal{I}} : \mathbb{T} \to \mathbb{R}^{3\times3}$ are the mass and positive definite inertia matrix of the UAV, also considered *unknown*.

We consider the tracking control problem of given time-varying reference trajectories $\boldsymbol{p}_r = [p_{x,r}, p_{y,r}, p_{z,r}]^T : [0, \infty) \to \mathbb{R}^3$, $\psi_r : [0, \infty) \to \mathbb{R}$ for the position and yaw angles with prescribed performance. $\boldsymbol{p}_r$ and ψ_r are assumed to be smooth functions of time with bounded first and second derivatives. We adapt the PPC methodology to achieve trajectory tracking with prescribed performance for the position and yaw-angle variables.

More specifically, the control objective is to guarantee that the errors

$$\boldsymbol{e}_p = \begin{bmatrix} e_{p_x} \\ e_{p_y} \\ e_{p_z} \end{bmatrix} = \boldsymbol{p} - \boldsymbol{p}_r \tag{3.2a}$$

$$e_\psi = \psi - \psi_r \tag{3.2b}$$

evolve strictly within a funnel dictated by the corresponding exponential performance functions $\rho_{p_x}(t)$, $\rho_{p_y}(t)$, $\rho_{p_z}(t)$, $\rho_\psi(t)$, which is formulated as

$$|e_{p_i}(t)| < \rho_{p_i}(t),\ i \in \{x, y, z\} \tag{3.3a}$$

$$|e_\psi(t)| < \rho_\psi(t), \tag{3.3b}$$

for all $t \geq 0$, given the initial funnel compliance

$$|e_{p_i}(0)| < \rho_{p_i}(0),\ i \in \{x, y, z\}$$
$$|e_\psi(0)| < \rho_\psi(0).$$

The adopted exponentially-decaying performance functions are

$$\rho_{p_i}(t) = (\rho_{p_i,0} - \rho_{p_i,\infty})e^{-l_{p_i}t} + \rho_{p_i,\infty},\ i \in \{x, y, z\}$$
$$\rho_\psi(t) = (\rho_{\psi,0} - \rho_{\psi,\infty})e^{-l_\psi t} + \rho_{\psi,\infty}.$$

3.2 Reformulating the Model

Since the UAV system is underactuated, the idea is to take advantage of the specific control inputs to control the vertical velocity v_z and angular velocity $\boldsymbol{\omega}$, but also introduce virtual control on the horizontal velocities $\boldsymbol{v}_{xy} = [v_x, v_y]^T$ such that the given reference is tracked. Let us first rewrite the system dynamics in a control suitable form.
Let

$$\boldsymbol{R}_{z,\psi} = \begin{bmatrix} \boldsymbol{R}_\psi & 0 \\ 0 & 1 \end{bmatrix}$$

and the factorization

$$\boldsymbol{R}_{WB}\boldsymbol{F}_T = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi}\begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_\psi \boldsymbol{T}_{\phi\theta} F_z \\ c_\theta c_\phi F_z \end{bmatrix}$$

where

$$\boldsymbol{T}_{\phi\theta} = \begin{bmatrix} s_\theta c_\phi \\ -s_\phi \end{bmatrix}, \tag{3.4}$$

and $\boldsymbol{R}_\psi \in SO(2)$. Then, the velocity dynamics $\boldsymbol{v} = [\boldsymbol{v}_{xy}^T, v_z]^T$ can be written as

$$\dot{\boldsymbol{v}}_{xy} = \frac{1}{m}\left(\boldsymbol{R}_\psi \boldsymbol{T}_{\phi\theta} F_z + \boldsymbol{F}_{d,xy}\right), \tag{3.5a}$$

$$\dot{v}_z = \frac{1}{m}\left(c_\theta c_\phi F_z + F_{d,z}\right) - g. \tag{3.5b}$$

Furthermore, let us define the matrices

$$\boldsymbol{J}_{\phi\theta} = \begin{bmatrix} -s_\theta s_\phi & c_\theta c_\phi \\ -c_\phi & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{R}_{\phi\theta} = \begin{bmatrix} \frac{c_\psi}{c_\theta} & \frac{s_\psi}{c_\theta} \\ -s_\psi & c_\psi \end{bmatrix},$$

that satisfy

$$\dot{\boldsymbol{T}}_{\phi\theta} = \boldsymbol{J}_{\phi\theta} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} \quad \text{and} \quad \boldsymbol{R}_T = \begin{bmatrix} \boldsymbol{R}_{\phi\theta} & \boldsymbol{0}_2 \\ c_\psi t_\theta & s_\psi t_\theta \quad 1 \end{bmatrix}$$

which will be used in the sequel. Note that $\boldsymbol{R}_{\phi\theta}$ is well-defined and invertible for $|\theta| < \frac{\pi}{2}$, while $\boldsymbol{J}_{\phi\theta}$ is invertible for $|\phi| < \frac{\pi}{2}$ and $|\theta| < \frac{\pi}{2}$. This is a commonly used assumption for UAV systems [13, 18, 22] and we adopt it in this paper:

Assumption 3.1. The roll and pitch angles satisfy $|\phi(t)| \leq \bar{\pi}$, $|\theta(t)| \leq \bar{\pi}$, for all $t \geq 0$ and some $\bar{\pi} < \frac{\pi}{2}$.

3.3 Control Design

We describe now the proposed control-design procedure.

PPC on position error

We first define the normalized position error

$$\boldsymbol{\xi}_p = \begin{bmatrix} \xi_{p_x} \\ \xi_{p_y} \\ \xi_{p_z} \end{bmatrix} = \boldsymbol{\rho}_p(t)^{-1} \boldsymbol{e}_p \tag{3.6}$$

where $\boldsymbol{\rho}_p = \text{diag}\{[\rho_{p_x}, \rho_{p_y}, \rho_{p_z}]\} \in \mathbb{R}^{3\times 3}$. Next, we define the transformation

$$\boldsymbol{\varepsilon}_p = \text{T}(\boldsymbol{\xi}_p) \tag{3.7}$$

where T is given by (2.33) and we design the reference velocity signal

$$\boldsymbol{v}_r = \begin{bmatrix} \boldsymbol{v}_{xy,r} \\ v_{z,r} \end{bmatrix} = -k_p \boldsymbol{\rho}_p^{-1} \boldsymbol{r}_p \boldsymbol{\varepsilon}_p \tag{3.8}$$

where $\boldsymbol{r}_p = \frac{\text{dT}(\boldsymbol{\xi}_p)}{\text{d}\boldsymbol{\xi}_p} = \text{diag}\left\{\frac{1}{1-\xi_{p_x}^2}, \frac{1}{1-\xi_{p_y}^2}, \frac{1}{1-\xi_{p_z}^2}\right\}$ and k_p is a positive control gain. From the first step, we obtain the velocity reference $\boldsymbol{v}_r$ that will be used shortly.

PPC on velocity error

Following a backstepping-like procedure, we define the error

$$\boldsymbol{e}_v = \begin{bmatrix} \boldsymbol{e}_{v_{xy}} \\ e_{v_z} \end{bmatrix} = \begin{bmatrix} e_{v_x} \\ e_{v_y} \\ e_{v_z} \end{bmatrix} = \boldsymbol{v} - \boldsymbol{v}_r = \begin{bmatrix} \boldsymbol{v}_{xy} \\ v_z \end{bmatrix} - \begin{bmatrix} \boldsymbol{v}_{xy,r} \\ v_{z,r} \end{bmatrix} \tag{3.9}$$

Next, we introduce the corresponding exponential performance functions

$$\rho_{v_i}(t) = (\rho_{v_i,0} - \rho_{v_i,\infty})e^{-l_{v_i}t} + \rho_{v_i,\infty},\ i \in \{x, y, z\}$$

such that $\rho_{v_i}(0) = \rho_{v_i,0} > |e_{v_i}(0)|$, for $i \in \{x, y, z\}$, which leads to the normalized error

$$\boldsymbol{\xi}_v = \begin{bmatrix} \boldsymbol{\xi}_{v_{xy}} \\ \xi_{v_z} \end{bmatrix} = \begin{bmatrix} \xi_{v_x} \\ \xi_{v_y} \\ \xi_{v_z} \end{bmatrix} = \boldsymbol{\rho}_v(t)^{-1}\boldsymbol{e}_v, \tag{3.10}$$

with $\boldsymbol{\rho}_v = \text{diag}\{\rho_{v_x}, \rho_{v_y}, \rho_{v_z}\}$. Next, we define the transformation

$$\boldsymbol{\varepsilon}_v = \begin{bmatrix} \boldsymbol{\varepsilon}_{v_{xy}} \\ \varepsilon_{v_z} \end{bmatrix} = \text{T}(\boldsymbol{\xi}_v) \tag{3.11}$$

and set **the control input** F_z as

$$F_z = -k_{v_z}\rho_{v_z}^{-1}r_{v_z}\varepsilon_{v_z} \tag{3.12}$$

where $r_{v_z} = \frac{\text{dT}(\xi_{v_z})}{\text{d}\xi_{v_z}} = \frac{1}{1-\xi_{v_z}^2}$ and k_{v_z} is a positive control gain.

Moreover, we define the reference signal for $\boldsymbol{T}_{\phi\theta}$, defined in (3.4), as

$$\boldsymbol{T}_{\phi\theta,r} = \begin{bmatrix} T_{\phi\theta_1,r} \\ T_{\phi\theta_2,r} \end{bmatrix} = -k_{v_{xy}}\frac{\boldsymbol{R}_\psi^T\boldsymbol{\rho}_{v_{xy}}^{-1}\boldsymbol{r}_{v_{xy}}\boldsymbol{\varepsilon}_{v_{xy}}}{F_z}, \tag{3.13}$$

where $k_{v_{xy}}$ is a positive control gain, $\boldsymbol{\rho}_{v_{xy}} = \text{diag}\{\rho_{v_x}, \rho_{v_y}\}$ is a matrix consisting of previously defined performance functions, and $\boldsymbol{r}_{v_{xy}} = \frac{\text{dT}(\boldsymbol{\xi}_{v_{xy}})}{\text{d}\boldsymbol{\xi}_{v_{xy}}} = \text{diag}\left\{\frac{1}{1-\xi_{v_x}^2}, \frac{1}{1-\xi_{v_y}^2}\right\}$.

As a result of this step in the control design, we obtain the control input F_z that will be directly applied to the system, and the reference $\boldsymbol{T}_{\phi\theta,r}$, that will be used to design the control for angles and angular velocities. Note that in Eq. (3.13), F_z is required to be not equal to zero, which will be a condition in the stability theorem.

PPC on angular errors

The reference signal $\boldsymbol{T}_{\phi\theta,r}$ implicitly assigns reference values for the angles ϕ, θ. Moreover, given the reference ψ_r, we define the respective errors

$$\boldsymbol{e}_{\phi\theta} = \begin{bmatrix} e_{\phi\theta_1} \\ e_{\phi\theta_2} \end{bmatrix} = \boldsymbol{T}_{\phi\theta} - \boldsymbol{T}_{\phi\theta,r} \tag{3.14a}$$

$$e_\psi = \psi - \psi_r \tag{3.14b}$$

By further introducing exponential performance functions

$$\rho_i(t) = (\rho_{i,0} - \rho_{i,\infty})e^{-l_i t} + \rho_{i,\infty}$$

such that $\rho_i(0) = \rho_{i,0} > |e_i(0)|$, for $i \in \{\phi\theta_1, \phi\theta_2, \psi\}$, we define the normalized errors

$$\boldsymbol{\xi}_{\phi\theta} = \begin{bmatrix} \xi_{\phi\theta_1} \\ \xi_{\phi\theta_2} \end{bmatrix} = \boldsymbol{\rho}_{\phi\theta}(t)^{-1}\boldsymbol{e}_{\phi\theta} \tag{3.15a}$$

$$\xi_\psi = \rho_\psi(t)^{-1}e_\psi \tag{3.15b}$$

where $\boldsymbol{\rho}_{\phi\theta} = \mathrm{diag}\{\rho_{\phi\theta_1}, \rho_{\phi\theta_2}\}$, and the transformations

$$\boldsymbol{\varepsilon}_{\phi\theta} = \mathrm{T}(\boldsymbol{\xi}_{\phi\theta}) \tag{3.16a}$$

$$\varepsilon_\psi = \mathrm{T}(\xi_\psi) \tag{3.16b}$$

In order to stabilize the aforementioned errors, we design reference signals for the angular velocities

$$\boldsymbol{\omega}_r = \begin{bmatrix} \boldsymbol{\omega}_{\phi\theta,r} \\ \omega_{\psi,r} \end{bmatrix} = \begin{bmatrix} \omega_{\phi,r} \\ \omega_{\theta,r} \\ \omega_{\psi,r} \end{bmatrix} = -\begin{bmatrix} k_{\phi\theta}\boldsymbol{R}_{\phi\theta}^{-1}\boldsymbol{J}_{\phi\theta}^{-1}\boldsymbol{\rho}_{\phi\theta}^{-1}\boldsymbol{r}_{\phi\theta}\boldsymbol{\varepsilon}_{\phi\theta} \\ k_\psi\rho_\psi^{-1}r_\psi\varepsilon_\psi + \omega_\phi c_\psi t_\theta + \omega_\theta s_\psi t_\theta \end{bmatrix} \tag{3.17}$$

where $k_{\phi\theta}$ and k_ψ are positive control gains, and $\boldsymbol{r}_{\phi\theta} = \frac{\mathrm{dT}(\boldsymbol{\xi}_{\phi\theta})}{\mathrm{d}\boldsymbol{\xi}_{\phi\theta}} = \mathrm{diag}\left\{\frac{1}{1-\xi_{\phi\theta_1}^2}, \frac{1}{1-\xi_{\phi\theta_2}^2}\right\}$. Note that $\boldsymbol{R}_{\phi\theta}^{-1}$ and $\boldsymbol{J}_{\phi\theta}^{-1}$ are well-defined due to Assumption 3.1.

PPC on angular velocity errors

The final step is the design of the control inputs $\boldsymbol{\tau}$ for tracking of the reference angular velocities designed in the previous step. To that end, we define first the angular velocity errors

$$\boldsymbol{e}_\omega = \begin{bmatrix} \boldsymbol{e}_{\omega_{\phi\theta}} \\ e_{\omega_\psi} \end{bmatrix} = \boldsymbol{\omega} - \boldsymbol{\omega}_r = \begin{bmatrix} \boldsymbol{\omega}_{\phi\theta} \\ \omega_\psi \end{bmatrix} - \begin{bmatrix} \boldsymbol{\omega}_{\phi\theta,r} \\ \omega_{\psi,r} \end{bmatrix} \tag{3.18}$$

We further introduce exponential performance functions

$$\rho_{\omega_i}(t) = (\rho_{\omega_i,0} - \rho_{\omega_i,\infty})e^{-l_{\omega_i}t} + \rho_{\omega_i,\infty}$$

such that $\rho_{\omega_i}(0) = \rho_{\omega_i,0} > |e_{\omega_i}(0)|$, for $i \in \{\phi, \theta, \psi\}$, to impose predefined performance on the error $\boldsymbol{e}_\omega$, and define the respective normalized error

$$\boldsymbol{\xi}_\omega = \begin{bmatrix} \xi_{\omega_\phi} \\ \xi_{\omega_\theta} \\ \xi_{\omega_\psi} \end{bmatrix} = \boldsymbol{\rho}_\omega(t)^{-1}\boldsymbol{e}_\omega \tag{3.19}$$

Finally, we define the transformation

$$\boldsymbol{\varepsilon}_\omega = \mathrm{T}(\boldsymbol{\xi}_\omega) \tag{3.20}$$

and design **the control input**

$$\boldsymbol{\tau} = -k_\omega \boldsymbol{\rho}_\omega^{-1}\boldsymbol{r}_\omega\boldsymbol{\varepsilon}_\omega \tag{3.21}$$

where $\boldsymbol{r}_\omega = \frac{\mathrm{dT}(\boldsymbol{\xi}_\omega)}{\mathrm{d}\boldsymbol{\xi}_\omega} = \mathrm{diag}\left\{\frac{1}{1-\xi_{\omega_\phi}^2}, \frac{1}{1-\xi_{\omega_\theta}^2}, \frac{1}{1-\xi_{\omega_\psi}^2}\right\}$, $\boldsymbol{\rho}_\omega = \mathrm{diag}\{\rho_{\omega_\phi}, \rho_{\omega_\theta}, \rho_{\omega_\psi}\}$, and k_ω is a positive gain.

With this, the control design procedure is finalized and we can proceed with the stability analysis of the proposed scheme.

3.4 Stability Analysis

The stability analysis of the proposed control protocol is summarized in the next theorem.

Theorem 3.1. *Consider the UAV dynamics* (3.1) *under the proposed control scheme* (3.6)-(3.21) *and Assumption 3.1. Further assume that*

$$F_z(t) \neq 0 \tag{3.22a}$$

$$\frac{k_{v_{xy}}}{k_{v_z}} > \frac{\max_{i\in\{\phi\theta_1,\phi\theta_2\}}\{\rho_{i,0}\}}{4\cos(\bar{\pi})^2} \tag{3.22b}$$

$$|T_{i,r}(t)| \leq \rho_i(t) + 1, i \in \{\phi\theta_1, \phi\theta_2\} \tag{3.22c}$$

for $t \geq 0$, where $\bar{\pi}$ is defined in Assumption 3.1. Then it holds that

$$\begin{aligned} |e_{p_i}(t)| &< \rho_{p_i}(t),\ i \in \{x, y, z\} \\ |e_\psi(t)| &< \rho_\psi(t) \end{aligned}$$

and all closed-loop signals are bounded, for all $t \geq 0$.

Remark 3.1. Condition (3.22a) is needed for the boundedness of the intermediate signal (3.13). Intuitively, it requires the UAV thrust, set in (3.12), to be always positive and compensate for the gravitational force $\boldsymbol{g}$. Note that this is a condition encountered in the related literature (e.g., [90]). One can guarantee (3.22a) by adding an integrator in (3.12) and adjust appropriately the performance function $\rho_{p_z}(t)$; for more details, we refer the reader to [89]. The parameter-related condition (3.22b), which is needed for the correctness of Theorem 3.1, essentially imposes restrictions on the angles θ, ϕ and the error $\boldsymbol{e}_{\phi\theta}$; such error must be small enough, as dictated by the initial performance values $\rho_{\phi\theta_1,0}$, $\rho_{\phi\theta_2,0}$ in the nominator of the right-hand-side of (3.22b), and the angles θ, ϕ themselves must be close to zero, maximizing the denominator of the right-hand-side of (3.22b). Intuitively, the aforementioned constraints require small reference signals $\boldsymbol{T}_{\phi\theta,r}$, which translate to slow reference trajectories $p_{x,r}(t)$ and $p_{y,r}(t)$. Similarly, (3.22c) is needed since the values of $\boldsymbol{T}_{\phi\theta}$ in (3.4) cannot exceed the value of 1. We can enforce such a condition by choosing slowly-converging performance functions $\rho_{\phi\theta_1}(t)$ and $\rho_{\phi\theta_2}(t)$ with large initial values.

Proof. The proof proceeds in three steps. First, we show the existence of a local solution such that $\boldsymbol{\xi}_p(t)$, $\boldsymbol{\xi}_v(t)$, $\boldsymbol{\xi}_\omega(t) \in (-1,1)^3$, $\xi_\psi(t) \in (-1,1)$, $\boldsymbol{\xi}_{\phi\theta}(t) \in (-1,1)^2$ for a time interval $t \in [0, \tau_{\max})$. Next, we show that the proposed control scheme retains the aforementioned normalized signals in compact subsets of $(-1,1)$, which leads to $\tau_{\max} = \infty$ in the final step, thus completing the proof.

Towards the existence of a local solution, consider first the overall state vector $\boldsymbol{x} = [\boldsymbol{p}^T, \boldsymbol{v}^T, \boldsymbol{\eta}^T, \boldsymbol{\omega}^T]^T \in \mathbb{X} = \mathbb{R}^6 \times (-\frac{\pi}{2}, \frac{\pi}{2})^2 \times (-\pi, \pi) \times \mathbb{R}^3$ and let us define the open set:

$$\begin{aligned}\Omega = \big\{ &(\boldsymbol{x}, t) \in \mathbb{X} \times [0, \infty) : \boldsymbol{\xi}_p \in (-1,1)^3, \boldsymbol{\xi}_v \in (-1,1)^3, \\ &\boldsymbol{\xi}_{\phi\theta} \in (-1,1)^2, \xi_\psi \in (-1,1), \boldsymbol{\xi}_\omega \in (-1,1)^3 \big\}. \end{aligned} \tag{3.23}$$

Note that the choice of the performance functions at $t = 0$ implies that $\boldsymbol{\xi}_p(0)$, $\boldsymbol{\xi}_v(0)$, $\boldsymbol{\xi}_\omega(0) \in (-1,1)^3$, $\boldsymbol{\xi}_{\phi\theta}(0) \in (-1,1)^2$, and $\xi_\psi(0) \in (-1,1)$, implying that Ω is nonempty. By combining (3.1), (3.12), and (3.21), we obtain the closed-loop system dynamics $\dot{\boldsymbol{x}} = f_{\boldsymbol{x}}(\boldsymbol{x}, t)$, where $f_{\boldsymbol{x}} : \mathbb{X} \times [0, \tau_{\max})$ is a function continuous in t and locally Lipschitz in $\boldsymbol{x}$.

Hence, the conditions of Theorem 2.2 are satisfied and we conclude that there exists a unique and local solution $\boldsymbol{x} : [0, \tau_{\max}) \to \mathbb{X}$ such that $(\boldsymbol{x}(t), t) \in \Omega$ for $t \in [0, \tau_{\max})$. Therefore, it holds that

$$\boldsymbol{\xi}_p \in (-1,1)^3 \tag{3.24a}$$
$$\boldsymbol{\xi}_v \in (-1,1)^3 \tag{3.24b}$$
$$\boldsymbol{\xi}_{\phi\theta} \in (-1,1)^2 \tag{3.24c}$$
$$\xi_\psi \in (-1,1) \tag{3.24d}$$
$$\boldsymbol{\xi}_\omega \in (-1,1)^3 \tag{3.24e}$$

for all $t \in [0, \tau_{\max})$.

Next, we proceed to show that the normalized errors in (3.24) remain in compact subsets of $(-1, 1)$. Note that (3.24) implies that that transformed errors $\boldsymbol{\varepsilon}_p$, $\boldsymbol{\varepsilon}_v$, $\boldsymbol{\varepsilon}_{\phi\theta}$, ε_ψ, $\boldsymbol{\varepsilon}_\omega$ are well-defined for $t \in [0, \tau_{\max})$. Consider now the candidate Lyapunov function:

$$V_p = \frac{1}{2}\|\boldsymbol{\varepsilon}_p\|^2$$

Differentiating V_p along the local solution $x(t)$ yields

$$\dot{V}_p = \boldsymbol{\varepsilon}_p^T \boldsymbol{r}_p \boldsymbol{\rho}_p^{-1}(\boldsymbol{v} - \dot{\boldsymbol{p}}_r - \dot{\boldsymbol{\rho}}_p \boldsymbol{\xi}_p)$$

By using $\boldsymbol{v} = \boldsymbol{e}_v + \boldsymbol{v}_r$, (3.8), the boundedness of $\dot{\boldsymbol{p}}_r$, $\dot{\boldsymbol{\rho}}_p$, and (3.24), $\dot{V}_p$ becomes

$$\dot{V}_p \leq -k_p \|\boldsymbol{\rho}_p^{-1} \boldsymbol{r}_p \boldsymbol{\varepsilon}_p\|^2 + \|\boldsymbol{\rho}_p^{-1} \boldsymbol{r}_p \boldsymbol{\varepsilon}_p\| \bar{F}_p$$

where $\bar{F}_p$ is a constant, independent of $\tau_{\max}$, satisfying

$$\|\boldsymbol{\rho}_v \boldsymbol{\xi}_v - \dot{\boldsymbol{p}}_r - \dot{\boldsymbol{\rho}}_p \boldsymbol{\xi}_p\| \leq \bar{F}_p,$$

for all $t \in [0, \tau_{\max})$. Therefore, we conclude that $\dot{V}_p < 0$ when $\|\boldsymbol{\rho}_p^{-1} \boldsymbol{r}_p \boldsymbol{\varepsilon}_p\| > \frac{\bar{F}_p}{k_p}$. In view of the definition of $\boldsymbol{r}_p$, we conclude that

$$\dot{V} < 0 \quad \text{when} \quad \|\boldsymbol{\varepsilon}_p\| > \frac{\bar{F}_p \max_{i \in \{x,y,z\}}\{\rho_{p_i,0}\}}{k_p}.$$

Hence, by invoking Theorem 2.4, we conclude that

$$\|\boldsymbol{\varepsilon}_p\| \leq \bar{\varepsilon}_p = \max\left\{\|\boldsymbol{\varepsilon}_p(0)\|, \frac{\bar{F}_p \max_{i \in \{x,y,z\}}\{\rho_{p_i,0}\}}{k_p}\right\} \tag{3.25a}$$

for $t \in [0, \tau_{\max})$, and by employing the inverse of (2.31), we obtain

$$|\xi_{p_i}(t)| \leq \bar{\xi}_p = \tanh \bar{\varepsilon}_p < 1 \tag{3.25b}$$

for $t \in [0, \tau_{\max})$ and $i \in \{x, y, z\}$. Therefore, we conclude the boundedness of $\boldsymbol{v}_r(t)$ and $\boldsymbol{v}(t) = \boldsymbol{e}_v(t) + \boldsymbol{v}_r(t) = \boldsymbol{\rho}_v(t)\boldsymbol{\xi}_v(t) + \boldsymbol{v}_r(t)$ for all $t \in [0, \tau_{\max})$. By differentiating $\boldsymbol{v}_r(t)$ and using (3.25), we further conclude the boundedness of $\dot{\boldsymbol{v}}_r(t)$ for all $t \in [0, \tau_{\max})$.

We consider next the function $V_v = \frac{1}{2}\|\boldsymbol{\varepsilon}_v\|^2$, whose derivative, in view of (3.1), and (3.5), yields

$$\begin{aligned}\dot{V}_v =& \boldsymbol{\varepsilon}_{v_{xy}}^T \boldsymbol{r}_{v_{xy}} \boldsymbol{\rho}_{v_{xy}}^{-1} \left(\frac{1}{m}(\boldsymbol{R}_\psi \boldsymbol{T}_{\phi\theta} F_z + \boldsymbol{F}_{d,xy}) - \dot{\boldsymbol{v}}_{xy,r} - \dot{\boldsymbol{\rho}}_{v_{xy}} \boldsymbol{\xi}_{v_{xy}}\right) \\ &+ \varepsilon_{v_z}^T r_{v_z} \rho_{v_z}^{-1} \left(\frac{1}{m}(c_\theta c_\phi F_z + F_{d,z}) - g - \dot{v}_{z,r} - \dot{\rho}_{v_z} \xi_{v_z}\right)\end{aligned}$$

for $t \in [0, \tau_{\max})$. By using $\boldsymbol{T}_{\phi\theta} = \boldsymbol{e}_{\phi\theta} + \boldsymbol{T}_{\phi\theta,r}$, (3.13), (3.12), the boundedness of $\dot{\boldsymbol{v}}_r$, $\dot{\boldsymbol{\rho}}_v$, and (3.24), and the continuity and boundedness of $\boldsymbol{F}_d(\boldsymbol{x}, \dot{\boldsymbol{x}}, t)$ in $(\boldsymbol{x}, \dot{\boldsymbol{x}})$ and t, respectively, we arrive at

$$\begin{aligned}\dot{V}_v \leq & -\frac{k_{v_{xy}}}{m}\|\boldsymbol{\rho}_{v_{xy}}^{-1}\boldsymbol{r}_{v_{xy}}\boldsymbol{\varepsilon}_{v_{xy}}\|^2 - \frac{k_{v_z}c_\theta c_\phi}{m}\|\rho_{v_z}^{-1}r_{v_z}\varepsilon_{v_z}\|^2 \\ & + \frac{k_{v_z}\bar{\rho}_{\phi\theta}}{m}\|\boldsymbol{\rho}_{v_{xy}}^{-1}\boldsymbol{r}_{v_{xy}}\boldsymbol{\varepsilon}_{v_{xy}}\|\|\rho_{v_z}^{-1}r_{v_z}\varepsilon_{v_z}\| + \|\boldsymbol{\rho}_v^{-1}\boldsymbol{r}_v\boldsymbol{\varepsilon}_v\|\bar{F}_v\end{aligned}$$

where $\bar{F}_v$ is a positive constant, independent of $\tau_{\max}$, satisfying

$$\|\boldsymbol{F}_d - \boldsymbol{g} - \dot{\boldsymbol{v}}_r - \dot{\boldsymbol{\rho}}_v\boldsymbol{\xi}_v\| \leq \bar{F}_v$$

for $t \in [0, \tau_{\max})$, and $\bar{\rho}_{\phi\theta} = \max_{i\in\{\phi\theta_1,\phi\theta_2\}}\{\rho_{i,0}\}$. Additionally, Assumption 3.1 implies that $c_\theta c_\phi \geq \bar{c} = \cos(\bar{\pi})^2 > 0$.

Let us now define a constant $\alpha > 0$ such that $k_{v_{xy}} > \frac{k_{v_z}\sqrt{\alpha}}{2}$, $\cos(\bar{\pi})^2 > \frac{\bar{\rho}_{\phi\theta}}{2\sqrt{\alpha}}$. Note that such a constant exists due to (3.22b). By completing the squares, $\dot{V}_v$ becomes

$$\dot{V}_v \leq -\kappa_v\|\boldsymbol{\rho}_v^{-1}\boldsymbol{r}_v\boldsymbol{\varepsilon}_v\|^2 + \|\boldsymbol{\rho}_v^{-1}\boldsymbol{r}_v\boldsymbol{\varepsilon}_v\|\bar{F}_v$$

for $t \in [0, \tau_{\max})$, where $\kappa_v = \min\{\kappa_{v_{xy}}, \kappa_{v_z}\}$, and

$$\kappa_{v_{xy}} = \frac{k_{v_{xy}}}{m} - \frac{k_{v_z}\sqrt{\alpha}}{2m}, \quad \kappa_{v_z} = k_{v_z}\left(\frac{\bar{c}}{m} - \frac{\bar{\rho_\phi}_\theta}{2m\sqrt{\alpha}}\right)$$

Therefore, by following a similar procedure as with V_p, we conclude that

$$\|\boldsymbol{\varepsilon}_v(t)\| \leq \bar{\varepsilon}_v = \max\left\{\|\boldsymbol{\varepsilon}_v(0)\|, \frac{\bar{F}_v \max_{i\in\{x,y,z\}}\{\rho_{v_i,0}\}}{\kappa_v}\right\} \tag{3.26a}$$

$$|\xi_{v_i}(t)| \leq \bar{\xi}_v = \tanh\bar{\varepsilon}_v < 1 \tag{3.26b}$$

for $t \in [0, \tau_{\max})$ and $i \in \{x, y, z\}$. Also, in view of (3.22a), we conclude the boundedness of $\boldsymbol{T}_{\phi\theta,r}$, F_z, $\boldsymbol{T}_{\phi\theta} = \boldsymbol{\rho}_{\phi\theta}(t) + \boldsymbol{T}_{\phi\theta,r}$, for $t \in [0, \tau_{\max})$. By differentiating $\boldsymbol{T}_{\phi\theta,r}$ and F_z and using (3.26) and (3.22a), we further conclude the boundedndess of $\dot{\boldsymbol{T}}_{\phi\theta,r}(t)$ and $\dot{F}_z(t)$, for all $t \in [0, \tau_{\max})$.

Following a similar line of proof, we consider now the function $V_\eta = \frac{1}{2}\|\boldsymbol{\varepsilon}_{\phi\theta}\|^2 + \frac{1}{2}\varepsilon_\psi^2$, whose derivative, in view of (3.1), becomes

$$\begin{aligned}\dot{V}_\eta = & \boldsymbol{\varepsilon}_{\phi\theta}^T\boldsymbol{r}_{\phi\theta}\boldsymbol{\rho}_{\phi\theta}^{-1}\left(\boldsymbol{J}_{\phi\theta}\boldsymbol{R}_{\phi\theta}\boldsymbol{\omega}_{\phi\theta} - \dot{\boldsymbol{T}}_{\phi\theta,r} - \dot{\boldsymbol{\rho}}_{\phi\theta}\boldsymbol{\xi}_{\phi\theta}\right) \\ & + \varepsilon_\psi\rho_\psi^{-1}r_\psi(c_\psi t_\theta\omega_\phi + s_\psi t_\theta\omega_\theta + \omega_\psi - \dot{\psi}_r - \dot{\rho}_\psi\xi_\psi)\end{aligned}$$

By using $\boldsymbol{\omega}_{\phi\theta} = \boldsymbol{e}_{\omega_{\phi\theta}} + \boldsymbol{\omega}_{\phi\theta,r}$, $\psi = \psi_r + e_\psi$, (3.17), (3.24), and the continuity of $\boldsymbol{J}_{\phi\theta}$, Assumption 3.1, and the boundedness of $\dot{\boldsymbol{T}}_{\phi\theta,r}$, $\dot{\boldsymbol{\rho}}_{\phi\theta}$, ρ_ψ, one obtains

$$\begin{aligned}\dot{V}_\eta \leq & -k_{\phi\theta}\|\boldsymbol{\rho}_{\phi\theta}^{-1}\boldsymbol{r}_{\phi\theta}\boldsymbol{\varepsilon}_{\phi\theta}\|^2 + \|\boldsymbol{\rho}_{\phi\theta}^{-1}\boldsymbol{r}_{\phi\theta}\boldsymbol{\varepsilon}_{\phi\theta}\|\bar{F}_{\phi\theta} - k_\psi(\rho_\psi^{-1}r_\psi\varepsilon_\psi)^2 + |\rho_\psi^{-1}r_\psi\varepsilon_\psi|\bar{F}_\psi \\ \leq & -k_\eta\|\boldsymbol{\rho}_\eta^{-1}\boldsymbol{r}_\eta\boldsymbol{\varepsilon}_\eta\|^2 + \|\boldsymbol{\rho}_\eta^{-1}\boldsymbol{r}_\eta\boldsymbol{\varepsilon}_\eta\|\bar{F}_\eta\end{aligned}$$

where $\bar{F}_{\phi\theta}$ and $\bar{F}_{\psi}$ are positive constants, independent of $\tau_{\max}$, satisfying

$$\|\boldsymbol{J}_{\phi\theta}\boldsymbol{R}_{\phi\theta}\boldsymbol{e}_{\omega_{\phi\theta}} - \dot{\boldsymbol{T}}_{\phi\theta,r} - \dot{\boldsymbol{\rho}}_{\phi\theta}\boldsymbol{\xi}_{\phi\theta}\| \leq \bar{F}_{\phi\theta} \quad \text{and} \quad |e_{\psi} - \dot{\psi}_r - \dot{\psi}\xi_{\psi}| \leq \bar{F}_{\psi}$$

for $t \in [0, \tau_{\max})$. We define $\boldsymbol{\rho}_{\eta} = \operatorname{diag}\{\boldsymbol{\rho}_{\phi\theta}, \rho_{\psi}\}$, $\boldsymbol{r}_{\eta} = \operatorname{diag}\{\boldsymbol{r}_{\phi\theta}, r_{\psi}\}$, $\boldsymbol{\varepsilon}_{\eta} = [\boldsymbol{\varepsilon}_{\phi\theta}^T, \varepsilon_{\psi}]^T$, $k_{\eta} = \min\{k_{\phi\theta}, k_{\psi}\}$, and $\bar{F}_{\eta} = \max\{\bar{F}_{\psi}, \bar{F}_{\phi\theta}\}$.

Thus, it holds that $\dot{V}_{\eta} < 0$ when $\|\boldsymbol{\rho}_{\eta}^{-1}\boldsymbol{r}_{\eta}\boldsymbol{\varepsilon}_{\eta}\| > \frac{\bar{F}_{\eta}}{k_{\eta}}$, which, similar to the previous steps, leads to

$$\|\boldsymbol{\varepsilon}_{\eta}(t)\| \leq \bar{\varepsilon}_{\eta} = \max\left\{\|\boldsymbol{\varepsilon}_{\eta}(0)\|, \frac{\bar{F}_{\eta}\max_{i\in\phi,\theta,\psi}\{\rho_{i,0}\}}{k_{\eta}}\right\} \tag{3.27a}$$

$$|\xi_i(t)| \leq \bar{\xi}_{\eta} = \tanh(\bar{\varepsilon}_{\eta}) \tag{3.27b}$$

for $t \in [0, \tau_{\max})$ and $i \in \{\phi\theta_1, \phi\theta_2, \psi\}$. Therefore, in view of the boundedness of $\boldsymbol{e}_{\omega}(t)$, we conclude the boundedness of $\boldsymbol{\omega}_{\phi\theta,r}(t)$, $\omega_{\psi,r}(t)$ and hence of $\omega(t)$, for all $t \in [0, \tau_{\max})$. By using (3.27), we further conclude the boundedness of $\dot{\boldsymbol{\omega}}_{\phi\theta,r}(t)$ and $\dot{\omega}_{\psi,r}(t)$ for all $t \in [0, \tau_{\max})$.

Finally, using a similar line of proof and considering the function $V_{\omega} = \frac{1}{2}\|\boldsymbol{e}_{\omega}\|^2$, we conclude that

$$\|\boldsymbol{\varepsilon}_{\omega}(t)\| \leq \bar{\varepsilon}_{\omega} = \max\left\{\|\boldsymbol{\varepsilon}_{\omega}(0)\|, \frac{\bar{F}_{\omega}\max_{i\in\{\phi,\theta,\psi\}}\{\rho_{\omega_i,0}\}}{k_{\omega}\lambda}\right\} \tag{3.28a}$$

$$|\xi_{\omega_i}(t)| \leq \tanh(\bar{\varepsilon}_{\omega}) < 1, \tag{3.28b}$$

for $t \in [0, \tau_{\max})$ and $i \in \{\phi, \theta, \psi\}$, where λ is the positive minimum eigenvalue of the positive definite inertia matrix $\boldsymbol{\mathcal{I}}(\boldsymbol{\eta})$. $\bar{F}_{\omega}$ is a positive constant satisfying

$$\|\boldsymbol{\tau}_d - \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{\omega} \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega} - \dot{\boldsymbol{\omega}}_r - \dot{\boldsymbol{\rho}}_{\omega}\boldsymbol{\xi}_{\omega}\| \leq \bar{F}_{\omega},$$

where we use the boundedness of $\boldsymbol{\xi}_{\omega}$ and $\boldsymbol{\omega}$ from (3.24), the boundedness of $\dot{\boldsymbol{\omega}}_{\phi\theta,r}(t)$ from the previous step, and the boudnedness of $\boldsymbol{\tau}_d$ due to its continuity in $(\boldsymbol{x}, \dot{\boldsymbol{x}})$ and boundedness in t. Finally, (3.28) leads to the boundedness of $\boldsymbol{\tau}(t)$ for all $t \in [0, \tau_{\max})$.

What remains to be shown is that $\tau_{\max} = \infty$. Towards that end, note that (3.25), (3.26), (3.27), and (3.28) imply that $(\boldsymbol{x}(t), t)$ remain in a compact subset of Ω, i.e., there exists a positive constant $\underline{d}$ such that $d((\boldsymbol{x}(t), t), \partial\Omega) \geq \underline{d} > 0$, for all $t \in [0, \tau_{\max})$, where $d((\boldsymbol{x}(t), t), \partial\Omega)$ is in view of definition within Theorem 2.3. Since all closed-loop signals have already been proven bounded, it holds that

$$\lim_{t\to\tau_{\max}^-}\left(\|\boldsymbol{x}(t)\| + \frac{1}{d((\boldsymbol{x}(t), t), \partial\Omega)}\right) \leq \bar{d}$$

for some finite constant $\bar{d}$, and hence direct application of Theorem 2.3 dictates that $\tau_{\max} = \infty$, which concludes the proof. □

Remark 3.2. From the aforementioned proof it can be deduced that the proposed control scheme achieves its goals without resorting to the need of rendering the ultimate bounds $\bar{\varepsilon}_p$, $\bar{\varepsilon}_v$,$\bar{\varepsilon}_\eta$, $\bar{\varepsilon}_\omega$ of the transformed errors arbitrarily small by adopting extreme values of the control gains k_p, $k_{v_{xy}}$, k_{v_z}, $k_{\phi\theta}$, k_ψ, and k_ω. Notice that (3.25), (3.26), (3.27), and (3.28) hold no matter how large the finite bounds $\bar{\varepsilon}_p$, $\bar{\varepsilon}_v$, $\bar{\varepsilon}_\eta$, $\bar{\varepsilon}_\omega$ are and regardless of the choice of the control gains. In the same spirit, large uncertainties involved in the nonlinear model (3.1) can be compensated, as they affect only the size of these bounds through $\bar{F}_v$ and $\bar{F}_\omega$, but leave unaltered the achieved stability properties. Hence, the actual performance given in (3.3), which is solely determined by the designer-specified performance functions, becomes isolated against model uncertainties, thus extending greatly the robustness of the proposed control scheme.

Remark 3.3. It should be noted that the selection of the control gains affects both the quality of evolution of the errors $\boldsymbol{e}_p$, e_ψ within the corresponding performance envelopes as well as the control input characteristics. Additionally, fine tuning might be needed in real-time scenarios, to retain the required control input signals within the feasible range that can be implemented by the actuators. Similarly, the control input constraints impose an upper bound on the required speed of convergence of $\rho_{p_i}(t)$, $i \in \{x, y, z\}$, $\rho_\psi(t)$, as obtained by the exponentials $e^{-l_{p_i}t}$, $i \in \{x, y, z\}$, and $e^{-l_\psi t}$, respectively. Hence, the selection of the control gains k_p, $k_{v_{xy}}$, k_{v_z}, $k_{\phi\theta}$, k_ψ, k_ω can have positive influence on the overall closed loop system response. More specifically, notice that $\bar{F}_v$ and $\bar{F}_\omega$ provide implicit bounds on $\bar{\varepsilon}_v$ and $\bar{\varepsilon}_\omega$, respectively. Therefore, invoking (3.12) and (3.21), we can select the control gains such that F_z and $\boldsymbol{\tau}$ are retained within certain bounds. Nevertheless, the constants $\bar{F}_v$ and $\bar{F}_\omega$ involve the parameters of the model and the external disturbances. Thus, an upper bound of the dynamic parameters of the system as well as of the exogenous disturbances should be given in order to extract any relations between the achieved performance and the input constraints. Finally, we stress that the scalar control gains k_p, $k_{v_{xy}}$, k_{v_z}, $k_{\phi\theta}$, k_ψ, k_ω can be replaced by diagonal matrices, adding more flexibility in the control design, without affecting the stability analysis.

3.5 Simulation Results

We evaluate the proposed control algorithm to the case of tracking reference trajectories $\boldsymbol{p}_r(t) = [p_{x,r}(t), p_{y,r}(t), p_{z,r}]^T$ in ascent and landing scenarios. To ensure that errors start inside of the funnel we choose the prescribed performance functions according to the parameters in Table 3.1. The control gains are selected as

$$\begin{aligned} k_p &= \text{diag}\{1.25, 1.25, 12.5\} \quad & k_{v_z} &= 10 \quad & k_{v_{xy}} &= \text{diag}\{1, 2\} \\ k_{\phi\theta} &= \text{diag}\{3, 1.5\} & k_\psi &= 1 & k_\omega &= 10 I_3 \end{aligned}$$

The parameters are identical in both scenarios.

description	function	indices	ρ_0	ρ_∞	l
position	$\rho_{p_i}(t)$	$i = \{x, y, z\}$	12	0.2	0.4
horizontal velocity	$\rho_{v_i}(t)$	$i = \{x, y\}$	3	0.5	0.5
vertical velocity	$\rho_{v_z}(t)$		5	0.2	1.5
angles	$\rho_{\phi\theta_j}(t)$	$j = \{1, 2\}$	0.5	0.25	0.5
	$\rho_\psi(t)$		0.4	0.05	0.1
angular velocity	$\rho_{\omega_j}(t)$	$j = \{\phi, \theta, \psi\}$	0.3	0.1	0.5

Table 3.1: Parameters of performance functions

Ascent trajectory

The reference ascent trajectory is constructed as a lemniscate ("∞-shaped" trajectory) in the horizontal plane with the ramp function in the vertical direction and

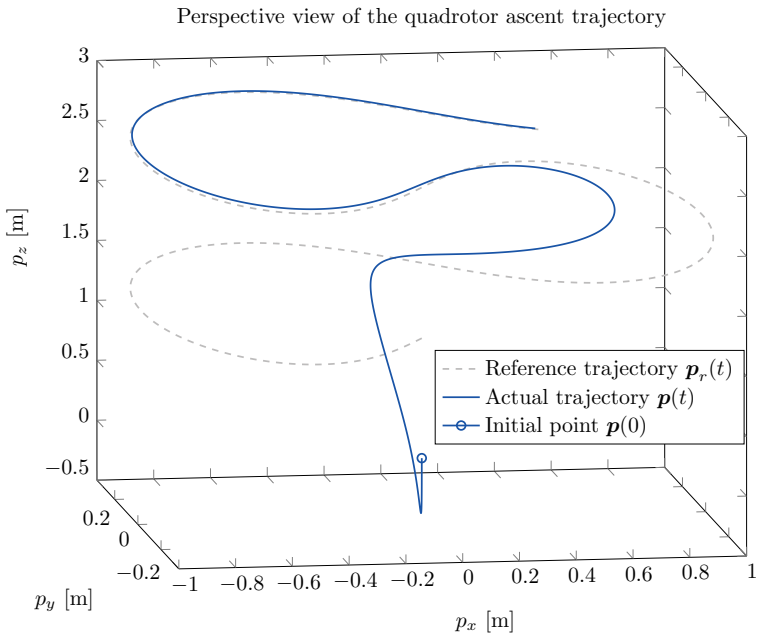

Figure 3.1: Perspective view of trajectory tracking. Note that initial drop in the actual trajectory occurs due to zero initial controlled thrust and no simulated surface beneath.

zero yaw reference

$$
\begin{aligned}
p_{x,r}(t) &= \frac{\cos(t)}{1+\sin^2(t)} \\
p_{y,r}(t) &= \frac{\sin(t)\cos(t)}{1+\sin^2(t)} \\
p_{z,r}(t) &= 1+\frac{1}{5}t \\
\psi_r(t) &= 0
\end{aligned}
$$

starting from the origin $\boldsymbol{p}(0) = [0,0,0]^T$. In Fig. 3.1 we see how the big initial displacement from the reference signal is gradually reduced until the tracking error is eliminated. A comparison of all state signals during time against the given and designed references is provided in Fig. 3.2. Finally, in Fig. 3.3 we observe that all error signals remain inside of the prescribed funnels during the simulation and that errors converge according to the specified performance functions.

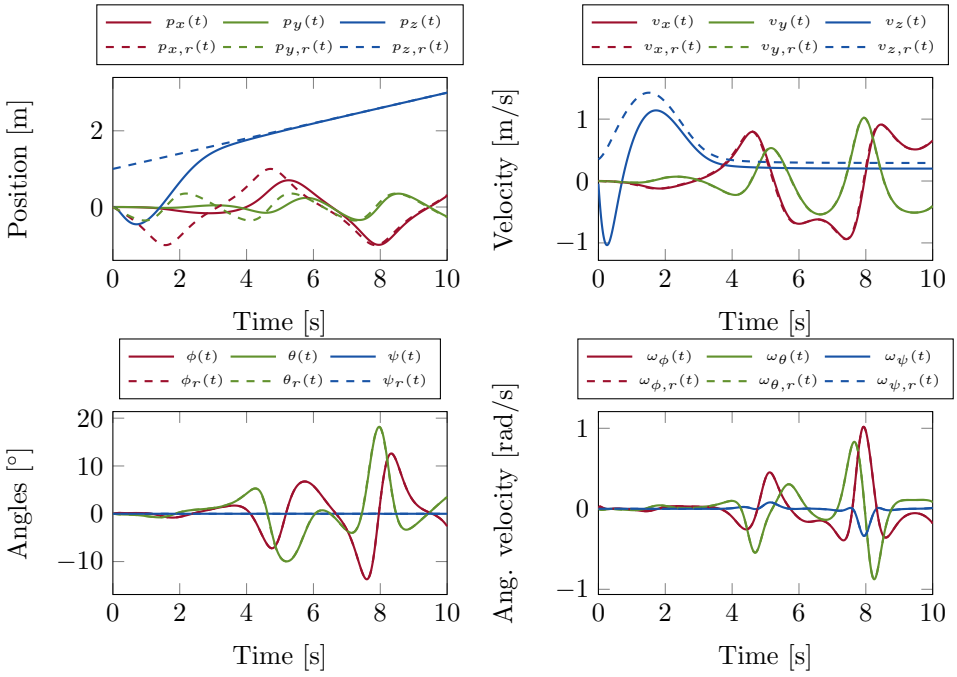

Figure 3.2: The evolution of state signals $\boldsymbol{p}(t), \boldsymbol{v}(t), \boldsymbol{\eta}(t), \boldsymbol{\omega}(t)$ compared to the given reference $\boldsymbol{p}_r(t), \psi_r(t)$ and the designed reference signals $\boldsymbol{v}_r(t), \boldsymbol{\omega}_r(t)$ as well as reference angles extracted from $\boldsymbol{T}_{\phi\theta,r}$.

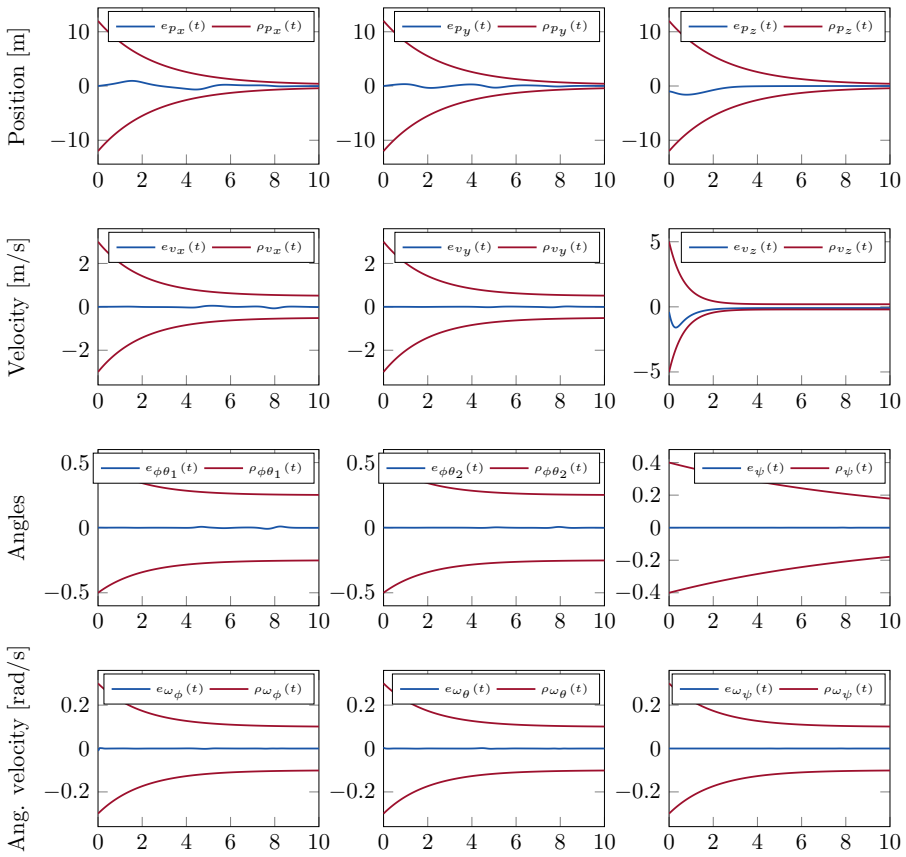

Figure 3.3: Error evolutions stay inside of the prescribed funnels during the experiment. Note that errors $\boldsymbol{e}_{\phi\theta} = [e_{\phi\theta_1}, e_{\phi\theta_2}]^T = \boldsymbol{T}_{\phi\theta} - \boldsymbol{T}_{\phi\theta,r}$ are dimensionless and e_ψ is in radians.

Landing scenario

Quadrotor UAV is landing on a boat USV moving according to a predefined trajectory $\boldsymbol{p}_b(t) = [p_{b,x}(t), p_{b,y}(t)]^T$ which is given as a solution of the following dynamical system with the control input $u(t)$

$$\begin{aligned} \dot{p}_{b,x}(t) &= \cos(\alpha(t)) \\ \dot{p}_{b,y}(t) &= \sin(\alpha(t)) \\ \dot{\alpha}(t) &= u(t) \end{aligned} \qquad u(t) = \begin{cases} -1 & 0 \leq t \leq \frac{3\pi}{4} \\ 1 & \frac{3\pi}{4} < t \leq \frac{9\pi}{4} \\ -1 & \frac{9\pi}{4} < t \leq \frac{11\pi}{4} \\ 0 & \frac{11\pi}{4} < t \leq 10 \end{cases}$$

It is assumed that the quadrotor is aware of the trajectory of the boat and the quadrotor reference is thus $\psi_r(t) = 0$, $\boldsymbol{p}_r(t) = [p_{b,x}(t), p_{b,y}(t), p_{z,r}(t)]^T$, with

$$p_{z,r}(t) = z_d \left(1 - \frac{1}{1 + e^{-(t-t_d)}}\right),$$

where $z_d = 5$ is the initial height and $t_d = 5$ is a tuning parameter of the descent time.

The perspective view of the landing is depicted in Fig. 3.4 and error signals evolution inside of funnels is shown in Fig. 3.5.

The errors between the references and states in the attitude subsystem in both cases are very small, thus achieving almost perfect tracking. The performance functions are chosen such that the roll and pitch angle errors through the transformation $\boldsymbol{T}_{\phi\theta}$ in (3.4), are always less than 15°, approximately. This makes the control effort sufficiently aggressive while at the same time enables handling of stronger disturbances. The position subsystem errors are allowed to be quite big at the beginning of the transient and are sharply reduced to 0.2m error bound. The most aggressive performance satisfaction is required on the vertical velocity v_z to offset the gravity. This is in line with the discussion in Remark 3.1.

Perspective view of the quadrotor trajectory in the landing scenario

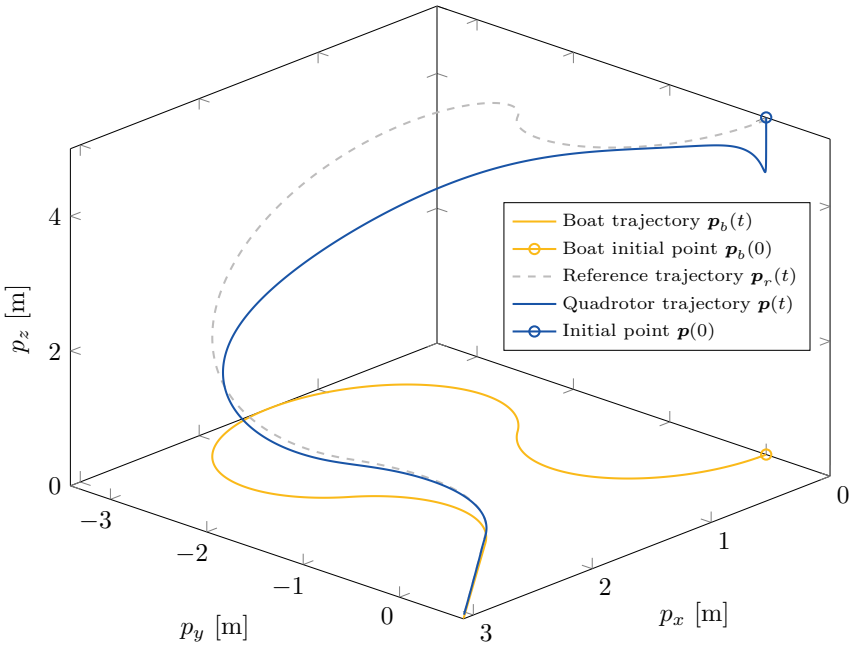

Figure 3.4: Perspective view of the landing scenario

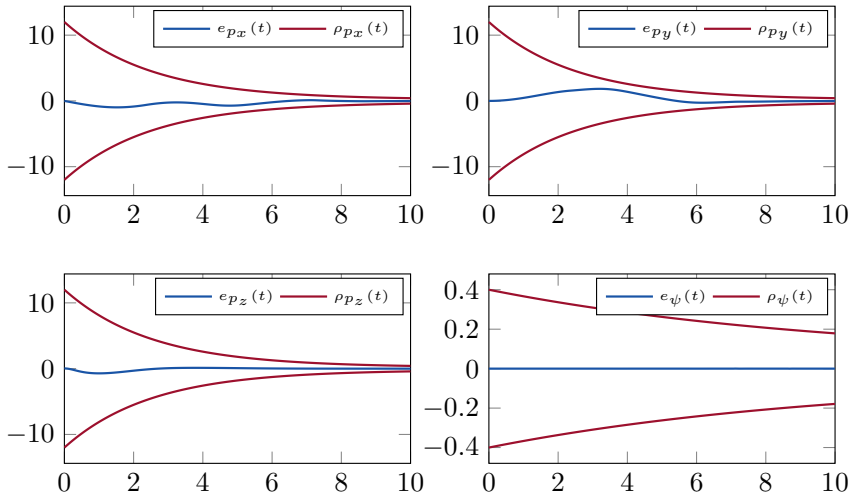

Figure 3.5: Error signals of the given references $\boldsymbol{p}_r(t), \psi_r(t)$ stay inside of the prescribed funnels during the experiment. The rest is omitted due to brevity.

3.6 Conclusion

In this chapter we presented an approach in control design for the trajectory tracking of a quadrotor UAV using the Prescribed Performance Control methodology. Theoretical guarantees are established and controller is validated in the simulation. Future work will include the deployment of the controller on the real quadrotor and testing it in experimental environments. Furthermore, we seek to extend the framework with trajectory generation and coordination in multi-agent scenarios. In the next chapter, we will present a modified PPC for USV and an advanced kinodynamic trajectory generation that can also be used for UAVs.

Chapter 4

Kinodynamic Motion Planning via Funnel Control for Unmanned Surface Vehicle

In this chapter, we continue the discussion about the funnel control for underactuated unmanned vehicles. Namely, we present an algorithm to control an underactuated unmanned surface vehicle (USV) using kinodynamic motion planning with funnel control (KDF). KDF has two main components: motion planning with RRT used to generate trajectories, and funnel control based on prescribed performance control to track the trajectory with respect to kinodynamic constraints. We extend prescribed performance control to accommodate for underactuation and control input saturation present on the USV. The stability is preserved under the user-defined prescribed performance scheme where model parameters are considered unknown. In addition, we utilize an optimization problem to obtain smooth and collision-free trajectories respecting the kinodynamic constraints. The algorithm is deployed on a USV and tested in real-world open water experiments.

We split the problem of control of a surface vehicle (a boat) into two layers: *motion planning*(or *guidance* as referred in some literature), which based on the user input and present environment, i.e. obstacles, space constraints, other agents, etc., generates a trajectory, and *trajectory tracking*(or *control*) that given the specified performance and robustness boundaries follows the trajectory.

Motion planning is one of the fundamental problems in robotics. In general, it is tasked with moving an agent or an object from one location to another location without colliding with the environment [91]. In this chapter we investigate kinodynamic sampling-based motion planning [92–94]. The advantage of kinodynamic motion planning is that produced trajectories respect the differential constraints imposed by the system dynamics. Moreover, in this chapter, we explore usage of B-splines [95] to generate smooth kinodynamic trajectories satisfying both the spatial and dynamical constraints. The B-splines have recently regained attention for motion primitives generation in fast and iterative schemes for quadrotors and other unmanned agents [96–98]. The trajectory tracking problem for surface vehicles or vessels has been broadly studied and a comprehensive literature review is available

in Subsection 1.1.1.

This chapter addresses two general challenges of PPC, namely, the control input constraints by providing the stability conditions, and the underactuation which requires modification of the original PPC scheme. Furthermore, we pose an optimization problem with respect to spatial and dynamical constraints resulting in the dynamically feasible collision-free trajectories.

The contributions of this chapter can be summarized in the following:

- PPC with control input saturation;
- The proposed modifications to PPC scheme to enable its application to the underactuated 3-DoF USV, which consequently extends the proposed KDF framework to the class of underactuated USVs;
- Theoretical stability guarantees for PPC of the considered system with input constraints;
- The trajectory generation algorithm based on KDF-RRT that generates smoother trajectories thus requiring less effort for the funnel control

It is worth noting that prescribed performance has been used for underwater underactuated vehicles in [39]. However, in our work we develop a more general framework that also includes motion planning, considers input constraints, and provides real-world experimental results.

In this chapter, first we propose a general solution on how to handle the input saturation in prescribed performance control schemes. Then, we focus on the needed modifications that handle the underactuation for USV and present the stability theorem with input saturations. We present and discuss the results obtained in real-world open water experiments on Saab Piraya USV depicted on Fig. 4.1. Finally, we propose an optimization algorithm to generate smooth collision-free reference trajectories with respect to the given acceleration and velocity bounds and evaluate it in KDF scheme.

Figure 4.1: Piraya autonomous unmanned surface vehicle

4.1 Prescribed Performance Control With Input Saturation

In this section, we show how a system with control input saturation can be controlled with PPC and provide the stability theorem for it. Let us consider an input affine nonlinear system of the form

$$\dot{x}(t) = f(x(t)) + u(t) \tag{4.1}$$

where $x \in \mathbb{R}$, and let us define the control input $u(t)$ with saturation as

$$u(t) = \begin{cases} v(t) & |v| \leq \bar{u} \\ \bar{u}\,\mathrm{sgn}(v(t)) & |v| > \bar{u} \end{cases} \tag{4.2}$$

where $\bar{u} > 0$, and $v(t)$ will be defined shortly. The goal is to track the reference $r(t)$ and keep the error $e(t) = x(t) - r(t)$ inside of a prescribed funnel. Then the normalized error $\xi(t)$ with respect to a performance function $\rho(t)$ is

$$\xi(t) = \frac{e(t)}{\rho(t)} \tag{4.3}$$

and the transformed error $\varepsilon(t)$ is

$$\varepsilon(t) = \frac{1}{2} \ln \frac{1 + \xi(t)}{1 - \xi(t)} \tag{4.4}$$

The controller is designed as in (4.2) and

$$v(t) = -k\varepsilon \tag{4.5}$$

where $k > 0$. Furthermore, let us assume that there exists a constant $\bar{F} > 0$ such that

$$\bar{F} \geq |f(x) - \dot{r} - \dot{\rho}\xi|, \quad \forall t \geq 0. \tag{4.6}$$

Then, we can state the stability result in the following theorem.

Theorem 4.1. *Consider the system* (4.1) *under the proposed controller with saturation* (4.2) *and* (4.5)*. If the following assumption hold*

$$\bar{u} \geq \bar{F} \tag{4.7}$$

where $\bar{F}$ *is as in* (4.6)*,* $\bar{u}$ *is the input constraint, then it holds that* $|e(t)| < \rho(t)$ *and all closed-loop signals are bounded, for all* $t \geq 0$*.*

Proof. The first part of the proof follows the argument as in Theorem 3.1 in which it is shown that there exist a local solution such that $\xi(t) \in (-1, 1)$ for a time interval $t \in [0, \tau_{max})$, and is therefore omitted. The second part shows that the normalized error ξ remains in the compact subset of $(-1, 1)$, which is achieved by considering a

candidate Lyapunov function $V = \frac{1}{2}\varepsilon^2$. The derivative of the candidate Lyapunov function is

$$\dot{V} = \varepsilon\dot{\varepsilon} = \varepsilon\rho^{-1}s(u + f(x) - \dot{r} - \dot{\rho}\xi) \tag{4.8}$$

where $s = \frac{1}{1-\xi^2}$ is the derivative of the transformation as defined in Eq. (2.32) and note that for $\xi \in (-1, 1)$ it holds that $\rho^{-1}s = |\rho^{-1}s|$.

Choose $v(t) = -k\varepsilon$ and consider the two cases when (a) $|v| \leq \bar{u}$ and (b) $|v| > \bar{u}$.

(a) Let $|v| \leq \bar{u}$. Then $u = v = -k\varepsilon$ and

$$\begin{aligned}\dot{V} &= -k\varepsilon^2\rho^{-1}s + \varepsilon\rho^{-1}s(f(x) - \dot{r} - \dot{\rho}\xi) \\ &\leq -k\varepsilon^2|\rho^{-1}s| + |\varepsilon||\rho^{-1}s|\bar{F}\end{aligned}$$

Thus,

$$\dot{V} \leq 0, \quad \forall|\varepsilon| \geq \frac{\bar{F}}{k}$$

and by Theorem 2.4, ε is ultimately bounded

$$|\varepsilon| \leq \bar{\varepsilon} := \max\left\{|\varepsilon(0)|, \frac{\bar{F}}{k}\right\}$$

(b) Let $|v| > \bar{u}$. Then $u = \bar{u}\,\mathrm{sgn}(-k\varepsilon) = -\bar{u}\,\mathrm{sgn}(\varepsilon)$ and

$$\begin{aligned}\dot{V} &= -\bar{u}\rho^{-1}s\varepsilon\,\mathrm{sgn}(\varepsilon) + \varepsilon\rho^{-1}s(f(x) - \dot{r} - \dot{\rho}\xi) \\ &= -\bar{u}\rho^{-1}s|\varepsilon| + \varepsilon\rho^{-1}s(f(x) - \dot{r} - \dot{\rho}\xi) \\ &\leq -\bar{u}|\varepsilon||\rho^{-1}s| + |\varepsilon||\rho^{-1}s|\bar{F} \\ &\leq -|\varepsilon||\rho^{-1}s|(\bar{u} - \bar{F})\end{aligned}$$

Then,

$$\dot{V} \leq 0, \quad \forall\varepsilon, \quad \text{if } \bar{u} \geq \bar{F}$$

and ε is bounded with its initial value $|\varepsilon| \leq \bar{\varepsilon} := |\varepsilon(0)|$.

Therefore, in both cases there exist a finite $\bar{\varepsilon}$ such that

$$|\xi(t)| \leq \bar{\xi} = \tanh\bar{\varepsilon} < 1$$

and the error $e(t)$ stays inside of the funnel for $t \in [0, \tau_{max})$. By the direct application of Theorem 2.3 we obtain that $\tau_{max} = \infty$ and all closed-loop signals are bounded for $t \geq 0$. □

To better explain the intuition behind the proof of Theorem 4.1, let us consider the illustration on Fig. 4.2 for a system in $\mathbb{R}^2$. From the stability proof we obtain that the trajectories in the non-saturated case will stay ultimately bounded with some $\bar{\varepsilon}$ which is depicted on the first and second illustration. Moreover, if the control input

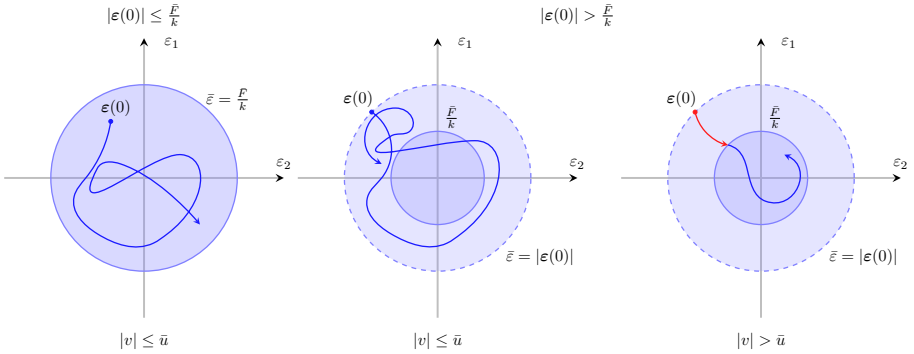

Figure 4.2: 2D illustration of the trajectories of $\boldsymbol{\varepsilon}$ for the different considered cases. Note that the radii of the balls between the cases are not drawn to scale. The first two illustrations correspond to the case of ultimately bounded trajectories, while the third corresponds to the saturated case when the trajectories (in red) will decrease until the control input stops being saturated. In the first illustration the initial state is inside of the ball of radius $\frac{\bar{F}}{k}$, while in the second two the initial state is outside.

is saturated and assumption (4.7) holds, the Lyapunov function is asymptotically decreasing while the input is saturated. This is depicted on the third illustration. Because $|\varepsilon|$ is also decreasing, eventually, the control input $|v|$ will decrease below the input constraint and we will again have the first case when the control is ultimately bounded. Note that the case when $|\boldsymbol{\varepsilon}(0)| \leq \frac{\bar{F}}{k}$ and $|v| > \bar{u}$ is not possible because $k|\varepsilon(t)| = |v| > \bar{u} \geq \bar{F} \geq k|\varepsilon(0)|$ is a contradiction for $t = 0$. The control input, however, can become saturated during the transient response which corresponds to $\varepsilon(t)$ being at the boundary of the inner ball of radius $\frac{\bar{F}}{k}$ but it will remain inside of the ball.

4.2 Prescribed Performance for USV

We consider the boat USV with one rotating thruster at the rear. The model is described in more details in Subsection 2.3.2, and we state it here for completeness.

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{b}} = \boldsymbol{R}_{z,\psi}(\psi_b)\boldsymbol{\nu}_{\boldsymbol{b}} \tag{4.9a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_{\boldsymbol{b}} + \boldsymbol{C}(\boldsymbol{\nu}_{\boldsymbol{b}})\boldsymbol{\nu}_{\boldsymbol{b}} + \boldsymbol{D}(\boldsymbol{\nu}_{\boldsymbol{b}})\boldsymbol{\nu}_{\boldsymbol{b}} = \boldsymbol{\tau}_{act} + \boldsymbol{\tau}_d \tag{4.9b}$$

where $\boldsymbol{\eta}_{\boldsymbol{b}} = [x, y, \psi]$, $(x, y) \in \mathbb{R}^2$ is the position in the world frame, $\psi \in [0, 2\pi)$ is the rotation of the boat in the world frame, $\boldsymbol{\nu}_{\boldsymbol{b}} = [u, v, r]$ is the velocity in the body frame, where u, v, r are forward velocity (surge), lateral velocity (sway) and angular velocity in yaw, respectively. $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu}_{\boldsymbol{b}})$ denotes the Coriolis and centripetal effects, $\boldsymbol{D}(\boldsymbol{\nu}_{\boldsymbol{b}})$ is the drag matrix. τ_d are unknown disturbances, and we also consider the drag induced by the relative velocity of the ship and

surrounding water as a disturbance. This leads to a linear damping matrix that is valid for low relative velocities, while at higher velocities nonlinear term dominates and discrepancies in the model are expected.

For the configuration with a thruster located at the rear

$$\boldsymbol{\tau}_{act} = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} F_T \cos(\alpha_r) \\ F_T \sin(\alpha_r) \\ \Delta_x F_T \sin(\alpha_r) \end{bmatrix}$$

where $\Delta_x > 0$.

The underactuation stems from the fact that only two control inputs F_T and α_r are available for control of the 3-DoF system. Moreover, two elements of τ_{act} are linearly dependent, i.e. $Y \propto N$, thus limiting the controlled behaviour of the dynamical system. Given the aforementioned observations, the dynamics in (4.9) can be rewritten as

$$m\dot{u} = X_u u + mvr + X + \tau_{d,x} \tag{4.10a}$$
$$m\dot{v} = Y_v v - mur + Y + \tau_{d,y} \tag{4.10b}$$
$$I_z \dot{r} = N_r r + N + \tau_{d,\psi} \tag{4.10c}$$

where m is the mass of the USV, I_z is the moment of inertia around z-axis, and X_u, Y_v, N_r are negative hydrodynamic damping coefficients.

We consider the tracking control problem of given time-varying reference trajectories $\boldsymbol{p}_r = [p_{x,r}, p_{y,r}]^T : [0, \infty) \to \mathbb{R}^2$ for the position in 2D with prescribed performance. $\boldsymbol{p}_r$ are assumed smooth functions of time with bounded first and second derivatives.

In order to accommodate the PPC methodology for the underactuated USV, let us consider the errors between the position $\boldsymbol{p} = [p_x, p_y]^T$ and given reference $\boldsymbol{p}_r$ in $\mathbb{R}^2$ space

$$e_x = p_{x,r} - p_x,$$
$$e_y = p_{y,r} - p_y.$$

Then, we introduce the following transformation and define the distance error e_d and orientation error e_o as

$$e_d = \sqrt{e_x^2 + e_y^2}, \tag{4.11}$$
$$e_o = \frac{e_x}{e_d} \sin\psi - \frac{e_y}{e_d} \cos\psi = \sin\psi_e \tag{4.12}$$

where ψ_e is the angle between $\vec{e}_d = [e_x, e_y]$ and the orientation vector $\vec{o} = [\cos\psi, \sin\psi]$. that is defined as the unit vector representing the orientation. e_o can be understood as $e_o = \frac{\vec{e}_d}{\|\vec{e}_d\|} \cdot \vec{n}$, where $\vec{n} = [\sin\psi, -\cos\psi]^T$ is a unit vector orthogonal to the orientation vector $\vec{o}$ and $e_d = \|\vec{e}_d\|$. The considered transformation and errors in NED (North-East-Down) inertial frame are depicted on Figure 4.3.

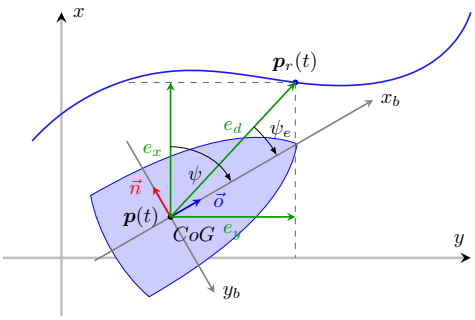

Figure 4.3: Sketch of the considered transformation.

The control objective is to guarantee that the distance and orientation errors evolve strictly within a funnel dictated by the corresponding exponential performance functions $\rho_d(t)$, $\rho_o(t)$, which is formulated as

$$|e_i(t)| < \rho_i(t), \ i \in \{d, o\} \tag{4.13}$$

for all $t \geq 0$, given the initial funnel compliance $|e_i(0)| < \rho_i(0)$, for $i \in \{d, o\}$. The adopted exponentially-decaying performance functions are

$$\rho_i(t) = (\rho_{i,0} - \rho_{i,\infty})e^{-l_i t} + \rho_{i,\infty}, \quad i \in \{d, o\}. \tag{4.14}$$

4.3 Reformulating the Model

We use the specific characteristics of the underactuated model of the USV in order to design prescribed performance control. Since $Y \propto N$ and we can effectively control one of them, the idea is to control the forward velocity dynamics u and rotation r, while keeping lateral velocity v below a specified threshold. Using the introduced transformation, the model can be rewritten as error dynamics

$$\dot{e}_d = -u\cos\psi_e + v\sin\psi_e + \dot{p}_{x,r}\cos(\psi - \psi_e) + \dot{p}_{y,r}\sin(\psi - \psi_e) \tag{4.15a}$$

$$\dot{e}_o = r\cos\psi_e + \frac{u}{e_d}\sin\psi_e\cos\psi_e + \frac{v}{e_d}\cos^2\psi_e \tag{4.15b}$$

$$- \frac{\dot{p}_{x,r}}{e_d}(\sin\psi_e\cos(\psi - \psi_e) - \sin\psi) - \frac{\dot{p}_{y,r}}{e_d}(\sin\psi_e\sin(\psi - \psi_e) + \cos\psi)$$

$$m\dot{u} = X_u u + mvr + X + \tau_{d,x} \tag{4.15c}$$

$$m\dot{v} = Y_v v - mur + \frac{1}{\Delta_x}N + \tau_{d,y} \tag{4.15d}$$

$$I_z\dot{r} = N_r r + N + \tau_{d,\psi} \tag{4.15e}$$

where we used the observation that $Y = \frac{1}{\Delta_x}N$. Furthermore, let the normalized errors $\xi(t)$ be defined as

$$\xi(t) = \rho(t)^{-1}e(t).$$

and the transformation normalized error $\xi(t)$ with a strictly increasing, bijective function $\mathrm{T} : (-1, 1) \to (-\infty, \infty)$

$$\mathrm{T}(\xi(t)) = \operatorname{atanh}(\xi(t)) = \frac{1}{2} \ln \frac{1 + \xi(t)}{1 - \xi(t)}. \tag{4.16}$$

The derivative of this transformation is $s = \frac{\mathrm{dT}(\xi)}{\mathrm{d}\xi} = \frac{1}{1-\xi^2}$. Note that we omitted subscripts for ξ, s and e as they will be introduced in the next section.

4.4 Control Design

Now, we describe the proposed control-design procedure.

PPC on distance error

We define the distance normalized error

$$\xi_d = \rho_d(t)^{-1} e_d(t) \tag{4.17}$$

where $\rho_d(t)$ is the prescribed performance function as defined in Eq. (4.14) such that $\rho_d(0) > |e_d(0)|$, and the transformation

$$\varepsilon_d = \mathrm{T}(\xi_d(t))$$

where T is given in (4.16). We design the forward velocity reference signal as

$$u_r = k_d \varepsilon_d \tag{4.18}$$

PPC on forward velocity error

Based on backstepping and in order to steer distance error to zero, we define the error

$$e_u = u - u_r$$

and introduce the performance function $\rho_u(t)$ with $\rho_u(0) > |e_u(0)|$ and the normalized error

$$\xi_u = \rho_u(t)^{-1} e_u(t) \tag{4.19}$$

and the transformation T gives us $\varepsilon_u = \mathrm{T}(\xi_u)$. Then, we can set

$$X = -k_u \varepsilon_u \tag{4.20}$$

PPC on orientation error

Following similar procedure as in first two parts, we define the normalized error and transformation

$$\xi_o = \rho_o(t)^{-1} e_o(t)$$
$$\varepsilon_o = \mathrm{T}(\xi_o(t))$$

with appropriate $\rho_o(t)$, $\rho_o(0) > |e_o(0)|$, which leads to the angular velocity reference

$$r_r = -k_o \varepsilon_o. \tag{4.21}$$

PPC on angular velocity error

Finnaly, we define angular velocity error

$$e_r = r - r_r,$$

its normalized version

$$\xi_r = \rho_r(t)^{-1} e_r$$

and the transformation $\varepsilon_r = \mathrm{T}(\xi_r)$. At the end, we can set

$$N = -k_r \varepsilon_r. \tag{4.22}$$

Remark 4.1. In the control design, we defined the orientation error as in (4.12) which can grow when the distance error e_d is approaching zero. This can be alleviated by designing an asymmetric performance function ρ_d such that $0 < \underline{\rho}_d < e_d(t) < \rho_d(t)$, where $\underline{\rho}_d$ is a positive constant. Furthermore, since e_d is by definition $e_d \geq 0$, implementing $\frac{1}{e_d}$ as $\frac{1}{e_d+\delta}$, with a small positive δ, for example, $\delta = 0.01$, will prevent e_o growing to infinity.

4.5 Handling Input Constraints

Let us introduce two virtual control inputs $u_\alpha, u_F \in \mathbb{R}$ to derive their saturated versions α_r and F_T, respectively. We define following two types of saturation functions

$$\sigma_{\alpha_r}(u_\alpha) = \begin{cases} u_\alpha, & |u_\alpha| \leq \bar{\alpha}_r \\ \bar{\alpha}_r \operatorname{sgn}(u_\alpha), & |u_\alpha| > \bar{\alpha}_r \end{cases}$$

and

$$\sigma_{F_T}(u_F) = \begin{cases} \bar{F}_T, & u_F > \bar{F}_T \\ u_F, & u_F \leq \bar{F}_T \\ 0, & u_F < 0 \end{cases}$$

Moreover, we set the virtual control inputs in relation to the derived X and N as following

$$\begin{aligned} X &= u_F \cos u_\alpha & X &= -k_u \varepsilon_u \\ N &= \Delta_x u_F \sin u_\alpha & N &= -k_r \varepsilon_r \end{aligned}$$

Therefore,

$$u_\alpha = \arctan\left(\frac{N}{\Delta_x X}\right) = \arctan\left(k_\alpha \frac{\varepsilon_r}{\varepsilon_u}\right)$$

where $k_\alpha = \frac{k_r}{\Delta_x k_u}$, and set

$$\alpha_r = \sigma_{\alpha_r}(u_\alpha) = \sigma_{\alpha_r}\left(\arctan\left(k_\alpha \frac{\varepsilon_r}{\varepsilon_u}\right)\right). \tag{4.23}$$

Furthermore,

$$u_F = \frac{X}{\cos\alpha_r} = -\frac{k_u \varepsilon_u}{\cos\alpha_r}$$

and

$$F_T = \sigma_{F_T}(u_F) = \sigma_{F_T}\left(-\frac{k_u \varepsilon_u}{\cos\alpha_r}\right). \tag{4.24}$$

This introduces another block within the closed-loop system and requires additional stability analysis. The prescribed performance control inputs are in $\mathbb{R}^2$ while the set of applicable control inputs is

$$\mathbb{U} = [0, \bar{F}_T] \times [-\bar{\alpha}_r, \bar{\alpha}_r], \tag{4.25}$$

where $\bar{F}_T > 0$ denotes the maximal thrust of the engine and $\bar{\alpha}_r \in (0, \pi/6]$ represents the maximum rudder position. In the next section, we show the conditions on the input constraints under which the stability is preserved.

4.6 Stability Analysis

We present the stability guarantees of the proposed control design with input constraints in the following theorem.

Theorem 4.2. *Consider the transformed USV dynamics* (4.15) *under the proposed control scheme* (4.17)-(4.22)*. If the following assumptions hold*

$$0 < \underline{F}_T \leq F_T \tag{4.26a}$$

$$\bar{F}_u \leq \bar{F}_T \cos\bar{\alpha}_r \tag{4.26b}$$

$$\bar{F}_r \leq \Delta_x \underline{F}_T \sin\bar{\alpha}_r \tag{4.26c}$$

$$|\psi_e(0)| < \frac{\pi}{2} \tag{4.26d}$$

where $\bar{F}_T$ *and* $\bar{\alpha}_r$ *are input constraints* (4.25)*,* $\underline{F}_T$ *is a positive constant,* $\bar{F}_u$ *and* $\bar{F}_r$ *are*

$$\left|\frac{X_u}{m}u + vr + \frac{\tau_{d,x}}{m} - \dot{u}_r - \dot{\rho}_u \xi_u\right| \leq \bar{F}_u$$

$$\left|\frac{N_r}{I_z}r + \frac{\tau_{d,\psi}}{I_z} - \dot{r}_r - \dot{\rho}_u \xi_u\right| \leq \bar{F}_r,$$

for $t \geq 0$*, then it holds that* $|e_d(t)| < \rho_d(t)$*,* $|e_o(t)| < \rho_o(t)$ *and all closed-loop signals are bounded, for all* $t \geq 0$*.*

Proof. The proof proceeds in four steps. First, we show the existence of a local solution such that $\xi_d(t)$, $\xi_o(t)$, $\xi_u(t)$, $\xi_r(t) \in (-1,1)^4$, for a time interval $t \in [0, \tau_{\max})$. Next, we show that the proposed control scheme retains the aforementioned normalized signals in compact subsets of $(-1,1)$, which leads to $\tau_{\max} = \infty$. In the third step, we show that the underactuated part of the system described with dynamics of v is bounded. Finally, we show that if the theorem assumptions hold the input constraints do not affect the stability of the system, thus completing the proof.

Part I: First, we consider the reduced version of the state vector $\boldsymbol{x} = [e_d, e_o, u, v, r]$ of the transformed error dynamics in (4.15) as $\boldsymbol{\chi} = [e_d, e_o, u, r]^T \in \mathbb{X} = \mathbb{R}^4$ without the underactuated part and we define an open set:

$$\Omega_\xi = \big\{(\boldsymbol{\chi}, t) \in \mathbb{X} \times [0, \infty) : \xi_d \in (-1,1), \xi_o \in (-1,1), \\ \xi_u \in (-1,1), \xi_r \in (-1,1)\big\}. \tag{4.28}$$

Note that the choice of the performance functions at $t = 0$ implies that $\xi_d(0)$, $\xi_o(0)$, $\xi_u(0)$, $\xi_r(0) \in (-1,1)^4$, implying that Ω is nonempty. By combining (4.15), (4.20), and (4.22), we obtain the closed-loop system dynamics $\dot{\boldsymbol{\chi}} = f_\chi(\boldsymbol{\chi}, t)$, where $f_\chi : \mathbb{X} \times [0, \tau_{\max})$ is a function continuous in t and locally Lipschitz in $\boldsymbol{\chi}$. Hence, the conditions of Theorem 2.2 are satisfied and we conclude that there exists a unique and local solution $\chi : [0, \tau_{\max}) \to \mathbb{X}$ such that $(\chi(t), t) \in \Omega$ for $t \in [0, \tau_{\max})$. Therefore, it holds that

$$\xi_d \in (-1,1) \tag{4.29a}$$
$$\xi_o \in (-1,1) \tag{4.29b}$$
$$\xi_u \in (-1,1) \tag{4.29c}$$
$$\xi_r \in (-1,1) \tag{4.29d}$$

for all $t \in [0, \tau_{\max})$. We next proceed to show that the normalized errors in (4.29) remain in compact subsets of $(-1,1)$. Note that (4.29) implies that that transformed errors ε_d, ε_o, ε_u, ε_r, are well-defined for $t \in [0, \tau_{\max})$.

Part II: Consider now the candidate Lyapunov function

$$V_o = \frac{1}{2}\varepsilon_o^2$$

Differentiating V_o along the local solution $\boldsymbol{\chi}(t)$ we obtain

$$\begin{aligned}\dot{V}_o =& \varepsilon_o s_o \rho_o^{-1}(\dot{e}_o - \dot{\rho}_o \xi_o) \\ =& \varepsilon_o s_o \rho_o^{-1}(r\cos\psi_e + \frac{u}{e_d}\sin\psi_e\cos\psi_e + \frac{v}{e_d}\cos^2\psi_e \\ & - \frac{\dot{p}_{x,r}}{e_d}(\sin\psi_e\cos(\psi - \psi_e) - \sin\psi) - \frac{\dot{p}_{y,r}}{e_d}(\sin\psi_e\sin(\psi - \psi_e) + \cos\psi) - \dot{\rho}_o\xi_o).\end{aligned}$$

Using $r = r_r + e_r$, (4.21), the boundedness of $\dot{\boldsymbol{p}}_r, \rho_o$ and (4.29) we obtain

$$\dot{V}_o \leq -k_o|s_o\rho_o^{-1}|\varepsilon_o^2\cos\psi_e + |s_o\rho_o^{-1}\varepsilon_o|\bar{F}_o$$

where

$$\begin{aligned}\bar{F}_o \geq &|e_r \cos\psi_e + \frac{u}{e_d}\sin\psi_e\cos\psi_e + \frac{v}{e_d}\cos^2\psi_e \\ &- \frac{\dot{p}_{x,r}}{e_d}(\sin\psi_e\cos(\psi-\psi_e) - \sin\psi) - \frac{\dot{p}_{y,r}}{e_d}(\sin\psi_e\sin(\psi-\psi_e) + \cos\psi) - \dot{\rho}_o\xi_o|.\end{aligned}$$

for all $t \in [0, \tau_{\max})$. Given the initial funnel compliance and (4.26d), the term $\cos\psi_e(0) > 0$ and from the Lyapunov derivative we can deduce the ultimate boundedness of ε_o for

$$|\varepsilon_o| \leq \bar{\varepsilon}_o = \max\left\{|\varepsilon_o(0)|, \frac{\bar{F}_o}{k_o}\right\}$$

and we get

$$|\xi_o(t)| \leq \bar{\xi}_o = \tanh\bar{\varepsilon}_o < 1.$$

Moreover, this also guarantees that $|\psi_r(t)| < \frac{\pi}{2}$, $\forall t \geq 0$.

Following the same procedure for e_d and considering a candidate Lyapunov function $V_d = \frac{1}{2}\varepsilon_d^2$, differentiating it and using $u = u_r + e_u$, (4.18), the boundedness of $\dot{\boldsymbol{p}}_r, \rho_d$ and (4.29) we obtain

$$\dot{V}_d \leq -k_d|s_d\rho_d^{-1}|\varepsilon_d^2\cos\psi_e + |s_d\rho_d^{-1}\varepsilon_d|\bar{F}_d$$

where $\bar{F}_d$ is a constant, independent of $\tau_{\max}$, satisfying

$$\bar{F}_d \geq |e_u\cos\psi_e - v\sin\psi_e + \dot{p}_{x,r}\cos(\psi-\psi_e) - \dot{p}_{y,r}\sin(\psi-\psi_e) - \dot{\rho}_d\xi_d|.$$

for all $t \in [0, \tau_{\max})$. This shows ultimate boundedness of ε_d, i.e.

$$\dot{V}_d < 0 \quad \text{when} \quad \frac{\bar{F}_d}{k_d} < |\varepsilon_d|,$$

thus ε_d is ultimately bounded by Theorem 2.4 for

$$|\varepsilon_d| \leq \bar{\varepsilon}_d = \max\left\{|\varepsilon_d(0)|, \frac{\bar{F}_d}{k_d}\right\}$$

for $t \in [0, \tau_{\max})$, and by employing the inverse of (4.16), we obtain

$$|\xi_d(t)| \leq \bar{\xi}_d = \tanh\bar{\varepsilon}_d < 1$$

Furthermore, consider a candidate Lyapunov function $V_u = \frac{1}{2}\varepsilon_u^2$. Differentiating we obtain

$$\begin{aligned}\dot{V}_u &= \varepsilon_u s_u\rho_u^{-1}\left(\frac{X_u}{m}u + vr + \frac{X}{m} + \frac{\tau_{d,x}}{m} - \dot{u}_r - \dot{\rho}_u\xi_u\right) \\ \dot{V}_u &\leq -k_u|s_u\rho_u^{-1}|\varepsilon_u^2 + |s_u\rho_u^{-1}\varepsilon_u|\bar{F}_u\end{aligned}$$

where we used (4.20) and

$$\bar{F}_u \geq \left| \frac{X_u}{m} u + vr + \frac{\tau_{d,x}}{m} - \dot{u}_r - \dot{\rho}_u \xi_u \right| \tag{4.30}$$

for all $t \in [0, \tau_{\max})$. This shows ultimate boundedness of ε_u for

$$|\varepsilon_u| \leq \bar{\varepsilon}_u = \max \left\{ |\varepsilon_u(0)|, \frac{\bar{F}_u}{k_u} \right\} \tag{4.31}$$

$$|\xi_u(t)| \leq \bar{\xi}_u = \tanh \bar{\varepsilon}_u < 1$$

Finally, following a similar procedure with $V_r = \frac{1}{2}\varepsilon_r^2$, using

$$\bar{F}_r \geq \left| \frac{N_r}{I_z} r + \frac{\tau_{d,\psi}}{I_z} - \dot{r}_r - \dot{\rho}_u \xi_u \right|. \tag{4.32}$$

for all $t \in [0, \tau_{\max})$, we conclude that

$$|\varepsilon_r| \leq \bar{\varepsilon}_r = \max \left\{ |\varepsilon_r(0)|, \frac{\bar{F}_r}{k_r} \right\} \tag{4.33}$$

$$|\xi_r(t)| \leq \bar{\xi}_r = \tanh \bar{\varepsilon}_r < 1 \tag{4.34}$$

which completes the second part of the proof.

Part III: In the third part, we consider the Lyapunov function

$$V_v = \frac{1}{2} m v^2$$

and by differentiating along the solution of (4.15d)

$$\begin{aligned} \dot{V}_v &= v(Y_v v - mur + c_Y N + \tau_{d,y}) \\ &= -k_v v^2 + v(mur - c_Y k_r \varepsilon_r + \tau_{d,y}) \\ &\leq -k_v |v|^2 + \bar{F}_v |v| \end{aligned}$$

where $k_v = -Y_v > 0$ due to the model properties, and

$$\bar{F}_v \geq |mur + c_Y k_r \bar{\varepsilon}_r + \tau_{d,y}|, \quad \text{for all } t \in [0, \tau_{\max})$$

Then,

$$|v| \leq \bar{v} = \max \left\{ |v(0)|, \frac{\bar{F}_v}{k_v} \right\}$$

is ultimately bounded.

Part IV: Finally, in the last part, we consider the input constraints. Since $\cos\alpha_r \in [\cos\bar{\alpha}_r, 1]$, therefore $\cos\alpha_r > 0, \forall\alpha_r \in [-\bar{\alpha}_r, \bar{\alpha}_r]$. Let us now, consider the saturation effects on the thrust (4.24)

$$F_T = \begin{cases} \bar{F}_T, & \text{for } \varepsilon_u < -\frac{1}{k_u}\bar{F}_T\cos\alpha_r \\ -\frac{k_u\varepsilon_u}{\cos\alpha_r}, & \text{for } -\frac{1}{k_u}\bar{F}_T\cos\alpha_r \leq \varepsilon_u < 0 \\ 0, & \text{for } \varepsilon_u \geq 0 \end{cases}$$

When $\varepsilon_u \geq 0$ we have a case in which the actual velocity u of the surface vehicle is greater than the reference velocity u_r, and a desired action is to decrease the velocity by reducing the thrust F_T. Since the considered vehicle can not go backwards in u slowing down is effectively achieved by setting the control input to zero and allowing the negative error e_u to grow again. However, in this case the stability is not compromised. For $-\frac{1}{k_u}\bar{F}_T\cos\alpha_r \leq \varepsilon_u < 0$ we have the unsaturated case as in the second part of the proof. For the saturated case, $\varepsilon_u < \tilde{\varepsilon}_u := -\frac{1}{k_u}\bar{F}_T\cos\alpha_r$, the applied force is $X = \bar{F}_T\cos\alpha_r = -k_u\tilde{\varepsilon}_u$, then $\dot{V}_u$ becomes

$$\dot{V}_u \leq -k_u|s_u\rho_u^{-1}|\varepsilon_u\tilde{\varepsilon}_u + |s_u\rho_u^{-1}\varepsilon_u|\bar{F}_u$$

Since $\varepsilon_u < \tilde{\varepsilon}_u < 0$, then

$$\begin{aligned} \dot{V}_u &\leq -k_u|s_u\rho_u^{-1}|\varepsilon_u\tilde{\varepsilon}_u + |s_u\rho_u^{-1}\varepsilon_u|\bar{F}_u \\ &= -k_u|s_u\rho_u^{-1}|(-|\varepsilon_u|)\tilde{\varepsilon}_u + |s_u\rho_u^{-1}\varepsilon_u|\bar{F}_u \\ &= |s_u\rho_u^{-1}\varepsilon_u|(k_u\tilde{\varepsilon}_u + \bar{F}_u) \leq 0 \end{aligned}$$

for $k_u\tilde{\varepsilon}_u + \bar{F}_u \leq 0$, which is valid because of the Assumption (4.26b), i.e., $\bar{F}_u \leq \bar{F}_T\cos\bar{\alpha}_r \leq \bar{F}_T\cos\alpha_r = -k_u\tilde{\varepsilon}_u$, the Lyapunov function is asymptotically stable during the saturated period and $|\varepsilon_u(t)|$ and $|\xi_u(t)|$ are upper bounded

$$\|\varepsilon_u\| \leq |\tilde{\varepsilon}_u|, \quad \text{for all } t \in [0, \tau_{max})$$

$$|\xi_u(t)| \leq \tilde{\xi}_u = \tanh\tilde{\varepsilon}_u < 1.$$

In the second part of the proof, using (4.23), we consider the saturation effects on the rudder angle

$$\alpha_r = \begin{cases} \arctan\left(k_\alpha\frac{\varepsilon_r}{\varepsilon_u}\right), & \text{for } |\arctan\left(k_\alpha\frac{\varepsilon_r}{\varepsilon_u}\right)| \leq \bar{\alpha}_r \\ -\bar{\alpha}_r\,\mathrm{sgn}(\varepsilon_r), & \text{for } |\arctan\left(k_\alpha\frac{\varepsilon_r}{\varepsilon_u}\right)| > \bar{\alpha}_r \end{cases}$$

where we used the fact that $\arctan(\cdot)$ is an odd function and that ε_u can only approach zero from the negative side. The stability of the unsaturated case is shown in the second part of the proof.

For the saturated case it holds that $N = -\Delta_x F_T \sin\bar{\alpha}_r \operatorname{sgn}(\varepsilon_r)$ then $\dot{V}_r$ becomes

$$\begin{aligned}
&\dot{V}_r|s_r\rho_r^{-1}|(-\varepsilon_r\Delta_x F_T \sin\bar{\alpha}_r \operatorname{sgn}(\varepsilon_r) + |s_r\rho_r^{-1}\varepsilon_r|\bar{F}_r)\\
&\quad = |s_r\rho_r^{-1}|(-|\varepsilon_r|\sin\bar{\alpha}_r\Delta_x F_T + |\varepsilon_r|\bar{F}_r)\\
&\quad = -|s_r\rho_r^{-1}\varepsilon_r|(\Delta_x F_T \sin\bar{\alpha}_r - \bar{F}_r)
\end{aligned}$$

and due to Assumptions (4.26a) and (4.26c), i.e., $\bar{F}_r \leq \Delta_x \underline{F}_T \sin\bar{\alpha}_r \leq \Delta_x F_T \sin\bar{\alpha}_r$, $\dot{V}_r \leq 0$ and ε_r is asymptotically stable during the saturated period and $|\varepsilon_r(t)|$ and $|\xi_r(t)|$ are bounded. This concludes the proof. □

Remark 4.2. As it has been already mentioned in the proof of Theorem 4.2, the considered surface vehicle is only capable of going forward (without relying on the drag or disturbances), thus $u \geq 0$. Therefore, it is not possible to reduce the forward velocity u when it is required by the reference (4.18) other than reducing the forward thrust or cutting it completely and relying on the drag forces. Because the model is unknown and no prediction is used this behaviour can occur which is an inherent property of the considered underactuated vehicle. Because of these reasons, Theorem 4.2 is not considering the non-controlled behaviour when $F_T = 0$ in which both X and N are zero. However, this problem is present, especially in the saturation effects on the rudder angle, where small values of ε_u cause α_r to be saturated while the actual orientation error might not require such action of the control input α_r. If one wants to consider these effects too, then the actuator model must be augmented with a part that depends on the velocities of the vehicle and surrounding water, and the position of rudder, which is outside of the scope of this work. Therefore, we restrict ourselves only to the case when the applied thrust is positive as in (4.26a).

Remark 4.3. The assumptions (4.26b) and (4.26c) can be intuitively explained as the guarantee that for the worst case of the applied control inputs X and N there exist enough energy to override the dynamics of the system. In a conservative observation, constants $\bar{F}_u$ and $\bar{F}_r$ can be understood as the upper bounds on the sum of four components that depend on the dynamics, disturbances, the derivative of the reference signal, in this case $\dot{u}_r$ and $\dot{r}_r$, respectively, and a term $l_i(\rho_{i,0} - \rho_{i,\infty})$, $i = u, r$, that prescribes the speed of convergence. Although we can tune, up to some degree, the latter two, the burden is on the upper bound of the dynamics which can cause these assumptions to be very conservative.

Remark 4.4. The requirement that $|\psi_e(0)| < \frac{\pi}{2}$ is needed to ensure the initial compliance and boundedness of the orientation error. Moreover, it is shown that ψ_e will remain bounded $|\psi_e(t)| < \frac{\pi}{2}$, for all $t \geq 0$, which means that the reference trajectory is always kept in front of the boat. This is a reasonable behaviour given the observations on the forward motion of the boat and Remark 4.2.

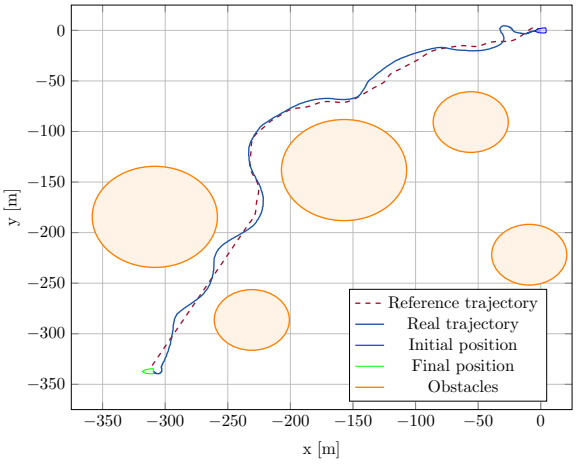

Figure 4.4: Top-view of an experimental run with KDF

4.7 Experimental Results

The overall scheme is tested in the real-world open water experiments. The reference trajectory is generated using KDF-RRT presented in [40]. The goal point is set approximately 450 meters away from the initial position with obstacles present. KDF-RRT takes into account the static obstacles and reference trajectory is generated with a safety buffer zone that is approximately 10m inflated zone around the obstacles can vary in the size.

The funnels are chosen to be static ($\rho_0 = \rho_\infty$ and $l = 0$) during the 3 minute long experiment because the desired error has the same performance criteria throughout the whole experiment. Moreover, they are set relatively loose to avoid saturating the control inputs too often. Thus, $\rho_d = 28$, $\rho_u = 25$, $\rho_o = \frac{\pi}{2}$, $\rho_r = 15$.

The top view of the experiment is depicted on Figure 4.4. We can observe that neglecting the deviation that occurred in the first moments of the experiment, the surface vehicle was able to follow the trajectory. The initial deviation occurred because the reference errors were relatively small to excite the control algorithm to produce the control thrust F_T so the boat was drifting due to the winds and open water currents. After few seconds, when the errors were bigger the control inputs were activated and the boat successfully recovered from the deviation. This, however can be alleviated by tightening the funnels.

The funnels are depicted on Figure 4.5, and several interesting phenomena can be observed. First, all signals are bounded, and moreover the orientation error is always less than 1, which corresponds to $|\psi_e(t)| < \frac{\pi}{2}$. The velocity error e_u is always nonpositive which is equivalent to $\varepsilon_u \leq 0$. The distance error, is of course always positive, but it is interesting to observe that it is kept relatively close to the funnel boundary during the experiment. This can be explained by the fact that loose funnels will activate the control thrust more and counteract the error when the error

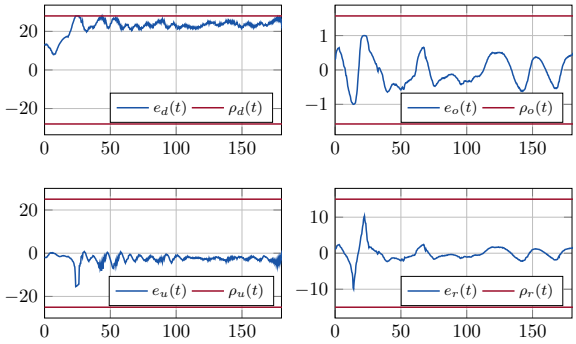

Figure 4.5: Funnels

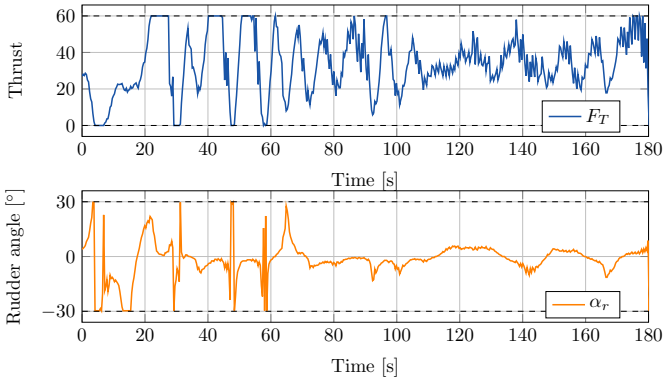

Figure 4.6: Control inputs

e_d is relatively large.

On Figure 4.6, the control inputs are presented and periods of saturation are visible. They occur especially in the first half of the experiment during which the boat was expected to recover from the initial deviation and subsequent overshoots. The most problematic behaviour is as expected when $F_T = 0$, which causes α_r to be saturated although the orientation error might not require that action. This is discussed in detail in Remark 4.2.

One of the reasons that was causing the unwanted saturations of the thrust F_T and then consequently the saturation of α_r, is the reference trajectory. The original KDF-RRT is not concerned with velocity and acceleration profiles because they can be successfully handled via funnel control for fully actuated agents. Therefore, although smooth, the reference trajectory could have significant perturbations in velocity, acceleration and jerk profiles. This could cause the agent, like the one considered in this chapter, to periodically accelerate and decelerate unnecessarily, which is visible on Figure 4.7. Thus, in the next section we propose a trajectory generation algorithm with jerk minimization to handle this issue.

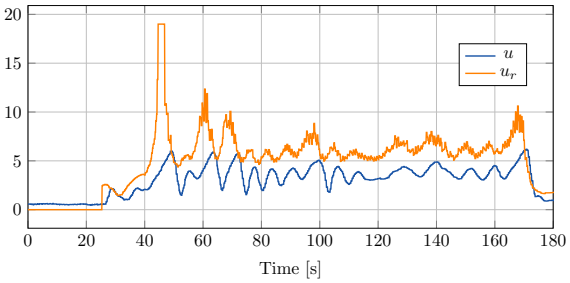

Figure 4.7: Forward velocity u and its reference u_r in time.

4.8 Kinodynamic Motion-Planning

In this part, we present Kinodynamic motion planning via Funnel control with Rapidly Exploring Random Trees (KDF-RRT) and propose enhancements with respect to the smoothness, velocity and acceleration profiles of the trajectory and safety of the agent near obstacles. KDF-RRT, introduced in [40], consists of RRT, that samples in the extended free space and obtains a path, and an algorithm for generating smooth time trajectories.

In this work, we consider cubic B-splines[95] and use some of their properties in order to generate the smooth trajectory. We pose the smoothening problem of the RRT obtained path $\{X_k\}_{k=0}^{N_X-1}$, $X_k \in \mathbb{R}^2$, as an optimization problem. The uniform B-splines are determined with $N = N_X + 4$ control points $\{q_k\}_{k=0}^{N-1}$, $q_k \in \mathbb{R}^2$, and $M = N + d + 1$ uniformly separated knots $t_k = k\Delta t$, $k = 4, ..., N$, where $d = 3$ is the degree of the curve. The rest of the knots are defined as $t_0 = t_1 = t_2 = t_3 = 0$ and $t_N = t_{N+1} = t_{N+2} = t_{N+3} = t_{N+4}$, to guarantee zero velocity and acceleration at the initial and finial position imposed with $q_k = X_0$, for $k = 0, 1, 2$ and $q_k = X_{N_X}$, for $k = N-3, N-2, N-1$. The goal is to minimize the distance from the parametric 2D curve $f(t_k)$ at each knot, which by using the properties for uniform B-splines, can be evaluated on a segment $t \in [t_k, t_{k+1})$ with the knowledge of any four consecutive control points $Q_k = [q_{k-3}, q_{k-2}, q_{k-1}, q_k]^T$, as

$$f(t) = \boldsymbol{u}^T M Q_k$$

with some abuse of notation, where $\boldsymbol{u} = [1, u, u^2, u^3]^T$ is the basis vector with $u = \frac{t-t_k}{t_{k+1}-t_k} \in [0, 1)$ and M is a fixed known matrix [99] independent of k

$$M = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 0 \end{bmatrix}.$$

Then the distance at t_k can be calculated as

$$\|f(t_k) - X_k\|^2 = \left\|\boldsymbol{u}^T(0) M Q_k - X_k\right\|^2 = \left\|m^T Q_k - X_k\right\|^2$$

where $m = \frac{1}{6}[1, 4, 1, 0]^T$.

We are interested to also keep the trajectory inside of the prespecified velocity v_{max} and acceleration bounds a_{max}. Using the derivative properties of B-splines we can obtain

$$v_k = \frac{d(q_k - q_{k-1})}{t_{k+d} - t_k} = \frac{q_k - q_{k-1}}{\Delta t} \tag{4.35}$$

$$a_k = \frac{(d-1)(v_k - v_{k-1})}{t_{k+d-1} - t_k} = \frac{v_k - v_{k-1}}{\Delta t} \tag{4.36}$$

for uniform knots. Moreover, it is beneficial to obtain as smooth curve as possible, thus we include third derivative (jerk) minimization in the cost function as

$$\|j_k\| \Delta t = \frac{1}{\Delta t} \|a_k - a_{k-1}\| \Delta t = \left\|b^T Q_k\right\| \tag{4.37}$$

where $b = [-1, 3, -3, 1]^T$. Since we are optimizing over a fixed number of control points and uniformly separated knots we can include Δt in the optimization problem in order to minimize the time duration as well.

Due to the fact that four consecutive control points create a convex hull around a spline segment, the collision avoidance can be done using linear separation. The obstacles are modeled as n-sided convex polygons with vertices at $p_{jl} \in \mathbb{R}^2$ of a j-th polygon, $l = 1, ..., n$, compactly written as $P_j = [p_{j1}, ...p_{jn}]$. The existence of a line that separates the two sets is based on the existence of $\boldsymbol{h}_{ij} \in \mathbb{R}^2$ and $d_{ij} \in \mathbb{R}$, where $i = 0, ..., N-4$ denotes i-th convex hull around a spline segment determined with Q_i, such that

$$\boldsymbol{h}_{ij}^T Q_i > d_{ij} \mathbf{1}_4, \tag{4.38}$$

$$\boldsymbol{h}_{ij}^T P_j < d_{ij} \mathbf{1}_n, \tag{4.39}$$

for all i, j, where $\mathbf{1}_n \in \mathbb{R}^n$ denotes a unit vector. The linear separation idea is depicted on Fig. 4.8.

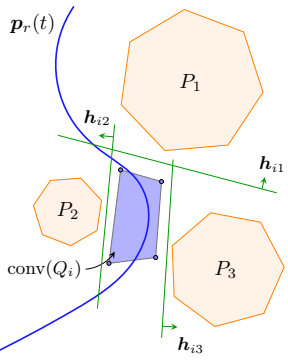

Figure 4.8: The convex hull is linearly separable from the obstacles.

Finally, we can state the nonlinear optimization problem

Problem 4.1 (Trajectory generation)**.**

$$\min_{\Delta t, q_k, \boldsymbol{h}_{ij}, d_{ij}} \sum_{k=3}^{N-4} w_1 \left\| m^T Q_k - X_{k-2} \right\|^2 + w_2 \left\| b^T Q_k \right\|^2 + w_3 \Delta t$$

subject to

$$\begin{aligned}
q_k &= X_0, && k \leq 2, && \text{(4.40a)}\\
q_k &= X_{N-5}, && k \geq N-3, && \text{(4.40b)}\\
\|q_k - q_{k-1}\| &\leq v_{max} \Delta t, && k \geq 1, && \text{(4.40c)}\\
\|q_k - 2q_{k-1} - q_{k-2}\| &\leq a_{max} \Delta t, && k \geq 2, && \text{(4.40d)}\\
\boldsymbol{h}_{ij}^T Q_i &> d_{ij} \mathbf{1}_4, && \forall i, j && \text{(4.40e)}\\
\boldsymbol{h}_{ij}^T P_j &< d_{ij} \mathbf{1}_n, && \forall i, j && \text{(4.40f)}
\end{aligned}$$

Where the hard constraints (4.40a) and (4.40b) denote the initial and final conditions, (4.40c)-(4.40d) velocity and acceleration constraints, (4.40e)-(4.40f) linear separability conditions as explained previously. Based on the weight choices $w_i \geq 0$, $i = 1, 2, 3$, in Problem 4.1, we can prioritize between the three objectives, namely, fitting the curve to the RRT obtained path, minimizing the jerk and minimizing the time, respectively. Note that we shifted indices of Q_k such that the control points are not optimized over the fixed control points determining the initial and final values of the curve.

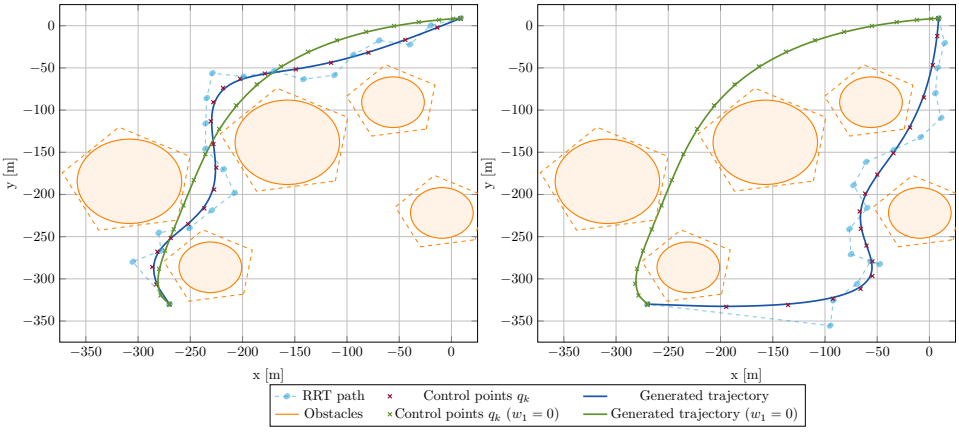

Figure 4.9: The figure shows two runs of the trajectory generation algorithm with RRT and the same setup of obstacles as in the real-world experiments. Two different RRT paths were obtained for comparison. The trajectory is then generated using the optimization problem in Problem 4.1. Moreover, we show the result of the optimization without RRT points with $w_1 = 0$ in green. Note that interpolating through the RRT points only would result in a hard trajectory to follow.

Also note that, it is possible to choose $w_1 = 0$ and thus completely avoid usage of RRT path. However, this option results with significantly higher execution time in the simulations, therefore making the RRT path a useful prior for the curve generation. A simulation example of the presented trajectory generation algorithm with $v_{max} = 10$, $a_{max} = 2$ is given on Fig. 4.9.

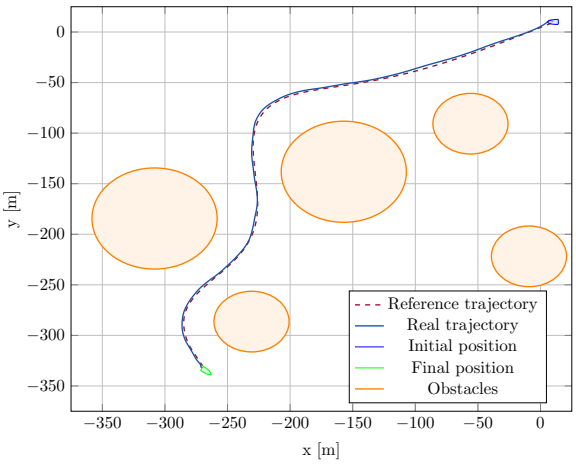

Figure 4.10: Simulated top view

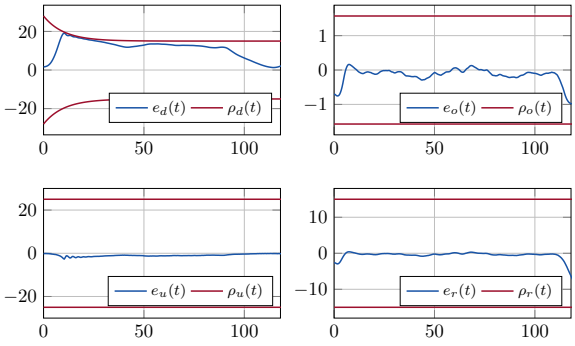

Figure 4.11: Funnels

The execution of the trajectory as well as the funnels and control inputs, with forward velocity comparison are given on Fig. 4.10-4.13. In these simulations, for the sake of fair comparison we added white noise and kept the funnels same, except $\rho_d(t)$ which has been tightened. It can be clearly observed that the tracking and utilization of the control inputs is considerably better. The inputs are only saturated in the beginning and in the end of the simulation run.

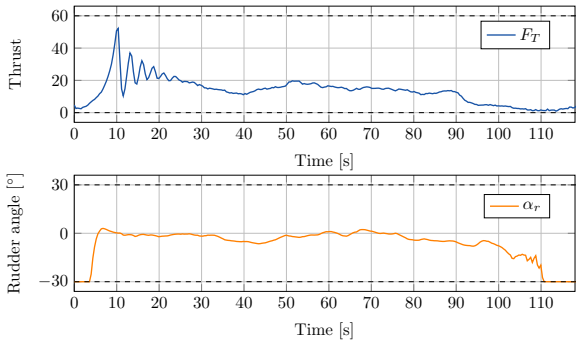

Figure 4.12: Control inputs

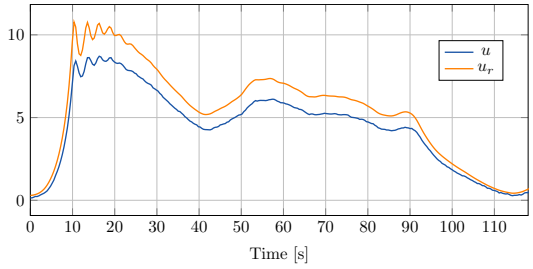

Figure 4.13: Forward velocity

4.9 Conclusion

In this chapter, we proposed an improved kinodynamic motion planning via funnel control (KDF). We proved the stability under input constraints and system underactuation. The presented framework is tested in real-world experiments. Moreover, a comparison with simulation results obtained using the proposed trajectory generation algorithm is provided. In the future work, it may be beneficial to explore the effect of the funnel size and performance functions on the behaviour of the vehicle. Moreover, we plan to redesign the motion planning procedure with B-splines as an iterative online scheme suitable for dynamic obstacles and environments.

Chapter 5

Multi-Agent Rendezvous with Distributed Predictive Control

In the previous two chapters, we elaborated on our developed trajectory tracking algorithms for the considered underactuated unmanned vehicles. The next two chapters are devoted to coordination and control of these agents considered as a multi-agent system. As motivated in Chapter 1, this is a very active research area that seeks for solutions to enable more autonomous behavior and operation of unmanned agents and multi-agent systems as a whole. Because multi-agent systems are composed of agents with embedded computing and communication units, a distributed control scheme is the most common control approach to these types of problems. A comprehensive literature review is available in Subsection 1.1.2.

This chapter addresses the rendezvous problem of a multi-agent system, that is formulated in general as Problem 1.2. We consider rendezvous control through Distributed MPC (DMPC), where the agents use an aperiodic exchange of information to negotiate and update their rendezvous point. The agents achieve cooperation through the iterative updates of the shared rendezvous point. The exchange of information occurs only when it is necessary to maintain the feasibility of the control action, thus reducing the necessary communication between the agents. The control algorithm is applied to nonlinear heterogeneous agents with state and input constraints, and tested and evaluated in simulation on an example of a UAV landing on a USV.

The main contributions presented in this chapter are published in [44] with addition of a stability proof and are outlined here as follows:

- We present a distributed rendezvous algorithm that enables the aperiodic communication between the agents based on the deviations from the predicted trajectory, thus eliminating unnecessary communication.
- Moreover, we synthesize the time-varying distributed terminal sets for tracking that depend on the rendezvous point. These terminal sets are the main ingredient in the recursive feasibility proof.

- Finally, we prove that the proposed algorithm guarantees recursive feasibility and stability.

The chapter is organized as follows. First, we present the distributed optimal control problem formulation and the rendezvous algorithm with triggering conditions. Then, we state the feasibility and stability theorems and provide proofs for them. Finally, we describe the models and their constraints which are used to generate the simulation results.

5.1 Distributed Optimal Control Problem

We consider M agents with nonlinear dynamics and additive disturbances:

$$\begin{aligned} \dot{\boldsymbol{x}}_i(t) &= f_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)) + \boldsymbol{w}_i(t), \\ \boldsymbol{y}_i(t) &= C_i \boldsymbol{x}_i(t), \end{aligned} \tag{5.1}$$

for $t \geq t_0$, where for each $i = 1, ..., M$, the state vector $\boldsymbol{x}_i(t) \in \mathbb{R}^{n_i}$ is measurable, $\boldsymbol{u}_i(t) \in \mathcal{U} \subseteq \mathbb{R}^{m_i}$ is the control input, the output $\boldsymbol{y}_i(t) \in \mathbb{R}^p$ consists of the states we aim to control for the rendezvous, $\boldsymbol{w}_i(t) \in \mathcal{W} \subseteq \mathbb{R}^{n_i}$ is the additive bounded disturbance, and $t_0 \in \mathbb{R}$ is the initial time.

The standard MPC assumptions are needed as stated in Assumption 2.1, which we state here again for completeness.

Assumption 5.1. It is assumed that

(i) the function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is twice continuously differentiable and $f(\mathbf{0}, \mathbf{0}) = \mathbf{0}$;

(ii) $\mathcal{U} \subseteq \mathbb{R}^m$ is compact, convex and $\mathbf{0} \in \mathbb{R}^m$ is contained in $\mathcal{U}$;

(iii) the system in (2.35) has a unique solution for any initial condition $\boldsymbol{x}_0 \in \mathbb{R}^n$, any piecewise continuous and right-continuous control $\boldsymbol{u} : [t_0, \infty) \to \mathcal{U}$, and any disturbance $\boldsymbol{w} : [t_0, \infty) \to \mathcal{W}$;

(iv) for the linearized system around the origin without disturbances, i.e., $\dot{\boldsymbol{x}} = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)$, where $A = \frac{\partial f}{\partial \boldsymbol{x}}(\mathbf{0}, \mathbf{0})$ and $B = \frac{\partial f}{\partial \boldsymbol{u}}(\mathbf{0}, \mathbf{0})$, the pair (A, B) is stabilizable;

(v) for the linearized dynamics around the origin, there exists a matrix K such that $A_k = A + BK$ is a stable Hurwitz matrix.

Remark 5.1. Note that the requirement $f_i(\mathbf{0}, \mathbf{0}) = \mathbf{0}$ is not restricted to the origin, but can be shifted to any equilibrium $(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)$, as well as the linearization in (iv).

Let $\hat{\boldsymbol{x}}_i(s;t_k), \hat{\boldsymbol{y}}_i(s;t_k)$ be the nominal state trajectory and output, respectively, calculated at time instant t_k given by

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}_i(s;t_k) &= f_i(\hat{\boldsymbol{x}}_i(s;t_k), \boldsymbol{u}_i(s;t_k)), \\ \hat{\boldsymbol{y}}_i(s;t_k) &= C_i \hat{\boldsymbol{x}}_i(s;t_k), \end{aligned} \tag{5.2}$$

for $s \in [t_k, t_k + T]$.

The control objective is to steer the relevant states of every agent $\boldsymbol{y}_i$ to a rendezvous point $\boldsymbol{\theta} \in \mathbb{R}^p$ in finite time. The set of all admissible rendezvous points is denoted with $\Theta \subseteq \mathbb{R}^p$.

Let us define a set $\mathcal{Z}_i(\boldsymbol{\theta})$ for each agent i and argument $\boldsymbol{\theta} \in \mathbb{R}^p$ with a tuple $(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i, \bar{\boldsymbol{y}}_i)$ such that

$$\mathcal{Z}_i(\boldsymbol{\theta}) = \{(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i, \bar{\boldsymbol{y}}_i) \in \mathbb{R}^{n_i+m_i+p} : \mathbf{0} = f_i(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i), \bar{\boldsymbol{y}}_i = C_i \bar{\boldsymbol{x}}_i = \boldsymbol{\theta}\}$$

Assumption 5.2. There exists a non-empty compact and convex set $\Theta \subseteq \mathbb{R}^p$ such that $\forall \boldsymbol{\theta} \in \Theta$, we have $\mathcal{Z}_i(\boldsymbol{\theta}) \neq \emptyset$ for all i.

Considering the motivating application, one can think of the set Θ as an inflated convex set in the plane of the USV landing platform that covers the unoccupied space that UAV and USV can reach.

By this assumption, it is also assumed that there exists an equilibrium for which the output reference $\boldsymbol{\theta}$ is attained for each agent. Moreover, such an equilibrium can be explicitly found with a given θ by the following linear mappings $H_{x_i} \in \mathbb{R}^{p \times n_i}$, $H_{u_i} \in \mathbb{R}^{p \times m_i}$

$$\bar{\boldsymbol{x}}_i = H_{x_i} \boldsymbol{\theta}, \quad \bar{\boldsymbol{u}}_i = H_{u_i} \boldsymbol{\theta}. \tag{5.3}$$

The following assumption is made to ensure that a such rendezvous point is reachable (in a similar manner to Assumption 2. in [100]):

Assumption 5.3. The time planning horizon T is long enough to reach at least one $\boldsymbol{\theta}$ in the rendezvous set Θ.

We choose the cost function to penalize the deviations of the system trajectories from the desired terminal steady-state $(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i, \bar{\boldsymbol{y}}_i)$:

$$\begin{aligned} J_i(\hat{\boldsymbol{x}}_i(t_k), \boldsymbol{u}_i(t_k), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) &= \|\hat{\boldsymbol{x}}_i(t_k + T; t_k) - \bar{\boldsymbol{x}}_i\|_{P_i}^2 \\ &+ \int_{t_k}^{t_k+T} \|\hat{\boldsymbol{x}}_i(s;t_k) - \bar{\boldsymbol{x}}_i\|_{Q_i}^2 + \|\boldsymbol{u}_i(s;t_k) - \bar{\boldsymbol{u}}_i\|_{R_i}^2 \, ds, \end{aligned} \tag{5.4}$$

where Q_i, R_i, P_i are positive definite weighting matrices, $T > 0$ is the time duration of prediction horizon.

Note that this formulation is a bit different from the standard tracking MPC formulations (see e.g. [101]), because of Assumption 5.2 that such a tuple $(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i, \bar{\boldsymbol{y}}_i)$ exists and is attainable.

Moreover, a standard formulation of distributed MPC for rendezvous (or consensus) as in [50], assumes exchange of predicted trajectories of the agents that are then utilized in the cost instead of a terminal steady-state. These algorithms exhibit very good convergence properties, the rendezvous location is computed online and it is not needed to be specified *a priori*, but they also require a significant usage of the communication channel and data, that could be a disadvantage in some practical multi-agent scenarios.

Before we formulate the distributed optimal control problem for rendezvous we will present a lemma on the local invariant terminal sets around a steady-state that is formulated following the ideas of [51], [102], [54].

Lemma 5.1. *For the nominal system* (5.2)*, if Assumption 5.1 holds, then there exists a positive constant* $\alpha_i \in (0, \bar{\alpha}_i]$*, a matrix* $P_i = P_i^T \succ 0$*, and a local state feedback control law* $\kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) = \bar{\boldsymbol{u}}_i + K_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \in \mathcal{U}_i$ *for a steady-state* $\bar{\boldsymbol{x}}_i$*, satisfying*

$$\frac{\partial V_{f,i}}{\partial \boldsymbol{x}_i}^T f_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i, \kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)) \leq -\frac{1}{2} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{Q_i^*}^2$$

for all $\boldsymbol{x}_i \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \alpha_i)$*, where* $V_{f,i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i) = \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i}^2$*,* $Q_i^* = Q_i + K_i^T R_i K_i$ *and the terminal set*

$$\mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \alpha_i) = \left\{ \boldsymbol{x}_i \in \mathbb{R}^{n_i} : V_{f,i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i) \leq \alpha_i^2 \right\}. \tag{5.5}$$

The proof is available in Appendix 5.A. From the proof of Lemma 5.1 it immediately follows how $\bar{\alpha}_i$ can be obtained.

Corollary 5.1. *(Calculation of* $\bar{\alpha}_i$*) The upper bound* $\bar{\alpha}_i$ *on* α_i*, can be calculated by solving the following optimization problem*

$$\bar{\alpha}_i = \max_{\boldsymbol{x}_i} \quad (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \tag{5.6a}$$

$$s.t. \quad \frac{16}{\lambda_{\min}^2(Q_{P_i})} \phi_i(\boldsymbol{x}_i)^T P_i \phi_i(\boldsymbol{x}_i) \leq (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \tag{5.6b}$$

$$\bar{\boldsymbol{u}}_i + K_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \in \mathcal{U}_i. \tag{5.6c}$$

where $\phi_i(\boldsymbol{x}_i) = f_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i, \kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)) - A_{k,i}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)$*, and* $Q_{P_i} = P_i^{-\frac{1}{2}} Q_i^* P_i^{-\frac{1}{2}}$.

Now, we can formulate the distributed optimal control problem with respect to our objective.

Problem 5.1. At time t_k with initial states $\boldsymbol{x}_i(t_k)$, $i = 1, ..., M$, and given reference $\boldsymbol{\theta}(t_k)$, the distributed optimal control problem is formulated as

$$\min_{\boldsymbol{u}_i(\cdot), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i} J_i(\hat{\boldsymbol{x}}_i(t_k), \boldsymbol{u}_i(\cdot), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \tag{5.7a}$$

subject to

$$\dot{\hat{\boldsymbol{x}}}_i(s;t_k) = f_i(\hat{\boldsymbol{x}}_i(s;t_k), \boldsymbol{u}_i(s;t_k)), \;\; s \in [t_k, t_k+T]\,, \tag{5.7b}$$
$$\hat{\boldsymbol{y}}_i(s;t_k) = C_i\hat{\boldsymbol{x}}_i(s;t_k), \tag{5.7c}$$
$$\hat{\boldsymbol{x}}_i(s;t_k) \in \mathcal{X}_i, \tag{5.7d}$$
$$\boldsymbol{u}_i(s;t_k) \in \mathcal{U}_i, \tag{5.7e}$$
$$\bar{\boldsymbol{x}}_i = H_{x_i}\boldsymbol{\theta}(t_k), \tag{5.7f}$$
$$\bar{\boldsymbol{u}}_i = H_{u_i}\boldsymbol{\theta}(t_k), \tag{5.7g}$$
$$\hat{\boldsymbol{x}}_i(t_k+T;t_k) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \alpha_i), \tag{5.7h}$$

for agents $i=1,...,M$. For the initial time t_0, $k=0$, the agents minimize the cost (5.7a) subject to (5.7b–5.7h) for a given $T > 0$.

5.2 Event-Triggered DMPC Rendezvous Algorithm

The distributed optimal control problem stated in Problem 5.1 depends on $\boldsymbol{\theta}(t_k)$ which is the rendezvous point in the subset of the output space $\mathbb{R}^p$ as stated in Assumption 5.2. Before we present the algorithm, we need to define how $\boldsymbol{\theta}(t_k)$ is going to be initialized and updated.

The rendezvous point $\boldsymbol{\theta}(t_k)$ at $k = 0$ can be initialized as a weighted average of the initial agent positions in the output space

$$\boldsymbol{\theta}(t_0) = \frac{1}{M}\sum_{i=1}^{M} c_i\boldsymbol{y}_i(t_0), \;\text{ s.t. }\; \frac{1}{M}\sum_{i=1}^{M} c_i = 1, c_i \geq 0, \tag{5.8}$$

where M is the number of agents.

We assume that there exists c_i, $i = 1,...,M$ such that $\boldsymbol{\theta}(t_0) \in \Theta$ according to Assumption 5.2. If the agents are operating in an unconstrained and obstacle-free output space, then any c_i will result with $\boldsymbol{\theta}(t_0) \in \Theta$. If this is not the case, then an admissible c_i would need to be determined by another layer of the optimization taking into account output-space constraints of all agents. Moreover, an interesting topic to investigate in the future work can include the conditions such that $\boldsymbol{\theta}(t_k)$ remains in a constrained output space Θ_c.

Let us denote the terminal output offset term V_o as

$$\begin{aligned} V_o = V_o(\hat{\boldsymbol{y}}_i, \boldsymbol{\theta}) &= V_o(\hat{\boldsymbol{y}}_i(t_k+T;t_k), \boldsymbol{\theta}(t_k)) \\ &= \|\hat{\boldsymbol{y}}_i(t_k+T;t_k) - \boldsymbol{\theta}(t_k)\|_2^2\,. \end{aligned} \tag{5.9}$$

After initialization, agent i updates $\boldsymbol{\theta}(t_k)$ according to the rule

$$\boldsymbol{\theta}(t_{k+1}) = \begin{cases} \boldsymbol{\theta}(t_k) & V_o \leq \varepsilon \\ \boldsymbol{\theta}(t_k) - \eta\boldsymbol{v}_\theta(t_k) & V_o > \varepsilon \end{cases} \tag{5.10}$$

where η and ε are tuning parameters and $\boldsymbol{v}_\theta(t_k)$ is defined as:

$$\boldsymbol{v}_\theta(t_k) = \frac{\partial V_o}{\partial \boldsymbol{\theta}(t_k)} \left\| \frac{\partial V_o}{\partial \boldsymbol{\theta}(t_k)} \right\|_2^{-1}. \tag{5.11}$$

Parameter η is a step size that must be chosen as a small value, in order to avoid overshooting, and it quantifies the correction of $\boldsymbol{\theta}$ in the output space. The rendezvous algorithm is presented as Algorithm 1.

Algorithm 1 Event-triggered DMPC Rendezvous

Require: prediction horizon T; sampling period δ; weighting matrices Q_i, R_i, P_i; initial states $\boldsymbol{x}_{i,0}$ at time t_0 for each agent $i = 1, ..., M$; $k = 0$; c_i, $\boldsymbol{\theta}(t_0)$ according to (5.8) and parameters η and ε;

1: **for** each agent $i = 1, .., M$ **do**
2: **if** new data message received **then**
3: download $\boldsymbol{\theta}(t_k)$
4: $\hat{\boldsymbol{u}}_i^*, \hat{\boldsymbol{y}}_i^* \leftarrow$ solve optimization problem (5.7)
5: **if** $V_o(\hat{y}_i(t_k + T; t_k), \theta(t_k)) > \varepsilon$ **then** ▷ Rendezvous condition
6: $\boldsymbol{\theta}(t_{k+1}) \leftarrow \boldsymbol{\theta}(t_k) - \eta \boldsymbol{v}_\theta(t_k)$
7: send data message $\{\boldsymbol{\theta}(t_{k+1})\}$ to other agents
8: **if** $\|y_i(t_k) - \theta(t_k)\| > \varepsilon$ **then** ▷ Stopping condition
9: apply $\hat{\boldsymbol{u}}_i^*(t_k; t_k)$
10: $k \leftarrow k + 1$

Remark 5.2. If the rendezvous condition at line 5 in Algorithm 1 is not satisfied, the only information that is sent from an agent i at time t_k is $\boldsymbol{\theta}(t_k)$, and other agents use that $\boldsymbol{\theta}$ as they receive it. Therefore, the algorithm is able to run in parallel and sequentially, see e.g. [103]. The parallel implementation of the algorithm assumes execution of the for loop (lines 2-10) by each agent at every time step.

Remark 5.3. If one of the agents terminates the execution of its part of the algorithm because, for example, it reached the rendezvous location before the duration of the prediction horizon T, then it will stay at that location until the end, unless the rendezvous location is changed. The change of location triggers the recomputation of the optimization problem and tests rendezvous and stopping conditions, which determine the further execution of the algorithm. Note that the choice of weights c_i can affect which agent may reach the rendezvous location earlier, therefore, in the scenarios with USV and UAV, we always prioritize that USV reaches the rendezvous location first.

5.3 Recursive Feasibility

In order to show feasibility of Problem 5.1, we will assume the initial feasibility and then show that the problem is recursively feasible.

Assumption 5.4. Problem 5.1 is feasible at time t_0 for each agent $i = 1, ..., M$ with $\boldsymbol{\theta}(t_0)$ initialized as in (5.8).

The main point in the proof of the rendezvous algorithm is to ensure feasibility on the consecutive steps where the rendezvous reference point $\boldsymbol{\theta}(t_k)$ is updated. The space shift of the terminal set $\mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \alpha_i)$ that occurs due to the reference change $\boldsymbol{\theta}(t_{k+1}) \neq \boldsymbol{\theta}(t_k)$ at some t_k can be quantified using the update rule (5.10).

Lemma 5.2. *For the nominal system with dynamics in Eq.* (5.2) *and reference change from* $\bar{\boldsymbol{x}}_i(t_k)$ *to* $\bar{\boldsymbol{x}}_i(t_{k+1})$*, given a local terminal set*

$$\mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \alpha_i) = \left\{ \boldsymbol{x}_i \in \mathbb{R}^{n_i} : V_{f,i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i) \leq \alpha_i^2 \right\}$$

it holds that if

$$\hat{\boldsymbol{x}}_i(t_k + T; t_k) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_k), \alpha_i(t_k))$$

then

$$\hat{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_{k+1}), \alpha_i(t_{k+1}))$$

where $\alpha_i(t_{k+1}) = \alpha_i(t_k) + \eta \left\| H_{x_i} \boldsymbol{v}_\theta(t_k) \right\|_{P_i}$.

Proof. Let us consider the optimal control law $\hat{\boldsymbol{u}}_i^*(s; t_k)$ for interval $s \in [t_k, t_k + T]$ obtained at t_k by solving Problem 5.1 and a candidate control law

$$\tilde{\boldsymbol{u}}_i(s; t_{k+1}) = \begin{cases} \hat{\boldsymbol{u}}_i^*(s; t_k) & s \in [t_{k+1}, t_k + T] \\ \bar{\boldsymbol{u}}_i + K_i(\tilde{\boldsymbol{x}}_i(s; t_k) - \bar{\boldsymbol{x}}_i) & s \in [t_k + T, t_{k+1} + T] \end{cases} \tag{5.12}$$

that generates the system trajectory $\tilde{\boldsymbol{x}}_i(s; t_{k+1})$ based on the dynamics in (5.2). It holds that $\tilde{\boldsymbol{x}}_i(t_k + T; t_{k+1}) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_k), \alpha_i(t_k))$ and, due to the invariance of the terminal set, $\tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_k), \alpha_i(t_k))$, i.e.

$$\left\| \tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) - \bar{\boldsymbol{x}}_i(t_k) \right\|_{P_i}^2 \leq \alpha_i^2(t_k). \tag{5.13}$$

Then,

$$\begin{aligned}
&\left\| \tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) - \bar{\boldsymbol{x}}_i(t_{k+1}) \right\|_{P_i} \\
&\leq \left\| \tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) - \bar{\boldsymbol{x}}_i(t_k) \right\|_{P_i} + \left\| \bar{\boldsymbol{x}}_i(t_k) - \bar{\boldsymbol{x}}_i(t_{k+1}) \right\|_{P_i} \\
&\overset{(5.13)}{\leq} \alpha_i(t_k) + \left\| \bar{\boldsymbol{x}}_i(t_k) - \bar{\boldsymbol{x}}_i(t_{k+1}) \right\|_{P_i} \\
&\overset{(5.3)}{=} \alpha_i(t_k) + \left\| H_{x_i} \boldsymbol{\theta}(t_k) - H_{x_i} \boldsymbol{\theta}(t_{k+1}) \right\|_{P_i} \\
&\overset{(5.10)}{=} \alpha_i(t_k) + \eta \left\| H_{x_i} \boldsymbol{v}_\theta(t_k) \right\|_{P_i} = \alpha_i(t_{k+1}).
\end{aligned}$$

Hence, $\tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_{k+1}), \alpha_i(t_{k+1}))$. □

Now, we can state the recursive feasibility theorem.

Theorem 5.1. *For the agents $i = 1, ..., M$ with system dynamics given by* (5.1)*, for which Assumptions 5.1 and 5.4 and Lemmas 5.1 and 5.2 hold, Problem 5.1 is feasible at $t_k, k \geq 0$.*

Proof. If the state $\boldsymbol{x}_i(t_{k+1}) \in \mathcal{X}_{f,i} \subseteq \mathcal{X}_i$ then by the invariance of the terminal set stated in Lemma 5.1, it will remain in that set. Therefore, using the terminal control law $\kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) = \bar{\boldsymbol{u}}_i + K_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \in \mathcal{U}_i$, the cost function in (5.4) is bounded and all constraints in (5.7) are satisfied.

Let us consider again the obtained optimal control law $\hat{\boldsymbol{u}}_i^*(s; t_k)$ at t_k for interval $s \in [t_k, t_k + T]$ and a candidate control law according to Eq. (5.12) that generates the system trajectory $\tilde{\boldsymbol{x}}_i(s; t_{k+1})$ based on the dynamics in (5.2).

Because of feasibility at t_k, the state $\tilde{\boldsymbol{x}}_i(s; t_{k+1}) \in \mathcal{X}_i$ for $s \in [t_{k+1}, t_k + T]$ and $\tilde{\boldsymbol{x}}_i(t_k + T; t_{k+1}) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_k), \alpha_i(t_k))$. Moreover, due to the terminal set properties from Lemma 5.1, and the result of Lemma 5.2 the candidate control law will ensure that the terminal state $\tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1})$ is in the shifted local terminal set $\tilde{\boldsymbol{x}}_i(t_{k+1} + T; t_{k+1}) \in \mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i(t_{k+1}), \alpha_i(t_{k+1}))$, which proves recursive feasibility. □

Note that this result only guarantees feasibility and does not imply convergence, which will be examined later in the chapter.

5.4 Simulation Results

In this section we evaluate Algorithm 1 implemented on nonlinear models of a quadrotor and a boat. The goal is to land the quadrotor on a boat landing platform, which is 1m×1m in size. We denote the quadrotor and the boat model and parameters with the subscripts $i = q$ and $i = b$, respectively.

5.4.1 Models and constraints

The simplified version of the quadrotor model used in this section assumes that $\boldsymbol{R}_T(\eta) = I_3$ which is a reasonable assumption if the perturbations from hover flight are small [74]. Moreover, we assume that the attitude dynamics are approximated with the first order dynamics for roll and pitch and that yaw can be instantaneously achieved [23], which allows us to directly set ϕ_{cmd}, θ_{cmd}, and $\dot{\psi}_{cmd}$. Then the approximated dynamics are

$$\begin{aligned}
\dot{\phi} &= \frac{1}{\tau_\phi}(k_\phi \phi_{cmd} - \phi), \\
\dot{\theta} &= \frac{1}{\tau_\theta}(k_\theta \theta_{cmd} - \theta), \\
\dot{\psi} &= \dot{\psi}_{cmd}
\end{aligned}$$

where $\tau_\phi, k_\phi, \tau_\theta, k_\theta$ are parameters that needs to be identified.

The state vector of the quadrotor model $\boldsymbol{x}_q$ is chosen as

$$\boldsymbol{x}_q = [p_x,\ p_y,\ p_z,\ v_x,\ v_y,\ v_z,\ \phi,\ \theta,\ \psi]^T,$$

and the input as $\boldsymbol{u}_q = \left[F_z,\ \phi_{cmd},\ \theta_{cmd},\ \dot{\psi}_{cmd}\right]^T$.

The position in $\mathbb{R}^3$ is represented with $\boldsymbol{y}_q = [p_x,\ p_y,\ p_z]^T$, and $[\dot{p}_x,\ \dot{p}_y,\ \dot{p}_z]^T = [v_x,\ v_y,\ v_z]^T$. Thus, matrix $C_q = [I_{3\times 3}, 0_{3\times 6}]$.

On the quadrotor we imposed several constraints to ensure the proper behaviour:

$$\begin{aligned} \sqrt{v_x^2 + v_y^2 + v_z^2} &\leq 17.0 \text{ m/s}, & |\dot{v}_{z,cmd}| &\leq 2.0 \text{ m/s}, \\ |v_z| &\leq 4.0 \text{ m/s}, & |\phi_{cmd}| &\leq 0.5 \text{ rad}, \\ |\phi| &\leq 0.5 \text{ rad}, & |\theta_{cmd}| &\leq 0.5 \text{ rad}, \\ |\theta| &\leq 0.5 \text{ rad}, & |\dot{\psi}_{cmd}| &\leq \pi/2 \text{ rad/s}. \end{aligned}$$

The constraints in the left column constitute the set $\mathcal{X}_q$. The first two constraints are related to the maximum velocity and vertical velocity respectively, which we want to limit to prevent fast descent. The latter two are constraints on the roll and pitch angles. The set $\mathcal{U}_q$ is formed of constraints in the right column.

The boat model is chosen as a fully actuated dynamical model with the linear drag for the purpose of these simulations. The state vector of boat model $\boldsymbol{x}_b$ is chosen as $\boldsymbol{x}_b = [p_x,\ p_y,\ \psi,\ u,\ v,\ r]^T$, and input $\boldsymbol{u}_b = [X,\ Y,\ N]^T$. The position in $\mathbb{R}^3$ space is represented with $\boldsymbol{y}_b = [p_x,\ p_y,\ 0]^T$. Matrix C_b is given as $C_b = [\text{diag}(1, 1, 0), 0_{3\times 3}]$.

The boat model set constraints $\mathcal{X}_b$ also has the velocity constraints and constraint on the angular velocity r, i.e. $\sqrt{v_x^2 + v_y^2} \leq 15.0$ m/s and $|r| \leq 0.5$ rad/s. Finally, the input constraints $\mathcal{U}_b$ has constraints on N, i.e. $|N| \leq 0.5$ rad/s^2.

5.4.2 Results

Algorithm 1 is initialized with the following parameters. The planning horizon is set as $T = 3$s and sampling period is $\delta = 0.1$s for both agents. The update parameters for $\boldsymbol{\theta}(t_k)$ are $\eta = 0.1$ and $\varepsilon = 0.1$. For the quadrotor we choose the weighting matrices as $Q_q = \text{diag}(30, 30, 6, 1, 1, 1, 1, 1, 1)$, $R_q = I$ and obtain P_q and $\bar{\alpha}_q = 0.2064$ according to the optimization problem (5.6) in Corollary 5.1. For the boat $Q_b = \text{diag}(5, 5, 1, 1, 1, 1)$, $R_b = I$, $\bar{\alpha}_b = 0.7129$. This choice of the tuning parameters prioritizes the synchronization of the agent's position in the xy-plane such that the quadrotor is above the boat and landing platform before the final descent.

We set the initial states of the quadrotor and boat such that the position in the output space is $\boldsymbol{y}_q = [4, 2, 5]^T$ and $\boldsymbol{y}_b = [-3, -1.5, 0]^T$, respectively. To determine the initial $\boldsymbol{\theta}(t_0)$ according to Eq. (5.8) we choose $c_q = 2/3$ and $c_b = 4/3$. If the initial $\boldsymbol{\theta}(t_0)$ is not changed then the agents will rendezvous at a point $\boldsymbol{\theta}(t_0) = [-0.67, -0.33, 0]^T$ that is twice closer to the boat than to the quadrotor

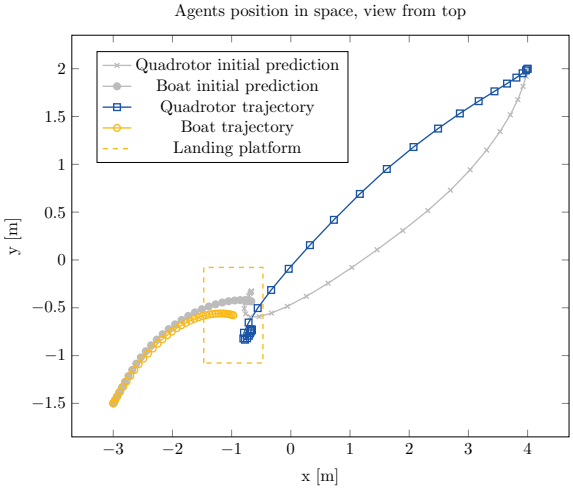

Figure 5.1: Nominal case with terminal constraints.

as the boat is slower. This is visible in Fig. 5.1 for the nominal case with terminal constraints without any disturbances. The difference between the initially predicted and actual trajectories results from the change of $\boldsymbol{\theta}(t_k)$ that occurred for the first four steps and $\boldsymbol{\theta}(t_{final}) = [-0.67, -0.73, 0]^T$. A perspective view of the same setup is shown in Fig. 5.2.

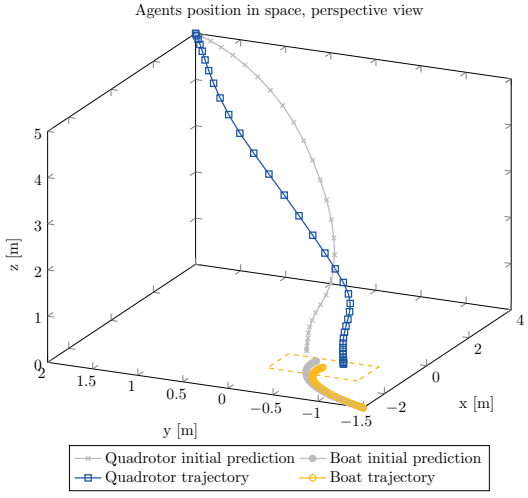

Figure 5.2: Perspective view of the setup for nominal case with terminal constraints.

In order to show the performance of the update rule for $\boldsymbol{\theta}(t_k)$ we added a strong wind disturbance in the positive y-axis direction acting from $t_1 = 0.5$s until $t_2 = 2$s, depicted in Figures 5.3 and 5.4. This causes the quadrotor to drift several meters in the direction of the disturbance. However, the feasibility is preserved at all time steps, and because of the imposed terminal constraints the updates of $\boldsymbol{\theta}(t_k)$ are small.

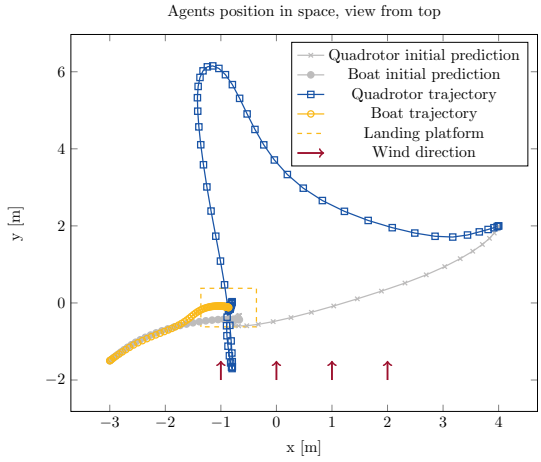

Figure 5.3: Strong wind active from $t_1 = 0.5$s until $t_2 = 2$s, case with terminal constraints

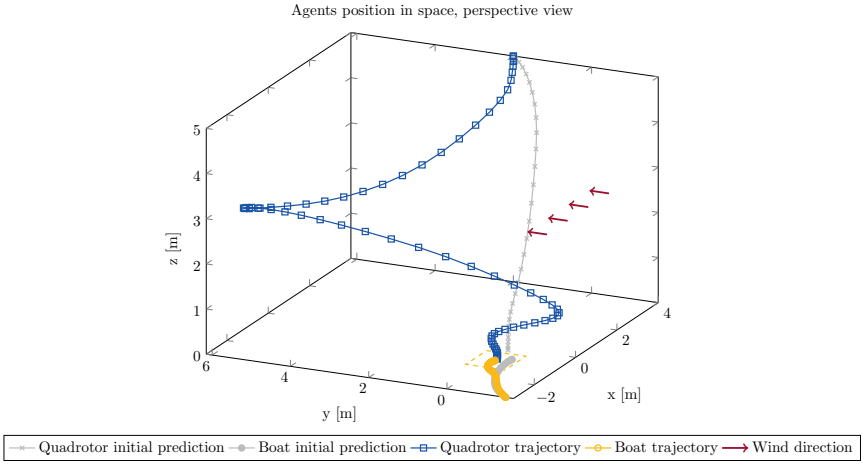

Figure 5.4: Strong wind active for $t = [0.5\text{s}, 2.0\text{s}]$, case with terminal constraints. Arrows show wind direction.

Finally, because we did not experience any feasibility issues, we removed the terminal constraints (5.7h) from Problem 5.1 to test Algorithm 1 and the update rule. In Fig. 5.5 we can notice that the boat made adjustments and approached to the quadrotor as a result of the rendezvous point updates by the quadrotor. The updates of $\boldsymbol{\theta}(t_k)$ are shown in Fig. 5.6. The bigger changes in $\boldsymbol{\theta}(t_k)$ compared to the case with the terminal constraints are due to the update rule. $V_o(\hat{\boldsymbol{y}}_i(t_k+T;t_k), \boldsymbol{\theta}(t_k))$ is evaluated at the last predicted $\hat{\boldsymbol{y}}_i(t_k+T;t_k)$ output for which the corresponding state $\hat{\boldsymbol{x}}_i(t_k+T;t_k)$ belongs to a very small set $\mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \bar{\alpha}_i)$.

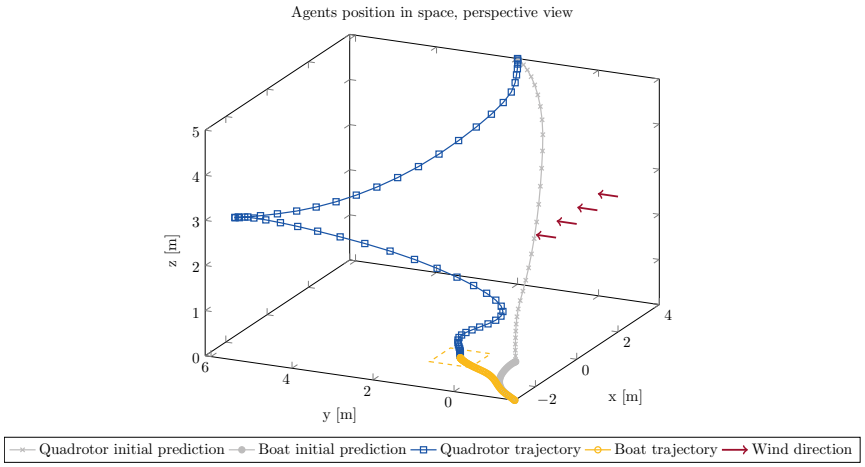

Figure 5.5: Strong wind active for $t = [0.5\text{s}, 2.0\text{s}]$, case without terminal constraints.

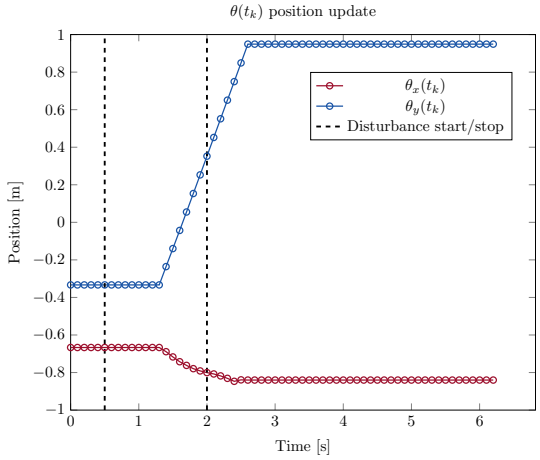

Figure 5.6: $\theta(t_k)$ evolution in time for the case without terminal constraints

5.5 Convergence Without Terminal Constraints

Analyzing the presented simulation results, one can notice that the terminal constraints can degrade the performance. For the current distributed terminal sets for tracking the rendezvous location, the agent usually terminate somewhere in the terminal set and not on the exact rendezvous location. However, by simply removing the terminal set constraints both agents terminate at the exact rendezvous location. Moreover, the recalculation of the distributed optimal control problem (DOCP) given as Problem 5.1 takes considerably less time than with the terminal constraints.

There have been several works on MPC without terminal constraints. One line of work follows the approach given in [104] which implicitly assumes existence and satisfaction of the terminal constraints, and the other one [79], [105] establishes convergence result by completely removing them. The paper [106] advocates for the usage of MPC with terminal constraints and regards the first approach as an approach with the implicit definition of the terminal constraints.

Since the implementation of Problem 5.1 is discrete, i.e. the model (5.7a) and cost (5.7b) are discretized, we conduct the convergence analysis with respect to the discretized equivalent of Problem 5.1. The notation will be slightly different than in the previous sections. We denote the system state trajectories with $\boldsymbol{x}(t)$, nominal state trajectories with $\hat{\boldsymbol{x}}(k|t)$ and optimal state trajectories with $\hat{\boldsymbol{x}}^*(k|t)$ at timestep $k+t$ predicted at timestep t.

Let us consider a nonlinear discrete time system with additive disturbances

$$\boldsymbol{x}_i(t+1) = f_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)) + \boldsymbol{w}_i(t), \tag{5.14a}$$

$$\boldsymbol{y}_i(t) = C_i \boldsymbol{x}_i(t). \tag{5.14b}$$

Let $\hat{\boldsymbol{x}}_i(k|t), \hat{\boldsymbol{y}}_i(k|t)$ be the nominal state trajectory and output, respectively, at time $k+t, k = 0, 1, ..., N$ calculated at time instant t, with assumption that $\hat{\boldsymbol{x}}_i(0|t) = \boldsymbol{x}_i(t)$, governed by the following difference equation

$$\hat{\boldsymbol{x}}_i(k+1|t) = f_i(\hat{\boldsymbol{x}}_i(k|t), \boldsymbol{u}_i(k|t)), \tag{5.15a}$$

$$\hat{\boldsymbol{y}}_i(k|t) = C_i \hat{\boldsymbol{x}}_i(k|t). \tag{5.15b}$$

Note that f_i here is not the same as in (5.1) or (5.2), but its discretized version.

The main focus of this section is to prove that under the proposed rendezvous update given with

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \boldsymbol{v}_\theta(t), \tag{5.16}$$

the agents converge to the rendezvous location.

Let us denote the next steady state and corresponding input obtained from the updated rendezvous location as

$$\bar{\boldsymbol{x}}_i^+ = H_{x_i} \boldsymbol{\theta}(t+1), \quad \bar{\boldsymbol{u}}_i^+ = H_{u_i} \boldsymbol{\theta}(t+1).$$

Let the cost function of an agent in Problem 5.1 be

$$J_i(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i, \boldsymbol{\theta}, N, t) = \|\hat{\boldsymbol{x}}_i(N|t) - \bar{\boldsymbol{x}}_i\|^2_{P_i} + \sum_{k=0}^{N-1} \left(\|\hat{\boldsymbol{x}}_i(k|t) - \bar{\boldsymbol{x}}_i\|^2_{Q_i} + \|\boldsymbol{u}_i(k|t) - \bar{\boldsymbol{u}}_i\|^2_{R_i} \right) \tag{5.17}$$

where we included the length of the prediction horizon N in the formulation. The first term represents the terminal cost

$$V_{f_i}(\hat{\boldsymbol{x}}_i(N|t), \bar{\boldsymbol{x}}_i) = \|\hat{\boldsymbol{x}}_i(N|t) - \bar{\boldsymbol{x}}_i\|^2_{P_i} \tag{5.18}$$

and the summand is the stage cost

$$l_i(\hat{\boldsymbol{x}}_i(k|t), \boldsymbol{u}_i(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) = \|\hat{\boldsymbol{x}}_i(k|t) - \bar{\boldsymbol{x}}_i\|^2_{Q_i} + \|\boldsymbol{u}_i(k|t) - \bar{\boldsymbol{u}}_i\|^2_{R_i} \tag{5.19}$$

Let the value function be

$$V_i^N(\boldsymbol{x}_i(t)) = \min_{\boldsymbol{u}_i} J_i(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i, \boldsymbol{\theta}, N, t). \tag{5.20}$$

We define the region of attraction $\mathcal{X}_i^{ROA} = \left\{ \boldsymbol{x}_i \in \mathcal{X}_i : V_i^N(\boldsymbol{x}_i) \leq V_{i,max}^N \right\}$ of the i-th DOCP controller as the set of states which can be steered to the desired state $\bar{\boldsymbol{x}}_i$ in N steps or less.

The idea is to prove that for all initial states in a region of attraction $\mathcal{X}_i^{ROA}$ the value function decreases over time, thus, establishing the asymptotic convergence of the agents under the proposed iterative update of the rendezvous location. This result is summarized in the following theorem and the proof is available in Appendix 5.C.

Theorem 5.2. *Given the bounds $V_{i,max}^N \in \mathbb{R}_{>0}$, for all $i = 1, ..., M$, there exist constants $\eta, \beta_i \in \mathbb{R}_{>0}$, $\gamma_i \geq 1$, and $N_0 \in \mathbb{N}$, such that for all $N > N_0$ and all initial conditions in the region of attraction $\boldsymbol{x}_i(0) \in \mathcal{X}_i^{ROA} = \left\{ \boldsymbol{x}_i \in \mathcal{X}_i : V_i^N(\boldsymbol{x}_i) \leq V_{i,max}^N \right\}$, the multi-agent system* (5.15) *under the rendezvous update rule* (5.16) *satisfies*

$$\|\boldsymbol{x}_i(t) - \bar{\boldsymbol{x}}_i\|^2_{Q_i} \leq V_i^N(\boldsymbol{x}_i(t)) \leq \gamma_i \|\boldsymbol{x}_i(t) - \bar{\boldsymbol{x}}_i\|^2_{Q_i}$$

$$V_i^N(\boldsymbol{x}_i(t+1)) - V_i^N(\boldsymbol{x}_i(t)) \leq -\frac{1}{2} l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)$$

for all i and $t \geq 0$. The sets $\mathcal{X}_i^{ROA}$ are stabilized and agents converge to a rendezvous location.

Corollary 5.2. *The rendezvous update step η can be chosen as $\eta = \min_i\{\bar{\eta}_i\}$ where*

$$\bar{\eta}_i := \sqrt{\frac{(\frac{1}{2} + 2\beta_i) l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) - 2\beta_i V_{i,max}^N}{(1 + \frac{1}{2\beta_i}) \lambda_{max}(H_{v_i})}}$$

and β_i can be obtained as

$$0 < \beta_i < \bar{\beta}_i := \frac{l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)}{4(V_{i,max} - l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i))}.$$

The matrix $H_{v_i} = H_{x_i}^T P_i H_{x_i} + (N-1)(H_{x_i}^T Q_i H_{x_i} + H_{u_i}^T R_i H_{u_i})$ *is a positive semi-definite matrix that depends on a priori known matrices* H_{x_i} *and* H_{u_i}*, and tuning matrices* Q_i *and* R_i*.*

5.6 Conclusion

In this chapter, we presented a rendezvous algorithm for the distributed MPC scheme for agents with nonlinear and heterogeneous dynamics. The algorithm is designed for the problem of autonomous cooperative landing of the quadrotor on the autonomous boat. During the landing the agents communicate only when it is necessary to update the rendezvous point and ensure the feasibility of the algorithm. The effectiveness of the proposed algorithm is shown with the simulation of the landing scenarios.

In future work, it can be beneficial to upgrade Theorem 5.2 and quantify the upper bound on the disturbance such that feasibility and convergence of the algorithm are preserved. Furthermore, it will be interesting to examine the behaviour of the algorithm on real systems. The inclusion of obstacles and constraints in the output space is the focus of the next chapter.

5.A Proof of Lemma 5.1

Proof. The linearized system from Assumption 5.1 around a steady state $(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)$ with the controller $\kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) = \bar{\boldsymbol{u}}_i + K_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)$ is the closed-loop asymptotically stable system $\dot{\boldsymbol{x}}_i = A_{k,i}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)$. The matrix P_i is such that following Lyapunov equation holds

$$A_{k,i}^T P_i + P_i A_{k,i} = -(Q_i + K_i^T R_i K_i) = -Q_i^*.$$

Let us define an auxiliary function $\phi_i(\boldsymbol{x}_i) = f_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i, \kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)) - A_{k,i}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)$. The derivative of $V_{f,i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i) = \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i}^2 = (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)$ along a trajectory of the nominal system $\dot{\boldsymbol{x}}_i = f_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i, \kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i))$ is

$$\begin{aligned}
V_{f,i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i) &= (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i f_i + f_i^T P_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \\
&= (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i (\phi_i(\boldsymbol{x}_i) + A_{k,i}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \\
&\quad + (\phi_i(\boldsymbol{x}_i) + A_{k,i}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \\
&= -(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T Q_i^* (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) + 2(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T P_i \phi_i(\boldsymbol{x}_i) \\
&\leq -(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T Q_i^* (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) + 2 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i} \|\phi(\boldsymbol{x}_i)\|_{P_i} \\
&\leq -(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i)^T Q_i^* (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \left(1 - \frac{2 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i} \|\phi_i(\boldsymbol{x}_i)\|_{P_i}}{\lambda_{\min}(Q_{P_i}) \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i}^2}\right) \\
&\leq -\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{Q_i^*} \left(1 - \frac{2 \|\phi_i(\boldsymbol{x}_i)\|_{P_i}}{\lambda_{\min}(Q_{P_i}) \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i}}\right)
\end{aligned}$$

where we used the facts that

$$\begin{aligned}
\boldsymbol{a}^T P_i \boldsymbol{b} &\leq \|\boldsymbol{a}\|_{P_i} \|\boldsymbol{b}\|_{P_i} \\
\lambda_{\min}(Q_{P_i}) \|\boldsymbol{a}\|_{P_i}^2 &\leq \|\boldsymbol{a}\|_{Q_i^*}^2, \forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n
\end{aligned}$$

where $Q_{P_i} = P_i^{-\frac{1}{2}} Q_i^* P_i^{-\frac{1}{2}}$.

Because $\frac{\|\phi_i(\boldsymbol{x}_i)\|_{P_i}}{\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i}} \to 0$ as $\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i} \to 0$, there exists $\bar{\alpha}_i > 0$ such that

$$\frac{\|\phi_i(\boldsymbol{x}_i)\|_{P_i}}{\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i}} \leq \frac{\lambda_{\min}(Q_{P_i})}{4}, \quad \text{for } \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i} \leq \bar{\alpha}_i. \tag{5.21}$$

Furthermore, there exists $0 < \alpha_i \leq \bar{\alpha}_i$ such that for all $\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{P_i} \leq \alpha_i$, the input constraints are satisfied, i.e. $\kappa_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) = \bar{\boldsymbol{u}}_i + K_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i) \in \mathcal{U}_i$.
Then, for a such α_i and for all $\boldsymbol{x}_i$ in

$$\mathcal{X}_{f,i}(\bar{\boldsymbol{x}}_i, \alpha_i) = \left\{\boldsymbol{x}_i \in \mathbb{R}^{n_i} : V_{f,i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i) \leq \alpha_i^2\right\}.$$

we get

$$\dot{V}_{f,i}(\boldsymbol{x}_i) = -\frac{1}{2} \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_{Q_i^*}.$$

□

5.B Lemma 5.3

Before we state the proof of Theorem 5.2, we introduce and prove a useful Lemma.

Lemma 5.3. *For any two vectors $a, b \in \mathbb{R}^n$, a positive semi-definite matrix Q and a constant $\beta > 0$ it holds that*

$$\|a+b\|_Q^2 \leq (1+2\beta)\|a\|_Q^2 + (1+\frac{1}{2\beta})\|b\|_Q^2 . \tag{5.22}$$

Proof. For any two vectors $x, y \in \mathbb{R}^n$, a positive semi-definite matrix Q and $\beta > 0$ it holds that

$$\begin{aligned} 0 &\leq (y-2\beta x)^T Q(y-2\beta x) \\ 0 &\leq y^T Qy - 2\beta y^T Qx - 2\beta x^T Qy + 4\beta^2 x^T Qx \\ 4\beta x^T Qy &\leq 4\beta^2 x^T Ax + y^T Qy \\ x^T Qy &\leq \beta x^T Qx + \frac{1}{4\beta} y^T Qy. \end{aligned}$$

Thus using the obtained inequality we get

$$\begin{aligned} \|a+b\|_Q^2 = (a+b)^T Q(a+b) &= a^T Qa + b^T Qb + 2a^T Qb \\ &\leq a^T Qa + b^T Qb + 2\beta a^T Qa + \frac{1}{2\beta} b^T Qb \\ &= (1+2\beta)a^T Qa + (1+\frac{1}{2\beta})b^T Qb \\ &= (1+2\beta)\|a\|_Q^2 + (1+\frac{1}{2\beta})\|b\|_Q^2 \end{aligned}$$

□

5.C Proof of Theorem 5.2

Proof. The proof proceeds in two parts. The first part shows the boundedness of $V_i^N(\boldsymbol{x}_i(t))$ for all $t \geq 0$, and the second part proves that the value function decreases for all states in the region of attraction at every time step.

By the definition of the value function

$$V_i^N(\boldsymbol{x}_i(t)) \geq l_i(\hat{\boldsymbol{x}}_i(0|t), \boldsymbol{u}_i(0|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \geq \|\hat{\boldsymbol{x}}_i(0|t) - \bar{\boldsymbol{x}}_i\|_{Q_i}^2 = \|\boldsymbol{x}_i(t) - \bar{\boldsymbol{x}}_i\|_{Q_i}^2$$

Moreover,

$$\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_Q^2 \leq V_i^N(\boldsymbol{x}_i(t)) \leq V_{i,max}^N$$

thus, there exist $\bar{\gamma} \geq 1$ for which

$$\bar{\gamma}\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_Q^2 = V_{i,max}^N$$

and

$$V_i^N(\boldsymbol{x}_i(t)) \leq V_{i,max}^N \leq \gamma \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i\|_Q^2$$

where $\gamma \geq \bar{\gamma} \geq 1$.

In the second part of the proof, let us consider the change of value function when the rendezvous update rule is applied which results in the steady-state parameters $\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i$ change into $\bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+$.

The value function at time step t is

$$V_i^N(x_i(t)) = V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i) + \sum_{k=0}^{N-1} l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \tag{5.23}$$

Let the candidate input sequence for the next time step be $\tilde{U}_i(t+1)$, consisting of the optimal input from the previous step and some appended $\boldsymbol{u}_i^N \in \mathcal{U}_i$ i.e.

$$\begin{aligned}\tilde{U}_i(t+1) &= (\tilde{\boldsymbol{u}}_i(0|t+1), \tilde{\boldsymbol{u}}_i(1|t+1), ..., \tilde{\boldsymbol{u}}_i(N-2|t+1), \tilde{\boldsymbol{u}}_i(N-1|t+1)) \\ &= (\boldsymbol{u}_i^*(1|t), \boldsymbol{u}_i^*(2|t), ..., \boldsymbol{u}_i^*(N-1|t), \boldsymbol{u}_i^N), \quad \boldsymbol{u}_i^N \in \mathcal{U}_i\end{aligned}$$

Assume that there are no disturbances, then

$$\boldsymbol{x}_i(t+1) = \tilde{\boldsymbol{x}}_i(0|t+1) = f_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i^*(0|t))$$

and for $k = 0, ..., N-2$, the input $\tilde{\boldsymbol{u}}_i(k|t+1) = \boldsymbol{u}_i^*(k+1|t)$ generate the state trajectory

$$\tilde{\boldsymbol{x}}_i(k|t+1) = \hat{\boldsymbol{x}}_i^*(k+1|t), \quad \text{for } k = 1, ..., N-1.$$

Then,

$$\begin{aligned}V_i^N(\boldsymbol{x}_i(t+1)) &\leq J_i(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i, \boldsymbol{\theta}^+, N, t+1) \\ &= \left\|\tilde{\boldsymbol{x}}_i(N|t+1) - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}^2 + \sum_{k=0}^{N-1} \left\|\tilde{\boldsymbol{x}}_i(k|t+1) - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\tilde{\boldsymbol{u}}_i(k|t+1) - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\ &= \left\|\tilde{\boldsymbol{x}}_i(N|t+1) - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}^2 + \sum_{k=0}^{N-2} \left\|\tilde{\boldsymbol{x}}_i(k|t+1) - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\tilde{\boldsymbol{u}}_i(k|t+1) - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\ &\quad + \left\|\tilde{\boldsymbol{x}}_i(N-1|t+1) - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\tilde{\boldsymbol{u}}_i(N-1|t+1) - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\ &= \left\|\tilde{\boldsymbol{x}}_i(N|t+1) - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}^2 + \sum_{k=1}^{N-1} \left\|\hat{\boldsymbol{x}}_i^*(k|t) - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\boldsymbol{u}_i^*(k|t) - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\ &\quad + \left\|\hat{\boldsymbol{x}}_i^*(N|t) - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\boldsymbol{u}_i^N - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2\end{aligned}$$

or in a compact form

$$V_i^N(\boldsymbol{x}_i(t+1)) \leq J_i(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i, \boldsymbol{\theta}^+, N, t+1) = V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) + \sum_{k=1}^{N-1} l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) + l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+)$$

Now, let us state the two main ingredients (proofs available in the next sections) for the convergence proof, for a chosen $\beta_i > 0$:

Ingredient 1: (proof in 5.C.1)

$$l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) \leq (1+2\beta_i) l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) + (1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2), \quad k = 1, ..., N-1 \tag{5.24}$$

Ingredient 2: (proof in 5.C.2)

$$V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) \leq -l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) + (1+2\beta_i) V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i) + (1+\frac{1}{2\beta_i})\Delta\bar{\boldsymbol{x}}_{P_i}^2 \tag{5.25}$$

Using the ingredients 1 and 2 and the equation (5.23) for the optimal cost we have,

$$\begin{aligned}
V_i^N(\boldsymbol{x}_i(t+1)) &\leq J_i(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i, \theta^+, t+1) = V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) \\
&\quad + \sum_{k=1}^{N-1} l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) + l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) \\
&\overset{(5.24)}{\leq} V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) + l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) \\
&\quad + (N-1)(1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2) \\
&\quad + (1+2\beta_i) \sum_{k=1}^{N-1} l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \\
&\overset{(5.25)}{\leq} -l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) + (1+\frac{1}{2\beta_i})\Delta\bar{\boldsymbol{x}}_{P_i}^2 \\
&\quad + (1+2\beta_i) V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i) \\
&\quad + l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) + (N-1)(1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2) \\
&\quad + (1+2\beta_i) \sum_{k=1}^{N-1} l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)
\end{aligned}$$

$$
\begin{aligned}
&= (1+2\beta_i)V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t),\bar{\boldsymbol{x}}_i) + (1+2\beta_i)\sum_{k=0}^{N-1} l_i(\hat{\boldsymbol{x}}_i^*(k|t),\boldsymbol{u}_i^*(k|t),\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{u}}_i)\\
&\quad - (1+2\beta_i)l_i(\hat{\boldsymbol{x}}_i^*(0|t),\boldsymbol{u}_i^*(0|t),\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{u}}_i)\\
&\quad + (1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{P_i}^2 + (N-1)(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2))\\
&\overset{(5.23)}{=} (1+2\beta_i)V_i^N(\boldsymbol{x}_i(t)) - (1+2\beta_i)l_i(\hat{\boldsymbol{x}}_i^*(0|t),\boldsymbol{u}_i^*(0|t),\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{u}}_i)\\
&\quad + (1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{P_i}^2 + (N-1)(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2))
\end{aligned}
$$

Therefore, we obtain

$$
\begin{aligned}
V_i^N(\boldsymbol{x}_i(t+1)) - V_i^N(\boldsymbol{x}_i(t)) \leq\ & 2\beta_i V_i^N(\boldsymbol{x}_i(t)) - (1+2\beta_i)l_i(\hat{\boldsymbol{x}}_i^*(0|t),\boldsymbol{u}_i^*(0|t),\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{u}}_i)\\
&+ (1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{P_i}^2 + (N-1)(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2))
\end{aligned}
$$

We know that

$$
\begin{aligned}
\theta(t+1) &= \theta(t) - \eta v_\theta(t),\\
\bar{\boldsymbol{x}}_i &= \bar{\boldsymbol{x}}_i(t) = H_{x_i}\theta(t), \qquad & \bar{\boldsymbol{x}}_i^+ &= \bar{\boldsymbol{x}}_i(t+1) = H_{x_i}\theta(t+1)\\
\bar{\boldsymbol{u}}_i &= \bar{\boldsymbol{u}}_i(t) = H_{u_i}\theta(t), \qquad & \bar{\boldsymbol{u}}_i^+ &= \bar{\boldsymbol{u}}_i(t+1) = H_{u_i}\theta(t+1)
\end{aligned}
$$

then

$$
\begin{aligned}
\Delta\bar{\boldsymbol{x}}_{Q_i}^2 &= \left\|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2\\
&= \left\|H_{x_i}\theta(t) - H_{x_i}\theta(t) + H_{x_i}\eta\boldsymbol{v}_\theta(t)\right\|_{Q_i}^2\\
&= \eta^2\left\|H_{x_i}\boldsymbol{v}_\theta(t)\right\|_{Q_i}^2\\
&= \eta^2\left\|\boldsymbol{v}_\theta(t)\right\|_{H_{x_i}^T Q_i H_{x_i}}^2
\end{aligned}
$$

Analogously,

$$
\begin{aligned}
\Delta\bar{\boldsymbol{u}}_{R_i}^2 &= \eta^2\left\|\boldsymbol{v}_\theta(t)\right\|_{H_{u_i}^T R_i H_{u_i}}^2\\
\Delta\bar{\boldsymbol{x}}_{P_i}^2 &= \eta^2\left\|\boldsymbol{v}_\theta(t)\right\|_{H_{x_i}^T P_i H_{x_i}}^2
\end{aligned}
$$

Then,

$$
\begin{aligned}
\Delta\bar{\boldsymbol{x}}_{P_i}^2 + (N-1)(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2) &= \eta^2\left\|\boldsymbol{v}_\theta(t)\right\|_{H_{x_i}^T P_i H_{x_i} + (N-1)(H_{x_i}^T Q_i H_{x_i} + H_{u_i}^T R_i H_{u_i})}^2\\
&= \eta^2\left\|\boldsymbol{v}_\theta(t)\right\|_{H_{v_i}}^2\\
&\leq \eta^2\lambda_{max}(H_{v_i})\left\|\boldsymbol{v}_\theta(t)\right\|^2\\
&= \eta^2\lambda_{max}(H_{v_i}).
\end{aligned}
$$

Where we utilized the facts that $\boldsymbol{v}_\theta(t)$ is the normalized unit vector and that the matrix $H_{v_i} = H_{x_i}^T P_i H_{x_i} + (N-1)(H_{x_i}^T Q_i H_{x_i} + H_{u_i}^T R_i H_{u_i})$ is a positive semi-definite matrix, because P_i, Q_i, R_i are positive definite matrices.

Finally, for all $\boldsymbol{x}_i \in \mathcal{X}_i^{ROA}$ it holds:

$$\begin{aligned} V_i^N(\boldsymbol{x}_i(t+1)) - V_i^N(\boldsymbol{x}_i(t)) &\leq 2\beta_i V_i^N(\boldsymbol{x}_i(t)) - (1+2\beta_i) l_i(\hat{\boldsymbol{x}}_i^*(0|t), u_i^*(0|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \\ &\quad + (1 + \frac{1}{2\beta_i})(\Delta \bar{\boldsymbol{x}}_{P_i}^2 + (N-1)\Delta \bar{\boldsymbol{x}}_{Q_i}^2 + \Delta \bar{\boldsymbol{u}}_{R_i}^2) \\ &\leq 2\beta_i V_i^N(\boldsymbol{x}_i(t)) - (1+2\beta_i) l_i(\hat{\boldsymbol{x}}_i^*(0|t), \boldsymbol{u}_i^*(0|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \\ &\quad + (1 + \frac{1}{2\beta_i})\eta^2 \lambda_{max}(H_{v_i}) \\ &\leq -(1+2\beta_i) l_i(\hat{\boldsymbol{x}}_i^*(0|t), \boldsymbol{u}_i^*(0|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) + 2\beta_i V_{i,max} \\ &\quad + (1 + \frac{1}{2\beta_i})\eta^2 \lambda_{max}(H_{v_i}) \end{aligned}$$

Then, using the fact that $\hat{\boldsymbol{x}}_i^*(0|t) = \boldsymbol{x}_i(t)$ and $\boldsymbol{u}_i^*(0|t) = \boldsymbol{u}_i(t)$ and by choosing $\eta = \min_i\{\bar{\eta}_i\}$ where

$$\bar{\eta}_i := \sqrt{\frac{(\frac{1}{2} + 2\beta_i) l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) - 2\beta_i V_{i,max}^N}{(1 + \frac{1}{2\beta_i})\lambda_{max}(H_{v_i})}}$$

because there exists $\beta_i > 0$ such that

$$\left(\frac{1}{2} + 2\beta_i\right) l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{x}_i, \bar{u}_i) - 2\beta_i V_{i,max} > 0$$

Hence,

$$0 < \beta_i < \bar{\beta}_i := \frac{l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i)}{4(V_{i,max} - l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i))}.$$

The obtained step η as $\eta = \min_i\{\bar{\eta}_i\}$ can be written as $\eta^2 + \Delta\eta_i = \bar{\eta}_i^2$, where $\Delta\eta_i \geq 0$, then

$$\begin{aligned} V_i^N(\boldsymbol{x}_i(t+1)) - V_i^N(\boldsymbol{x}_i(t)) &\leq -\frac{1}{2} l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) - (1 + \frac{1}{2\beta_i})\Delta\eta_i \lambda_{max}(H_{v_i}) \\ &\leq -\frac{1}{2} l_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \end{aligned}$$

because $(1 + \frac{1}{2\beta_i})\Delta\eta_i \lambda_{max}(H_{v_i}) \geq 0$, which completes the proof. □

5.C.1 Ingredient 1

For $k = 1, ..., N-1$ and a chosen $\beta_i > 0$ it holds

$$\begin{aligned} l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) \leq & (1+2\beta_i) l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) \\ & + (1 + \frac{1}{2\beta_i})(\Delta \bar{\boldsymbol{x}}_{Q_i}^2 + \Delta \bar{\boldsymbol{u}}_{R_i}^2) \end{aligned}$$

where

$$\Delta\bar{\boldsymbol{x}}_{Q_i} = \left\|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}, \quad \Delta\bar{\boldsymbol{u}}_{R_i} = \left\|\bar{\boldsymbol{u}}_i - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}$$

Proof.

$$\begin{aligned}
l_i(\hat{\boldsymbol{x}}_i^*(k|t),\boldsymbol{u}_i^*(k|t),\bar{\boldsymbol{x}}_i^+,\bar{\boldsymbol{u}}_i^+) &= \left\|\hat{\boldsymbol{x}}_i^*(k|t) - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\boldsymbol{u}_i^*(k|t) - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\
&= \left\|\hat{\boldsymbol{x}}_i^*(k|t) - \bar{\boldsymbol{x}}_i + \bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 + \left\|\boldsymbol{u}_i^*(k|t) - \bar{\boldsymbol{u}}_i + \bar{\boldsymbol{u}}_i - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\
&\overset{(5.22)}{\leq} (1+2\beta_i)\left\|\hat{\boldsymbol{x}}_i^*(k|t) - \bar{\boldsymbol{x}}_i\right\|_{Q_i}^2 + (1+\frac{1}{2\beta_i})\left\|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{Q_i}^2 \\
&\quad + (1+2\beta_i)\left\|\boldsymbol{u}_i^*(k|t) - \bar{\boldsymbol{u}}_i\right\|_{R_i}^2 + (1+\frac{1}{2\beta_i})\left\|\bar{\boldsymbol{u}}_i - \bar{\boldsymbol{u}}_i^+\right\|_{R_i}^2 \\
&= (1+2\beta_i) l_i(\hat{\boldsymbol{x}}_i^*(k|t), \boldsymbol{u}_i^*(k|t), \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{u}}_i) + (1+\frac{1}{2\beta_i})(\Delta\bar{\boldsymbol{x}}_{Q_i}^2 + \Delta\bar{\boldsymbol{u}}_{R_i}^2)
\end{aligned}$$

□

5.C.2 Ingredient 2

It holds that

$$\begin{aligned}
V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) \leq &- l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) + (1+2\beta_i) V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i) \\
&+ (1+\frac{1}{2\beta_i})\Delta\bar{\boldsymbol{x}}_{P_i}^2
\end{aligned}$$

where

$$\Delta\bar{\boldsymbol{x}}_{P_i} = \left\|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}.$$

Proof. Let us use the standard lemma for the terminal sets which states that in a positive invariant set $\mathcal{X}_{f,i}$ it holds

$$V_{f_i}(\boldsymbol{x}_i^+, \bar{\boldsymbol{x}}_i^+) - V_{f_i}(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_i^+) \leq -l_i(\boldsymbol{x}_i, \boldsymbol{u}_{k,i}, \bar{\boldsymbol{x}}^+, \bar{\boldsymbol{u}}^+)$$

with some terminal control law $\boldsymbol{u}_{k,i} = \bar{\boldsymbol{u}}_i^+ + K_i(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_i^+)$ and Lyapunov function V_{f_i} around a steady state $(\bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+)$.

Since $\hat{\boldsymbol{x}}_i(N,t) = \tilde{\boldsymbol{x}}_i(N-1, t+1) \in \mathcal{X}_{f,i}$ then there exists a terminal law $\boldsymbol{u}_i^N = \boldsymbol{u}_{k,i} = \bar{\boldsymbol{u}}_i^+ + K_i(\tilde{\boldsymbol{x}}_i(N-1,t+1) - \bar{\boldsymbol{x}}_i^+)$ such that $\tilde{\boldsymbol{x}}_i(N,t+1) \in \mathcal{X}_{f,i}$ and it holds

$$V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) \leq V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i^+) - l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) \tag{5.26}$$

We have

$$\begin{aligned} V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i^+) &= \left\|\hat{\boldsymbol{x}}_i^*(N|t) - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}^2 \\ &= \left\|\hat{\boldsymbol{x}}_i^*(N|t) - \bar{\boldsymbol{x}}_i + \bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}^2 \\ &\overset{(5.22)}{\leq} (1+2\beta_i)\left\|\hat{\boldsymbol{x}}_i^*(N|t) - \bar{\boldsymbol{x}}_i\right\|_{P_i}^2 + (1+\frac{1}{2\beta_i})\left\|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_i^+\right\|_{P_i}^2 \\ &= (1+2\beta_i)V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i) + (1+\frac{1}{2\beta_i})\Delta\bar{\boldsymbol{x}}_{P_i}^2 \end{aligned}$$

and replacing it in the equation (5.26) we obtain

$$\begin{aligned} V_{f_i}(\tilde{\boldsymbol{x}}_i(N|t+1), \bar{\boldsymbol{x}}_i^+) \leq & (1+2\beta_i)V_{f_i}(\hat{\boldsymbol{x}}_i^*(N|t), \bar{\boldsymbol{x}}_i) + (1+\frac{1}{2\beta_i})\Delta\bar{\boldsymbol{x}}_{P_i}^2 \\ & - l_i(\hat{\boldsymbol{x}}_i^*(N|t), \boldsymbol{u}_i^N, \bar{\boldsymbol{x}}_i^+, \bar{\boldsymbol{u}}_i^+) \end{aligned}$$

□

Chapter 6

Leader-follower Rendezvous with Distributed Predictive Control

In this chapter, we redesign the multi-agent rendezvous problem as the leader-follower problem with one leader and one or more follower agents for autonomous landings. The leader in our case is the boat due to the less agile dynamics compared to the quadrotor that is considered as a follower.

We propose a novel distributed model predictive control (DMPC) based algorithm with a trajectory predictor for a scenario of landing of UAVs on a moving USV. The algorithm is executing DMPC with exchange of trajectories between the agents at a sufficient rate. In the case of loss of communication, and given the sensor setup, agents are predicting the trajectories of other agents based on the available measurements and prior information. The predictions are then used as the reference inputs to DMPC. During the landing, the followers are tasked with avoidance of USV-dependent obstacles and inter-agent collisions. In the proposed distributed algorithm, all agents solve their local optimization problem in parallel and we prove the convergence of the proposed algorithm.

The considered agents in our setup are equipped with sensors, i.e. a camera on the quadrotor (UAV) side, and a radar on the boat (USV) side, that enable the accurate position measurement of the neighbouring agents with a frequency significantly higher than the real-time execution frequency of the MPC. In the case of loss of communication between the agents, an agent is left only with the data history and locally available measurements. During challenging maneuvers such as landing or navigating through a space with moving obstacles, the frequency of execution of the DMPC relative to the moving speeds of the agents and obstacles can severely affect the performance and safety. This is especially the case when the references are based only on the current measurements and not on the committed trajectories, which indeed occurs in the loss of communication scenario. Note that, the communication losses we consider can be both temporary and permanent, because even an occasional package drop for few seconds that can occur during the hardware-in-the-loop experiments can lead to disastrous outcomes.

The hierarchical approaches presented in the literature [17, 107] deal with the described problem by separating the control in different layers. Usually, there is a high-level planning layer and a safety layer, for example with Control Barrier Functions (CBF) operating on a much higher frequency that is tasked with safety in the presence of obstacles [108]. Fundamentally, the concept of MPC is to unify these layers. Both approaches have their advantages and disadvantages and all aspects of the chosen concept must be taken into account to avoid unwanted behaviors. One of the ways is to use a prediction scheme to complement the available data. In [60], the authors propose a vision-based framework with the state estimation for the ground vehicle considered as a moving target. Estimation-based control protocol proposed in [61] uses only local observations of the state of the neighboring agents for rendezvous and flocking control. In another estimation-based protocol, authors in [62] use particle filter to predict aircraft trajectories. Prediction-based navigation in a decentralized event-based scheme is studied in [63]. Moreover, learning-based prediction approaches for multi-agent systems in the recent years became very popular area of research [64–68]. Unfortunately, most of these approaches require extensive amount of data for the training purpose and direct transferability to our particular application is unknown.

In this chapter, we case the multi-agent and heterogeneous (involving UAVs and USVs) rendezvous problem as a leader-follower network with one leader and one or more follower agents for autonomous landings. We formulate the problem as a cooperative distributed model predictive control problem with collision avoidance constraints. In the case when the predicted trajectory is available to the followers, and the follower is able to dynamically follow the given reference trajectory, the landing can be executed using a relatively simple MPC for a single agent follower. However, a sudden communication loss can severely destabilize the landing leading to a collision with obstacles on the boat. Based on this observation, we adopt a prediction scheme in the case of communication loss to enhance the safety and performance.

Moreover, we generalize the scheme to multiple-follower rendezvous in which the inter-agent collisions must be handled. There are several challenges in the inter-agent collision avoidance in this case. First, in the sequential application of the algorithm each agent is optimizing its control strategy based on the shared trajectories that other agents have committed to. In this case, each agent must wait until all other agents have shared their new trajectories to begin the computation of its next control input. Otherwise, using the old shared trajectories without additional safety measures can lead to a collision. Second, in the parallel version the agents operate with the most recent available shared trajectories and account for the worst-case deviations as a safety measure. However, this can be very conservative and lead to deadlock. Third, if one or more agents lose the communication for some duration safety can be compromised. This, in some sense, resembles a non-cooperative scenario and we propose appropriate safety measures.

The contributions of this chapter are

- a rendezvous algorithm based on leader-follower DMPC formulation for autonomous landing in presence of obstacles
- a convergence proof for a probabilistic safe landing in the case of communication loss
- an inter-agent collision avoidance robust to communication losses

The chapter is organized as follows. We first present the cooperative distributed leader-follower MPC formulation and define the safety constraints. Then, we elaborate the prediction scheme and the proposed rendezvous algorithm. Furthermore, we present the convergence analysis and the simulation results.

Notation

We denote the discrete time step with t and set of agents with $\mathcal{N} := \{l, f_1, ..., f_M\}$. The state trajectories are denoted with $\boldsymbol{x}_i(t)$, the predicted nominal state trajectories with $\hat{\boldsymbol{x}}_i(k|t)$ and optimal state trajectories with $\hat{\boldsymbol{x}}_i^*(k|t)$ for an agent i at time step $k+t$ predicted at time step t. The collection of trajectories from time step t until $t+N$ is denoted with $\boldsymbol{x}_i(\cdot|t)$; $\mathcal{B}_r := \{\boldsymbol{a} : \|\boldsymbol{a}\| \leq r\}$ is a ball of radius r; $\oplus$ is Minkowski sum addition defined as $\boldsymbol{x} \oplus S := \{\boldsymbol{x} + \boldsymbol{a} : \boldsymbol{a} \in S\}$. For an element $i \in \mathcal{N}$, we denote a set excluding the element i with $\mathcal{N}_{-i} := \mathcal{N} \setminus \{i\}$.

6.1 Problem Formulation

Consider a multi-agent system consisting of one leader and M follower agents that are dynamically decoupled and behaving according to the nonlinear discrete time dynamics

$$\boldsymbol{x}_i(t+1) = f_i(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)) + \boldsymbol{w}_i(t), \tag{6.1}$$

where $i \in \mathcal{N} = \{l, f_1, ..., f_M\}$, l denotes the leader, $f_1, ..., f_M$ denote the followers, M is the number of follower agents, $\boldsymbol{x}_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$, $\boldsymbol{u}_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ denote the state and input of an agent i that are subject to the state and input constraints $\mathcal{X}_i$ and $\mathcal{U}_i$, respectively, and $\boldsymbol{w}_i \in \mathcal{W}_i \subset \mathbb{R}^{n_i}$ is unknown but bounded disturbance in a compact set $\mathcal{W}_i$.

The follower agents have the same dynamics and state space of a quadrotor UAV as in (2.23) with the approximation of the attitude dynamics as introduced in Subsection 5.4.1. The dynamics and the state space of the follower agents are different from the leader ones modeled as an underactuated 3DoF boat USV model introduced in Subsection 2.3.2. We assume that the landing platform is rigidly attached to the boat USV and neglect the heave (vertical motion), roll and pitch motion. Furthermore, the first three elements of state vectors of all agents denote the position $\boldsymbol{p}_i \in \mathbb{R}^3$.

The overall multi-agent system dynamics in stack-vector form are

$$\boldsymbol{x}(t+1) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t), \tag{6.2}$$

where $\boldsymbol{x} = [\boldsymbol{x}_l^T, \boldsymbol{x}_{f_1}^T, ..., \boldsymbol{x}_{f_M}^T]^T$, $\boldsymbol{u} = [\boldsymbol{u}_l^T, \boldsymbol{u}_{f_1}^T, ..., \boldsymbol{u}_{f_M}^T]^T$, $\boldsymbol{w} = [\boldsymbol{w}_l^T, \boldsymbol{w}_{f_1}^T, ..., \boldsymbol{w}_{f_M}^T]^T$. The state and input constraints of the overall system are $\boldsymbol{x} \in \mathcal{X} := \mathcal{X}_l \times \mathcal{X}_{f_1} \times ... \times \mathcal{X}_{f_M}$, and $\boldsymbol{u} \in \mathcal{U} := \mathcal{U}_l \times \mathcal{U}_{f_1} \times ... \times \mathcal{U}_{f_M}$, the set of all disturbances is $\boldsymbol{w} \in \mathcal{W} := \mathcal{W}_l \times \mathcal{W}_{f_1} \times ... \times \mathcal{W}_{f_M}$.

We formulate the inter-agent collision avoidance constraints as

$$h_{ij}(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) \geq 0, \quad \forall i, j \in \mathcal{N}, i \neq j, \tag{6.3}$$

where $h_{ij} : \mathcal{X}_i \times \mathcal{X}_j \to \mathbb{R}$ is a function that encodes collisions and will be defined later. Given a solution $\boldsymbol{x}(t)$ of the system (6.2), if the constraint functions (6.3) are satisfied for all $t \geq t_0$ then $\boldsymbol{x}(t)$ is a collision-free solution, i.e. $\boldsymbol{x}(t) \in \mathcal{F} \subseteq \mathcal{X}$, where $\mathcal{F}$ denotes the collision-free space.

This chapter considers the problem of navigating the leader agent to follow a given reference and follower agents to a rendezvous position with respect to the leader while avoiding inter-agent collisions. In general this corresponds to Problem 1.2, but here it is stated in a more detailed manner. We denote the leader reference $\boldsymbol{x}_{r,l}(t) = \boldsymbol{x}_r(t)$ and references of the followers $\boldsymbol{x}_{r,i}(\boldsymbol{x}_l(t), \boldsymbol{c}_i) := [(\boldsymbol{p}_l(t) + \boldsymbol{c}_i)^T, \boldsymbol{0}_{n_i-3}^T]^T$ with respect to the position $\boldsymbol{p}_l(t)$ of the leader $\boldsymbol{x}_l$ and a given offset $\boldsymbol{c}_i \in \mathbb{R}^3$ encoding the particular landing position of an i-th agent, $i \in \mathcal{N}_{-l}$. With a slight abuse of notation, the problem treated in the chapter can be stated as follows.

Problem 6.1. Consider a multi-agent system (6.1). Design a control policy $\boldsymbol{u}$ such that

$$\begin{aligned} &\boldsymbol{x}(t) \in \mathcal{F}, \quad t \geq t_0 \\ &\lim_{t\to\infty} (\boldsymbol{x}_l(t), \boldsymbol{u}_l(t)) = (\boldsymbol{x}_r(t), \boldsymbol{0}_{n_l}) \\ &\lim_{t\to\infty} (\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)) = (\boldsymbol{x}_{r,i}(\boldsymbol{x}_l(t), \boldsymbol{c}_i), \boldsymbol{0}_{n_i}), \forall i \in \mathcal{N}_{-l} \end{aligned}$$

Moreover, we assume that agents are able to communicate and share their current and predicted positions $\hat{\boldsymbol{z}}_i(\cdot|t)$ asynchronously. Because the follower agents have different dynamics from the leader, and different state vectors, we define the following mapping from the leader state space $\mathcal{X}_l$ to the follower state space $\mathcal{X}_i$, $i \in \mathcal{N}_{-l}$

$$\hat{\boldsymbol{z}}_l(t) = H\hat{\boldsymbol{x}}_l(t), \tag{6.4}$$

where $H = \mathrm{diag}(I_3, 0_{(n_i-3)\times(n_l-3)})$. This mapping effectively maps only the position of the leader to the follower state space. For the follower agents it holds $\hat{\boldsymbol{z}}_i(t) = \hat{\boldsymbol{x}}_i(t)$, $i \in \mathcal{N}_{-l}$.

When the communication is without losses, all follower agents have access to $\hat{\boldsymbol{z}}_i(\cdot|t-1)$ for all $i \in \mathcal{N}$, where all data arrived between time steps $t-1$ and t is cast as data from time step $t-1$. However, we want that the designed control policy $\boldsymbol{u}$ is able to achieve the goal stated in Problem 6.1 even in the case of communication loss. In the context of this chapter, the communication loss is considered as an inability of an agent to retrieve the latest shared data from another agent. More formally, the communication loss is an inability of an agent i at time step t to retrieve the

shared data $\hat{\boldsymbol{z}}_j(\cdot|t-k)$ from another agent j generated at time step $t-k$, where $k \geq 1$. Thus, we will impose stronger assumptions on the behaviour of the followers that will be stated in Sections 6.6 and 6.7.

6.2 Distributed MPC Formulation

We choose to address Problem 6.1 using distributed MPC where each agent solves a distributed optimal control problem for a planning horizon of N time steps and applies the first control input during the control horizon Δt [50]. Note that all time steps in the control architecture are discrete $t \in \mathbb{N}_0$ and Δt denotes the continuous time duration between two time steps. The procedure is then repeated at every time step for each agent.

Let $\hat{\boldsymbol{x}}_i(k|t)$ be the nominal state trajectory at time $k+t$, $k = 0, 1, ..., N$ calculated at time instant t, where $\hat{\boldsymbol{x}}_i(0|t) = \boldsymbol{x}_i(t)$, governed by the following difference equation

$$\hat{\boldsymbol{x}}_i(k+1|t) = f_i(\hat{\boldsymbol{x}}_i(k|t), \boldsymbol{u}_i(k|t)) \tag{6.5}$$

Each agent is provided with the state reference trajectory at time step t until $t+N$ which is given as $\boldsymbol{x}_{r,i}(\cdot|t) = \{\boldsymbol{x}_{r,i}(0|t), ..., \boldsymbol{x}_{r,i}(N|t)\}$. The references are provided by the proposed algorithm which will be elaborated in Sections 6.7 and 6.6. The control objective of each agent at time step t is to minimize the following cost function

$$J_i(\hat{\boldsymbol{x}}_i(\cdot|t), \boldsymbol{u}_i(\cdot|t), \boldsymbol{x}_{r,i}(\cdot|t), N, t) = \sum_{k=0}^{N} \|\hat{\boldsymbol{x}}_i(k|t) - \boldsymbol{x}_{r,i}(k|t)\|_{Q_i}^2 \tag{6.6}$$

while respecting the constraints, where the summand represents the stage cost and Q_i is a positive-definite weighting matrix. We denote also $J = \sum_{i \in \mathcal{N}} J_i$.

We formulate the distributed optimal control problem with respect to the objective.

Problem 6.2. Let the states of the agents at time t be $\boldsymbol{x}_i(t)$, $i \in \mathcal{N}$. Given the references $\boldsymbol{x}_{r,i}(\cdot|t)$ and the predicted trajectories of other agents $\hat{\boldsymbol{z}}_j(\cdot|t)$, $j \in \mathcal{N}_{-i}$, the distributed optimal control problem is formulated as

$$\min_{\boldsymbol{u}_i(\cdot|t)} J_i(\hat{\boldsymbol{x}}_i(\cdot|t), \boldsymbol{u}_i(\cdot|t), \boldsymbol{x}_{r,i}(\cdot|t), N, t) \tag{6.7a}$$

subject to

$$\hat{\boldsymbol{x}}_i(k+1|t) = f_i(\hat{\boldsymbol{x}}_i(k|t), \boldsymbol{u}_i(k|t)), \tag{6.7b}$$
$$\hat{\boldsymbol{x}}_i(k|t) \in \mathcal{X}_i, \tag{6.7c}$$
$$\boldsymbol{u}_i(k|t) \in \mathcal{U}_i, \tag{6.7d}$$
$$\hat{\boldsymbol{x}}_i(k|t) \in \mathcal{X}_{i,j}(\hat{\boldsymbol{z}}_j(k|t)), \quad \text{for } i \in \mathcal{N}_{-l}, j \in \mathcal{N}_{-i}, \tag{6.7e}$$

for $k = 0, 1, ..., N$, for all $i \in \mathcal{N}$.

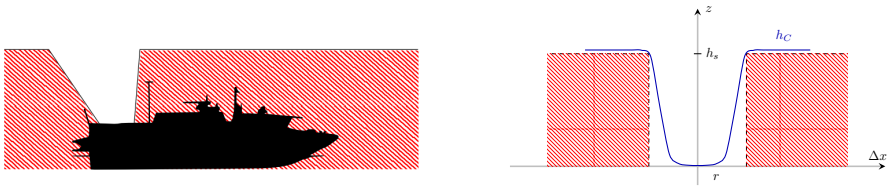

Figure 6.1: Restricted area and side view of the constraint h_C from (6.8)

Set $\mathcal{X}_i$ denotes the set of model state constraints and $\mathcal{U}_i$ the input constraints. $\mathcal{X}_{i,j}(\hat{\boldsymbol{z}}_j(k|t))$ is the set of spatiotemporal safety constraints with respect to the other agents that will be formulated in the next section. Note that the leader is not subjected to the inter-agent collision constraints (6.7e), but only the follower agents.

6.3 Spatiotemporal Safety Constraints

The desired specification on UAV movements in the 3D space are to fly above some prescribed height and avoid obstacles. However, in the problem of UAV landing on a boat, we are particularly interested in the obstacles that arise from the boat shape, equipment and its movement. These specific obstacles surrounding the landing area on the boat are depicted in Figure 6.1.

This area can be modeled as two convex areas with a binary variable that depends on the altitude of the UAV and determines which of the two constraints should be enforced within the mixed-integer optimization problem [42]. However, it is of interest to model the restricted flying area with one nonlinear continuous function such that the usage of mixed-integer programming is avoided due to the computational burden it can cause.

In the following considerations, we assume the landing platform is circular with the radius r, $r_{safe} < r$ is the safety radius needed for safe landing and h_s is the safety height below which the UAV is not allowed to descent unless above the landing platform. Let $\boldsymbol{z}_l = [p_{x_c}, p_{y_c}, 0^T_{n_f-2}]^T \in \mathcal{X}_f$ be the center of the boat landing platform and $\boldsymbol{x}_f = [p_x, p_y, p_z, 0^T_{n_f-3}]^T \in \mathcal{X}_f$ be the position of the UAV.

We define the constraint function h_C in a novel way as

$$h_C(\boldsymbol{x}_f, \boldsymbol{z}_l) := p_z - \frac{h_s}{1 + e^{-\beta((p_x - p_{x_c})^2 + (p_y - p_{y_c})^2 - r^2)}} \geq 0 \tag{6.8}$$

where $\beta > 0$ is a tuning parameter that determines the slope of the funnel.

Requiring that $h_C(\boldsymbol{x}_f, \boldsymbol{z}_l) \geq 0$, the UAV will always be above the boat position-dependent constraint function. The boundary of the restricted area defined with h_C in 3D is visible on Fig. 6.2.

The spatiotemporal constraint imposed on the position of the follower $\hat{\boldsymbol{x}}_i(k|t)$, $i \in \mathcal{N}_{-l}$ at time step $t + k$, predicted at time t, for $i = 0, 1, ..., N$, with respect to

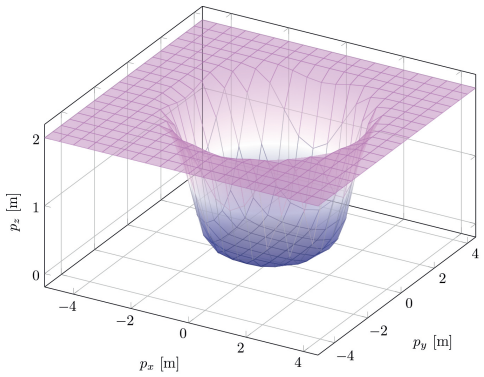

Figure 6.2: h_C from (6.8), for $h_s = 2, r = 2.5, \beta = 1$

the future trajectory of the leader position $\hat{\boldsymbol{z}}_l(k|t)$ is

$$h_C(\hat{\boldsymbol{x}}_i(k|t), \hat{\boldsymbol{z}}_l(k|t)) \geq 0 \quad \text{for all } k = 0, ..., N. \tag{6.9}$$

Therefore, we define the safe set $\mathcal{X}_{i,l}(\hat{\boldsymbol{z}}_l(k|t))$ in (6.7e) for all $i \in \mathcal{N}_{-l}$ as

$$\mathcal{X}_{i,l}(\hat{\boldsymbol{z}}_l(k|t)) := \{\hat{\boldsymbol{x}}_i(k|t) : h_C(\hat{\boldsymbol{x}}_i(k|t), \hat{\boldsymbol{z}}_l(k|t)) \geq 0\}\,. \tag{6.10}$$

Note that in the current formulation we assume that the height of the landing platform is zero.

6.4 Inter-agent Collision Avoidance

The previous section defines the safety constraint function that each follower enforces in its optimization problem with respect to the leader. The inter-agent collision avoidance condition is formulated on a set of follower agents $\mathcal{M} = \mathcal{N}_{-l}$ as

$$\|C(\hat{\boldsymbol{x}}_i(k|t) - \hat{\boldsymbol{z}}_j(k|t))\| \geq R, \text{ for } j \in \mathcal{M}_{-i} \tag{6.11}$$

for $k = 0, 1, ..., N$, where $R > 0$ is the minimal distance between the follower agents. The matrix $C = \text{diag}(1, 1, c, \mathbf{0}_{n_i-3})$, with $c > 1$, determines the shape of the super ellipsoidal set constraint such that it prevents the collision and the downwash effect that can occur if a UAV ends up below another.

Moreover, the condition (6.11) can be stated in form (6.3) as

$$h_{ij}(\hat{\boldsymbol{x}}_i(k|t), \hat{\boldsymbol{z}}_j(k|t)) := \|C(\hat{\boldsymbol{x}}_i(k|t) - \hat{\boldsymbol{z}}_j(k|t))\| - R \geq 0$$

and the safe set $\mathcal{X}_{i,j}(\hat{\boldsymbol{z}}_j(k|t))$ in (6.7e) for all $i \in \mathcal{M}$ and $j \in \mathcal{M}_{-i}$ is defined as

$$\mathcal{X}_{i,j}(\hat{\boldsymbol{z}}_j(k|t)) := \{\hat{\boldsymbol{x}}_i(k|t) : h_{ij}(\hat{\boldsymbol{x}}_i(k|t), \hat{\boldsymbol{z}}_j(k|t)) \geq 0\}\,.$$

6.5 EKF Predictor

In order to handle the communication losses, we equip the agents with an Extended Kalman Filter (EKF) as an N-step predictor of the future trajectory of the other agents. There is a significant portion of literature considering the trajectory prediction of the other agents in the multi-agent system [60–63]. From Extended Kalman Filter, Particle filter to predicting human-like behaviour using machine learning, that can be based on cooperative or even adversarial behaviour of the other agent [64–68].

EKF Predictor is formulated for a general nonlinear system $\boldsymbol{x}^+ = f(\boldsymbol{x})$ with linearized dynamics around a measured state and unknown input. The N-step prediction is done by repeating the prediction step N times in open loop. The outputs of the EKF Predictor used in the control architecture are the predicted trajectories $\hat{\boldsymbol{z}}_{p,i}(\cdot|t)$ and the prediction covariance matrices $P_{p,i}(\cdot|t)$. The prediction covariance matrix $P_{p,i}(k|t)$ provides us with the estimate of a super ellipsoidal set determined by its eigenvalues $\lambda_k(P_{p,i})$.

The tracked agent is considered to have Newtonian dynamics of second order, thus the predictor with slight modifications can be used to track other follower agents such as quadrotors. EKF Predictor is given as Alg. 2 and formulated for a general system $\boldsymbol{x}^+ = f(\boldsymbol{x})$ with linearized dynamics around a measured state

$$A(t-1) := \frac{\partial f(\boldsymbol{x}_m(t-1))}{\partial \boldsymbol{x}} \tag{6.12}$$

with measurement matrix $H(t)$, which is different than fixed H matrix used in the previous sections to map the leader states to the follower state space for rendezvous.

EKF Predictor will be used to predict the movements of other agents in the scheme, if their shared trajectories become unavailable, due to an eventual loss of communication. Note that EKF Predictor is not using the control input as it is unknown to the agent predicting the future trajectory. Therefore, the covariance matrix $P_p(k|t)$ provide us with the estimated super ellipsoidal set that will be used for the inter-agent collision as well.

6.6 Rendezvous Algorithm

The multi-agent formulation in which we have multiple followers is challenging due the possible inter-agent collisions even in the case when all agents share their committed trajectories. In this section, we formulate the algorithm for multiple-follower rendezvous landing.

We assume that the landing platform is large enough to accommodate M agents that can land simultaneously at prespecified positions $\boldsymbol{c}_i \in \mathcal{B}_{r-r_{safe}}$ relative to the center of the landing platform. Moreover, due to the application testbed in hand [109], we will constrain the scheme to M agents, and additional follower agents can be included if a prioritization procedure is included in the proposed algorithm. We state the following assumption on the initial conditions and feasibility that will be used to analyze the algorithm convergence.

Algorithm 2 EKF Predictor

Require: prediction horizon N; model and covariance matrices $A(t), Q(t), R(t), H(t)$ for all $t \geq 0$, covariance matrices $P_m(0), P_p(0)$;

function PREDICT
  $\hat{\boldsymbol{x}}_p(t) = f(\hat{\boldsymbol{x}}_m(t-1))$
  $P_p(t) = A(t-1)P_m(t-1)A^T(t-1) + Q(t-1)$
  return $\hat{\boldsymbol{x}}_p(t)$, $P_p(t)$
function UPDATE($\boldsymbol{z}(t)$)
  $K(t) = P_p(t)H^T(t)(H(t)P_p(t)H^T(t) + R(t))^{-1}$
  $\hat{\boldsymbol{x}}_m(t) = \hat{\boldsymbol{x}}_p(t) + K(t)(\boldsymbol{z}(t) - H(t)\hat{\boldsymbol{x}}_p(t))$
  $P_m(t) = (I - K(t)H(t))P_p(t)$
  return $\hat{\boldsymbol{x}}_m(t)$, $P_m(t)$
function PREDICT TRAJECTORY(N)
  $\hat{\boldsymbol{x}}_p(k|t) = \hat{\boldsymbol{x}}_m(t)$
  $P_p(k|t) = P_m(t)$
  for $k = 0$ to N **do**
    $\hat{\boldsymbol{x}}_p(k+1|t) = f(\hat{\boldsymbol{x}}_p(k|t))$
    $P_p(k+1|t) = A(k|t)P_p(k|t)A^T(k|t) + Q(k|t)$
  return $\hat{\boldsymbol{x}}_p(\cdot|t)$, $P_p(\cdot|t)$
while not landed **do**
  if new measurement $\boldsymbol{z}(t)$ received **then**
    $\hat{\boldsymbol{x}}_p(t)$, $P_p(t) \leftarrow$ PREDICT() ▷ prediction step
    $\hat{\boldsymbol{x}}_m(t)$, $P_m(t) \leftarrow$ UPDATE($\boldsymbol{z}(t)$) ▷ update measurement step
    $\hat{\boldsymbol{x}}_p(\cdot|t)$, $P_p(\cdot|t) \leftarrow$ PREDICT TRAJECTORY(N) ▷ trajectory prediction

Assumption 6.1. All agents at time t_0 have initial conditions $\boldsymbol{x}_i(t_0)$, $i \in \mathcal{N}$ such that

$$\boldsymbol{x}_i(t_0) \in \mathcal{X}_{i,j}(\boldsymbol{z}_j(t_0)) \quad \text{for } i \neq l, j \in \mathcal{N}_{-i}.$$

Moreover, it holds that

$$\|\boldsymbol{c}_i - \boldsymbol{c}_j\| > R, \text{ for all } i, j \in \mathcal{M}, i \neq j.$$

Before we state the algorithm, let us introduce a data collection $D_i(t)$ that is available to an agent i at time step t as $D_i(t) = \{\hat{\boldsymbol{z}}_j(\cdot|t_j)\}_{j\in\mathcal{N}_{-i}}$ consisting of the shared future trajectories of all other agents broadcast at time $t_j < t, \forall j \in \mathcal{N}_{-i}$.

Remark 6.1. An issue that can occur in the multi-agent case with inter-agent collision avoidance in general, is that the agents can end up in a deadlock and be prevented to effectively find a way to navigate to the goal position. In that case, the deadlock can be resolved by forcing the agents to solve Problem 6.2 sequentially [47, 103]. This would guarantee that each agent takes into account the current predicted trajectory of other agents and waits until the process is completed. Thus,

Algorithm 3 Multiple-Follower Rendezvous Algorithm

Require: initial states $\boldsymbol{x}_i(0)$ at time $t = 0$, landing locations $\boldsymbol{c}_i$ according to Assumption 6.1 for $i \in \mathcal{M}$, a tolerance parameter ε;

1: **for** each agent $i \in \mathcal{M}$ **do**
2: update $D_i(t)$ and $\hat{\boldsymbol{z}}_j(\cdot|t)$, $j \in \mathcal{N}_{-i}$ using (6.14)
3: $\hat{\boldsymbol{x}}_i^*(\cdot|t), \hat{\boldsymbol{u}}_i^*(\cdot|t) \leftarrow$ solve Problem 6.2
4: **if** $\|\hat{\boldsymbol{x}}_i(t) - \hat{\boldsymbol{z}}_l(t) + \boldsymbol{c}_i\| > \varepsilon$ **then**
5: apply $\hat{\boldsymbol{u}}_i^*(0|t)$
6: broadcast $\hat{\boldsymbol{x}}_i^*(\cdot|t)$
7: $t \leftarrow t + 1$

the generated trajectories will not end up in a deadlock. Note that communication in this case is required. Therefore, Algorithm 3 requires that the first iteration of the algorithm is done in a sequential manner and that the initial feasibility is established.

6.7 Convergence Analysis

Let us consider the one-follower case of the rendezvous landing problem, $\mathcal{N} = \{l, f\}$, in which the leader reference $\boldsymbol{x}_{r,l}(t)$ is given, and the follower reference is based on the position of the leader $\boldsymbol{x}_{r,f}(t) = \boldsymbol{z}_l(t)$ that corresponds to the center of the landing platform. In order to analyze system behavior, we introduce an assumption on the follower's capability to track the leader with respect to the leader dynamics and spatiotemporal constraints.

Assumption 6.2. There exists a control law $\kappa : \mathcal{X}_f \times \mathcal{X}_f \to \mathcal{U}_f$ such that

$$\|\boldsymbol{x}_f^+ - \boldsymbol{z}_l^+\|^2 \leq \rho \|\boldsymbol{x}_f - \boldsymbol{z}_l\|^2 \tag{6.13}$$

with $\rho \in (0, 1)$ and

$$\begin{aligned} \boldsymbol{x}_f^+ &= f_f(\boldsymbol{x}_f, \kappa(\boldsymbol{x}_f, \boldsymbol{z}_l)) \\ \boldsymbol{x}_l^+ &= f_l(\boldsymbol{x}_l, \boldsymbol{u}_l), \\ \boldsymbol{z}_l &= H x_l, \\ h_C(\boldsymbol{x}_f, \boldsymbol{z}_l) &\geq 0 \end{aligned}$$

for all $\boldsymbol{x}_f \in \mathcal{X}_f, \boldsymbol{x}_l \in \mathcal{X}_l$, and $\boldsymbol{u}_l \in \mathcal{U}_l$.

Assumption 6.2 states that for any control action the leader takes, there exists a control law for the follower that will reduce the distance between them in every consecutive time step. However, in the case of communication losses, the follower

must be capable to asymptotically reduce the distance regardless of the leader's behaviour.

Furthermore, Assumption 6.2 is similar to Assumptions 4 and 6 in [110] from which the notion of incremental stability as in [111, Def. 2.1],[112] can be elaborated. Compared to [110], the reference $\boldsymbol{z}_l$ in our case has different dynamics than the followers' and it does not take into account the control action of the leader.

Referring to Problem 6.2, let the value function at time step t be

$$\begin{aligned} V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) &= \min_{\boldsymbol{u}_f(\cdot|t)} J_f(\hat{\boldsymbol{x}}_f(\cdot|t), \boldsymbol{u}_f(\cdot|t), \hat{\boldsymbol{z}}_l(\cdot|t), N, t) \\ &= \sum_{k=0}^{N-1} \left\| \hat{\boldsymbol{x}}_f^*(k|t) - \hat{\boldsymbol{z}}_l(k|t) \right\|_{Q_f}^2 \end{aligned}$$

We define the region of attraction $\mathcal{X}_f^{ROA}(\boldsymbol{z}_l) := \{\boldsymbol{x}_f \in \mathcal{X}_f : V_N(\boldsymbol{x}_f, \boldsymbol{z}_l) \leq V_{N,max}\}$ of the MPC controller as the set of states which can be steered to the desired leader state $\boldsymbol{z}_l$ in N or fewer steps.

The convergence result for the case of one follower is based on ensuring that for all initial states in the region of attraction, the value function is a Lyapunov function decreasing at each time step.

Theorem 6.1 (Convergence with one follower)**.** *Let Assumption 6.2 hold. For any $V_{N,max} \in \mathbb{R}_{>0}$, there exist constants $\gamma \geq 1$, and $N_0 \in \mathbb{N}$, such that for all $N > N_0$ and all initial conditions in the region of attraction $\boldsymbol{x}_f(0) \in \mathcal{X}_f^{ROA}(\boldsymbol{z}_l(0))$, there exists $\alpha_N \in \mathbb{R}_{>0}$ such that the multi-agent system* (6.1)*, with $\boldsymbol{w}_i = 0$, $i \in \mathcal{N} = \{l, f\}$ satisfies*

$$\|\boldsymbol{x}_f - \boldsymbol{z}_l\|_{Q_f}^2 \leq V_N(\boldsymbol{x}_f, \boldsymbol{z}_l) \leq \gamma \|\boldsymbol{x}_f - \boldsymbol{z}_l\|_{Q_f}^2,$$

$$V_N(\boldsymbol{x}_f^+, \boldsymbol{z}_l^+) - V_N(\boldsymbol{x}_f, \boldsymbol{z}_l) \leq -\alpha_N \|\boldsymbol{x}_f - \boldsymbol{z}_l\|_{Q_f}^2,$$

for all $t \geq 0$. Furthermore, for all $\boldsymbol{x}_f(0) \in \mathcal{X}_f^{ROA}(\boldsymbol{z}_l(0))$ the follower converges to the leader-dependent rendezvous location exponentially, i.e. $\boldsymbol{x}_f(t) \to \boldsymbol{z}_l(t)$ as $t \to \infty$.

The proof is given in Appendix 6.A.

Remark 6.2. Note that the optimization problem in Problem 6.2 does not use the terminal ingredients and thus they are not used in Theorem 6.1, although it is a common way to prove the stability of MPC scheme [51, 78]. In this chapter, we avoid usage of the terminal ingredients by considering a sufficiently long planning horizon N in the region of attraction $\mathcal{X}^{ROA}$. This builds upon the methodology suggested in [110, 113]. Moreover, the initial feasibility in the region of attraction is implicitly assumed in Theorem 6.1 by the same principle, i.e. by assuming that there exists N_0 for which the optimization problem in Prob. 6.2 is feasible for all initial conditions in $\mathcal{X}^{ROA}$. Also note that the disturbances are not considered in the theorem.

It is straightforward to show, using the same argument as in Theorem 6.1, that the leader agent will follow its reference trajectory as well, thus achieving the objective of Problem 6.1.

6.7.1 Robustness to Communication Losses

In case of communication loss, the latest available shared trajectory of the leader used as the follower reference is shifted and the missing part of the trajectory is predicted with the EKF predictor. If the latest time of arrival t_a of the shared trajectory is $t_a < t - k$, $0 < k < N$, then the trajectory is shifted for k time steps and the rest is predicted as follows

$$\hat{\boldsymbol{z}}_i(l|t) = \begin{cases} \hat{\boldsymbol{z}}_i(l+k|t_{a,i}), & \text{for } l < N-k \\ \hat{\boldsymbol{z}}_{p,i}(l+k|t_{a,i}), & \text{for } l \geq N-k \end{cases} \tag{6.14}$$

for $l = 0, 1, ..., N$. However, the predicted trajectory has some uncertainty such that $\boldsymbol{z}_i(l|t) \in \hat{\boldsymbol{z}}_{p,i}(l|t_i) \oplus P_i(l|t)$ where the set $P_i(l|t) := \{b : \|b\|^2_{P_{p,i}(l|t)} \leq s\}$ is determined by the prediction covariance matrix $P_{p,i}(l|t)$ and a parameter $s = -2\ln(1-p)$, that depends on the chosen probability confidence $p \in (0,1)$. In case $k = N-1$ which means that no predicted future steps are available, the data collection is updated only with the EKF Predictor.

Remark 6.3. Note that the next state uncertainty can be estimated without using EKF Predictor as $\boldsymbol{z}_i(l+1|t) \in \hat{\boldsymbol{z}}_i(l|t_i) \oplus \mathcal{B}_{r_i}$ where the choice of the safety radius r_i determines the conservativeness of the used set estimates. By setting r_i to

$$r_i = \max_{\boldsymbol{x}_i \in \mathcal{X}_i, \boldsymbol{u}_i \in \mathcal{U}_i} \|\boldsymbol{x}_i^+ - f(\boldsymbol{x}_i, \boldsymbol{u}_i)\| \tag{6.15}$$

one can guarantee that the next state is within the ball of the given radius. However, this is an overly conservative approach given that the uncertainty sets grow and can become very large at the end of the horizon thus preventing the follower agents to land. In that case, the collision checking can be restricted only for the one-step ahead prediction, i.e.

$$\|C(\hat{\boldsymbol{x}}_i(1|t) - \hat{\boldsymbol{z}}_j(1|t))\| \geq R + r_j, \quad \text{for } j \in \mathcal{N}_{-i}$$

and use the worst case radius r_j as in (6.15).

Theorem 6.2 (p-probabilistically safe landing)**.** *Let the conditions of Theorem 6.1 hold. Given the probabilistic confidence $p \in (0,1)$, the radius of the landing platform r, and the radius necessary for the safe landing r_{safe}, if the following condition holds*

$$r_{safe} + \sqrt{s\lambda_{\max}(P_{p,i}(t))} < r$$

where $s = -2\ln(1-p)$, then the landing is considered as probabilistically safe with probability p.

Proof. The proof is based on the worst-case estimate of the landing position. Given the probabilistic confidence p and the covariance matrix $P_{p,i}(t)$, the worst-case distance from the actual landing position $\boldsymbol{z}_l(t)$ and its estimate $\hat{\boldsymbol{z}}_l(t)$ is $d = \sqrt{s\lambda_{\max}(P_{p,i}(t))}$. Thus, $\|\boldsymbol{z}_l(t) - \hat{\boldsymbol{z}}_l(t)\| = d < r - r_{safe}$ which means that the follower applying the control input obtained with Prob. 6.2 is guaranteed to land inside of the landing platform with probability p. □

The extension to the multiple-follower case is given by the following result:

Corollary 6.1. *Let Assumption 6.2 and 6.1 hold. Moreover, let the conditions of Theorem 6.2 hold for all follower agents $i \in \mathcal{M}$. Then, Algorithm 3 converges and all follower agents meet on the leader landing platform without a collision.*

Proof. The proof is based on two parts. First, that the all agents converge to the landing platform and second, that their trajectories are collision-free. Given that all stated conditions hold there exists a feasible configuration for all follower agents to rendezvous on the leader landing platform. Moreover, all follower agents also satisfy the conditions from Theorem 6.2, and thus there exist a feasible landing trajectory robust to communication losses. Because each agent is solving the optimization problem in Problem 6.2, the inter-agent collision avoidance is enforced in every feasible landing trajectory. Therefore, all agents rendezvous on the leader landing platform without a collision. □

6.8 Simulation Results

In this section we present a landing scenario with $M = 6$ agents. The leader is unable to communicate with the follower agents and thus follower agents must use EKF Predictor to estimate the position of the leader. Moreover, the leader measurements are taken from real-world experiments and thus have disturbances. The follower agents share their predicted trajectories until time step $k = 10$ (continuous time $t_c = 2s$) when one of the agents (Agent f_1) also loses the communication with the rest of the agents. Then the rest of the agents in the scenario must also predict the future trajectory of Agent f_1 and Agent f_1 predicts the trajectories of all other agents in the scenario.

The state and input constraints on the models are defined similarly to 5.4.1. The initial positions of follower agents are $\boldsymbol{x}_i(0) = [5\cos(2i\pi/M), 5\sin(2i\pi/M), 10, \mathbf{0}_6^T]^T$, and the leader is at origin $\boldsymbol{x}_l(0) = \mathbf{0}_6$. The radius of the whole landing platform for all agents is $5r_{safe}$, and the safe radius for landing is $r_{safe} = 0.5m$. The matrices $Q_{f_i} = \mathrm{diag}(10, 10, 5, 1, 1, 1, 1, 1, 1)$, $Q_l = \mathrm{diag}(10, 10, 10, 1, 1, 1)$, thus primary penalizing the position in x and y and then in z for quadrotors, and orientation ψ for the boat. $\lambda_{\max}(Q_{f_i}) = 10$, and we pick $V_{N,max} = 240$ such that $\bar{\gamma} = 1.99$ and $N = 20$, and the region of attraction is large enough and encompass the initial displacement with a sufficient margin.

The landing locations are equidistantly distributed as on Fig. 6.3 and assigned to an agent positioned at the opposite side diagonally. Assumption 6.1 is satisfied

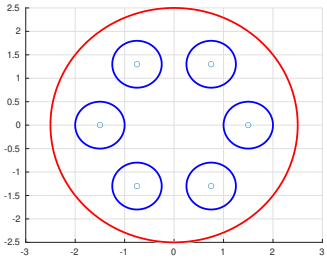

Figure 6.3: The red circle marks the boundary of the landing platform, while the blue circles are the landing locations with safety radius $r_{safe} = 0.5m$. The distance from the center of the platform to a landing location is $3r_{safe}$.

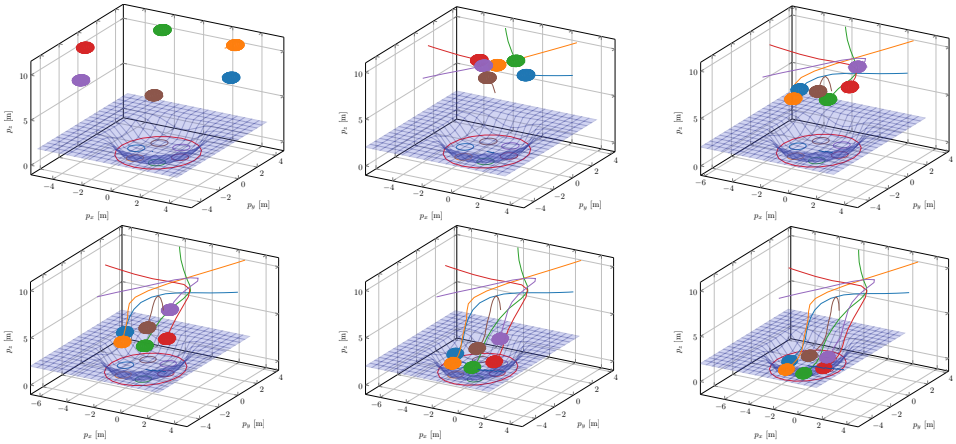

Figure 6.4: 3D view of a landing scenario at discrete time steps $t = \{0, 5, 10, 13, 17, 23\}$ with $\Delta t = 0.2s$ between two time steps. The followers are unable to communicate with the leader and a loss of communication with Agent f_1 occurs during the experiment at $t = 10$. Agent f_1 is marked in orange.

with $R = 2r_{safe}$, and $C = I_3$. The landing is considered safe if the conditions of Theorem 6.2 hold with $p = 0.95$. Assuming that all agents will have the identical estimation of the landing platform position and inter-agent collisions are handled it is sufficient to consider the safety with respect to the outer boundary. Thus, with $p = 0.95$, $\lambda_{max}(P_{p,l}(t)) < (5r_{safe} - 3r_{safe})^2/s \approx 0.17$. From the experimental results, $\lambda_{max}(P_{p,l}(t)) < 0.06$ thus satisfying Theorem 6.2. Algorithm 3 and Prob. 6.2 are implemented with CasADi [114] and results are shown on Figures 6.4 and 6.5. All trajectories are collision-free, and the predicted trajectory of Agent f_1 by other agents and vice versa do not induce much conservativeness to Algorithm. This is mainly because the first part of the trajectory is generated using the shift mechanism as in (6.14) and the small eigenvalues of the covariance matrices compared to the considered safety radii.

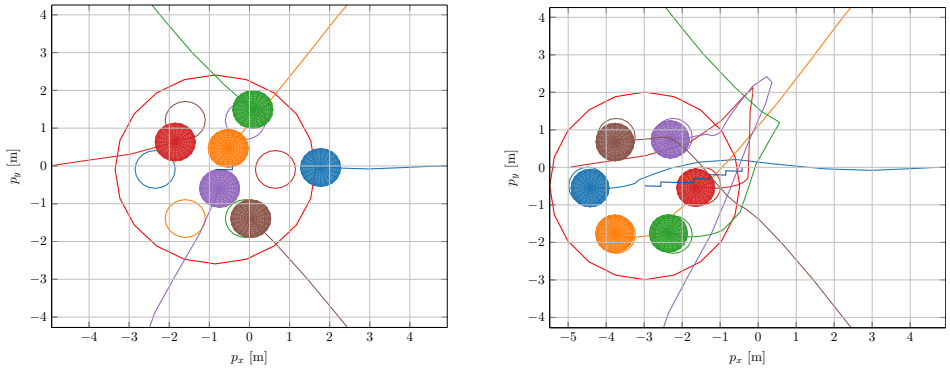

Figure 6.5: Left: Top view of the second situation at $t = 5$, when it might look from the perspective view that the agents are too close, shows that the inter-agent collision avoidance constraint is enforced. Right: Final situation at $t = 23$. h_C is removed for better visibility.

6.9 Conclusion

In this chapter, we presented a rendezvous algorithm based on the leader-follower scheme and distributed MPC with robustness to communication losses. The algorithm is designed for autonomous landing of multiple quadrotors on moving unmanned surface vehicles. We introduced a novel formulation of the safety constraints for the autonomous landing scenario and proved the convergence theorem under nominal conditions without the need for terminal ingredients. Moreover, the scheme is complemented with EKF Predictor that can enable the landing on the leader landing platform even in the case when there is no communication between the leader and follower.

The convergence analysis of the algorithm is presented and the effectiveness of the proposed algorithm is demonstrated with the simulation of a landing scenario. In the future work, we aim to include the disturbances in the analysis and quantify the upper bounds such that the convergence is preserved. Moreover, it will be interesting to include learning methods in order to speed-up the computation.

6.A Proof of Theorem 6.1

Proof. The proof proceeds in two parts. The first part shows the boundedness of $V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t))$ for all $t \geq 0$, and the second part proves that the value function decreases for all states in the region of attraction at every time step.

By the definition of the value function

$$V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) \geq \|\hat{\boldsymbol{x}}_f(0|t) - \hat{\boldsymbol{z}}_l(0|t)\|^2_{Q_f} = \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|^2_{Q_f}$$

Moreover, for $\boldsymbol{x}_f(t) \in \mathcal{X}_f^{ROA}(\boldsymbol{z}_l)$

$$\|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|^2_{Q_f} \leq V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) \leq V_{N,max}$$

thus, there exists $\bar{\gamma} \geq 1$ for which $\bar{\gamma}\,\|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|^2_{Q_f} = V_{N,max}$ and $\gamma \geq \bar{\gamma} \geq 1$ such that

$$V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) \leq V_{N,max} \leq \gamma \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|^2_{Q_f}$$

The second part of the proof uses the contraction of the error from Assumption 6.2. Let a feasible (suboptimal) input sequence for the next time step be $\tilde{\boldsymbol{u}}_f(\cdot|t+1)$, defined as

$$\tilde{\boldsymbol{u}}_f(k|t+1) = \begin{cases} \boldsymbol{u}_f^*(k+1|t) & \text{for } k = 0, 1, ..., N-1 \\ \kappa(\hat{\boldsymbol{x}}_f^*(N|t), \hat{\boldsymbol{z}}_l(N|t)) & \text{for } k = N \end{cases}$$

consisting of the shifted optimal input from the previous step and some appended $\boldsymbol{u}_f^N = \kappa(\hat{\boldsymbol{x}}_f^*(N|t), \hat{\boldsymbol{z}}_l(N|t)) \in \mathcal{U}_f$ that satisfies Assumption 6.2. Assuming there are no disturbances, then $\boldsymbol{x}_f(t+1) = \tilde{\boldsymbol{x}}_f(0|t+1) = f_f(\boldsymbol{x}_f(t), \boldsymbol{u}_f^*(0|t))$ and

$$\tilde{\boldsymbol{x}}_f(k|t+1) = \begin{cases} \hat{\boldsymbol{x}}_f^*(k+1|t) & \text{for } k = 0, 1, ..., N-1 \\ f_f(\hat{\boldsymbol{x}}_f^*(N|t), \kappa(\hat{\boldsymbol{x}}_f^*(N|t), \hat{\boldsymbol{z}}_l(N|t))) & \text{for } k = N \end{cases}$$

Moreover, due to Assumption 6.2

$$\|\tilde{\boldsymbol{x}}_f(N|t+1) - \hat{\boldsymbol{z}}_l(N|t+1)\|^2 \leq \rho \left\|\hat{\boldsymbol{x}}_f^*(N|t) - \hat{\boldsymbol{z}}_l(N|t)\right\|^2$$

and

$$\|\tilde{\boldsymbol{x}}_f(N|t+1) - \hat{\boldsymbol{z}}_l(N|t+1)\|^2_{Q_f} \leq \rho \frac{\lambda_{max}(Q_f)}{\lambda_{min}(Q_f)} \left\|\hat{\boldsymbol{x}}_f^*(N|t) - \hat{\boldsymbol{z}}_l(N|t)\right\|^2_{Q_f}$$

Because $V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) \leq \gamma \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|^2_{Q_f}$, then $\forall\ k\ \geq\ 1$, there exist $N > 1$ and $\gamma \geq \bar{\gamma} \geq 1$ such that

$$\|\hat{\boldsymbol{x}}_f^*(k|t) - \hat{\boldsymbol{z}}_l(k|t)\|^2_{Q_f} \leq \frac{V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t))}{N} \leq \frac{\gamma}{N} \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|^2_{Q_f}$$

Let us consider the the value function at time step $t+1$, then

$$
\begin{aligned}
&V_N(\boldsymbol{x}_f(t+1), \boldsymbol{z}_l(t+1)) \\
&\quad \leq \sum_{k=0}^{N} \|\tilde{\boldsymbol{x}}_f(k|t+1) - \hat{\boldsymbol{z}}_l(k|t+1)\|_{Q_f}^2 \\
&\quad = \sum_{k=0}^{N-1} \|\tilde{\boldsymbol{x}}_f(k|t+1) - \hat{\boldsymbol{z}}_l(k|t+1)\|_{Q_f}^2 + \|\tilde{\boldsymbol{x}}_f(N|t+1) - \hat{\boldsymbol{z}}_l(N|t+1)\|_{Q_f}^2 \\
&\quad = \sum_{k=1}^{N} \left\|\hat{\boldsymbol{x}}_f^*(k|t) - \hat{\boldsymbol{z}}_l(k|t)\right\|_{Q_f}^2 + \|\tilde{\boldsymbol{x}}_f(N|t+1) - \hat{\boldsymbol{z}}_l(N|t+1)\|_{Q_f}^2 \\
&\quad = \sum_{k=0}^{N} \left\|\hat{\boldsymbol{x}}_f^*(k|t) - \hat{\boldsymbol{z}}_l(k|t)\right\|_{Q_f}^2 - \left\|\hat{\boldsymbol{x}}_f^*(0|t) - \hat{\boldsymbol{z}}_l(0|t)\right\|_{Q_f}^2 \\
&\qquad + \|\tilde{\boldsymbol{x}}_f(N|t+1) - \hat{\boldsymbol{z}}_l(N|t+1)\|_{Q_f}^2 \\
&\quad = V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) - \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|_{Q_f}^2 + \|\tilde{\boldsymbol{x}}_f(N|t+1) - \hat{\boldsymbol{z}}_l(N|t+1)\|_{Q_f}^2 \\
&\quad \leq V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) - \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|_{Q_f}^2 + \rho \frac{\gamma}{N} \frac{\lambda_{max}(Q_f)}{\lambda_{min}(Q_f)} \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|_{Q_f}^2 \\
&\quad \leq V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t)) - \alpha_N \|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|_{Q_f}^2
\end{aligned}
$$

where $\alpha_N := 1 - \rho \frac{\gamma}{N} \frac{\lambda_{max}(Q_f)}{\lambda_{min}(Q_f)}$. Thus by choosing $N > N_0 := \bar{\gamma} \frac{\lambda_{max}(Q_f)}{\lambda_{min}(Q_f)}$, $\alpha_N > 0$. Finally, the value function $V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t))$ is decreasing for all $t \geq 0$.

Using the decrease property and the boundedness in the region of attraction $\mathcal{X}_f^{ROA}(\boldsymbol{z}_l)$ proven in the first part, $V_N(\boldsymbol{x}_f(t), \boldsymbol{z}_l(t))$ is a Lyapunov function in $\mathcal{X}_f^{ROA}(\boldsymbol{z}_l)$. Thus, the error $\|\boldsymbol{x}_f(t) - \boldsymbol{z}_l(t)\|$ exponentially goes to zero, which concludes the proof. □

Chapter 7

Summary and Future Research Directions

In the thesis, we have investigated trajectory tracking and prediction-based rendezvous for the unmanned aerial and surface vehicles. In particular we have focused on the topics of prescribed performance control, model predictive control and distributed control.

In Chapter 3 we have shown an effective way to use prescribed performance control for an underactuated dynamical system such as the UAV. We provided the theoretical guarantees for the trajectory tracking with PPC and numerical simulations.

PPC is also applied to USV, another underactuated system of interest for this thesis, in Chapter 4 as a part of framework called Kinodynamic motion-planning via funnel control (KDF). Moreover, we investigated the control input saturations that can occur due to the physical limitations of the systems. We have proven that the stability is preserved under certain conditions. Furthermore, the control framework is applied and tested in the real-world open-sea experiments. After analyzing the experimental results, we proposed an enhanced trajectory planning algorithm based on RRT and B-splines that generate smooth trajectories with respect to the kinodynamic constraints. The enhanced framework is tested in the simulations and its advantages over KDF are verified.

The second part of the thesis is devoted to the prediction-based rendezvous algorithm with focus on a specific application of autonomous landing. In Chapter 5 we focus on an event-triggered distributed MPC algorithm with minimal shared data consisting only of the rendezvous position. The recursive feasibility of the algorithm is shown using the moving terminal sets. The numerical simulations show that the agents are able to rendezvous and execute the landing. We noticed that the performance of the algorithm is better when the terminal sets are not enforced in the optimization problem. Thus, we posed the optimization problem without enforcing the terminal sets and proved the convergence of the algorithm.

The leader-follower scheme is investigated in Chapter 6. This approach is chosen due to the specific spatiotemporal constraints that the boat (the leader) is imposing on the quadrotor movement (the follower). We proposed a novel formulation of the

constraints and proved the convergence of followers to the landing platform on the boat, when the boat is following its own predefined trajectory. Since during the experiments we noticed that communication losses can occasionally happen, we equipped the follower agents with EKF Predictor to predict the future trajectories of all other agents in the scenario. This case is examined in the simulations using the real data from the boat. Moreover, the inter-agent collision avoidance condition is added to the optimization problem and we examined how the communication loss can affect the feasibility of the algorithm. Thus, the proposed algorithm has safety measures to handle those cases.

7.1 Possible Future Directions

In the first two chapters we focused on PPC for specific underactuated systems. In this cases we leveraged the known dynamics but considered that all parameters are unkown. However, it might be of interest to design a general control framework for general underactuated systems with PPC.

When it comes to the trajectory generation with respect to the kinodynamical constraints that is covered in Section 4.8, formulating the optimization problem 4.1 in a receding horizon fashion with shorter planning horizon might be interesting and relevant. The collision avoidance criteria used in this problem are linear compared to the second-order constraints used in Section 6.4 for the inter-agent collision avoidance in our multiple-follower formulation of DMPC. Moreover, this approach that does not require the usage of dynamics but only kinodynamical limitations of an agent can potentially generate the trajectories at a faster rate than the nonlinear MPC used in our optimization problems in the second part of the thesis. Then, a low-level controller such as PPC can be used to guarantee that an agent will remain in some neighborhood of the reference trajectory.

In Chapter 6, the trajectory predictor can be potentially improved using machine learning or conformal prediction methods. The learning methods can eventually be used to learn disturbances or even the behaviour of the agent with MPC in order to speed-up the computation. Moreover, the resolution of potentially unwanted behaviours such as deadlocks that occur in the parallel execution of the proposed algorithm can be of interest to further investigate.

On a more general level, a possible future direction can include using Signal-temporal logic (STL) or delegation framework for mission planning and coverage problems which can be then coupled with our developed algorithms for prediction-based control, trajectory generation and tracking. This direction would eventually enable more autonomous missions of unmanned agents operating in multiple domains.

Furthermore, extension of the proposed algorithms to account for disturbances and creating additional safety measures in the scenarios that are not examined in this thesis might be of interest. Finally, creating more experimental scenarios and use cases of the proposed algorithms and conducting more real-world experiments.

Bibliography

[1] EASA. Easy access rules for unmanned aircraft systems, 2022.

[2] Jeremi Gancet, Gautier Hattenberger, Rachid Alami, and Simon Lacroix. Task planning and control for a multi-uav system: architecture and algorithms. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1017–1022. IEEE, 2005.

[3] SAE On-Road Automated Vehicle Standards Committee et al. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J*, 3016:1–16, 2014.

[4] Tuna Toksoz, Joshua Redding, Matthew Michini, Matthew Vavrina, John Vian, Bernard J Michini, and Jonathan P How. Automated battery swap and recharge to enable persistent uav missions. 2011.

[5] Gennaro Notomista and Magnus Egerstedt. Persistification of robotic tasks. *IEEE Transactions on Control Systems Technology*, 29(2):756–767, 2020.

[6] Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control–a review. *Journal of process control*, 19(5):723–731, 2009.

[7] Samir Bouabdallah, Andre Noth, and Roland Siegwart. Pid vs lq control techniques applied to an indoor micro quadrotor. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2451–2456. IEEE, 2004.

[8] Teppo Luukkonen. Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 22:22, 2011.

[9] Jun Li and Yuntang Li. Dynamic analysis and pid control for a quadrotor. In *2011 IEEE International Conference on Mechatronics and Automation*, pages 573–578. IEEE, 2011.

[10] Lucas M Argentim, Willian C Rezende, Paulo E Santos, and Renato A Aguiar. Pid, lqr and lqr-pid on a quadcopter platform. In *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 1–6. IEEE, 2013.

[11] Nguyen Khoi Tran, Eitan Bulka, and Meyer Nahon. Quadrotor control in a wind field. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 320–328. IEEE, 2015.

[12] Philipp Foehn and Davide Scaramuzza. Onboard state dependent lqr for agile quadrotors. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6566–6572. IEEE, 2018.

[13] Tarek Madani and Abdelaziz Benallegue. Backstepping control for a quadrotor helicopter. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3255–3260, 2006.

[14] Ahmed Aboudonia, Ayman El-Badawy, and Ramy Rashad. Active anti-disturbance control of a quadrotor unmanned aerial vehicle using the command-filtering backstepping approach. *Nonlinear Dynamics*, 90(1):581–597, 2017.

[15] Samir Bouabdallah and Roland Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2247–2252. IEEE, 2005.

[16] Krittaya Runcharoon and Varawan Srichatrapimuk. Sliding mode control of quadrotor. In *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, pages 552–557. IEEE, 2013.

[17] Brett T Lopez, Jean-Jacques Slotine, and Jonathan P How. Robust collision avoidance via sliding control. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2962–2969. IEEE, 2018.

[18] Mu Huang, Bin Xian, Chen Diao, Kaiyan Yang, and Yu Feng. Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping. *American Control Conference (ACC)*, pages 2076–2081, 2010.

[19] Buddy Michini and Jonathan How. L1 adaptive control for indoor autonomous vehicles: Design process and flight testing. In *AIAA Guidance, Navigation, and Control Conference*, page 5754, 2009.

[20] Simone Formentin and Marco Lovera. Flatness-based control of a quadrotor helicopter via feedforward linearization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6171–6176. IEEE, 2011.

[21] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar. Geometric control and differential flatness of a quadrotor uav with a cable-suspended load. In *52nd IEEE Conference on Decision and Control*, pages 2269–2274. IEEE, 2013.

[22] Guilherme V Raffo, Manuel G Ortega, and Francisco R Rubio. An underactuated $\mathcal{H}_\infty$ control strategy for a quadrotor helicopter. *European Control Conference (ECC)*, pages 3845–3850, 2009.

[23] Mina Kamel, Michael Burri, and Roland Siegwart. Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles. *IFAC-PapersOnLine*, 50(1):3463–3469, 2017.

[24] Huan Nguyen, Mina Kamel, Kostas Alexis, and Roland Siegwart. Model predictive control for micro aerial vehicles: A survey. In *2021 European Control Conference (ECC)*, pages 1556–1563. IEEE, 2021.

[25] Nitin Sydney, Brendan Smyth, and Derek A Paley. Dynamic control of autonomous quadrotor flight in an estimated wind field. In *52nd IEEE Conference on Decision and Control*, pages 3609–3616. IEEE, 2013.

[26] Andrea Tagliabue, Aleix Paris, Suhan Kim, Regan Kubicek, Sarah Bergbreiter, and Jonathan P How. Touch the wind: Simultaneous airflow, drag and interaction sensing on a multirotor. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1645–1652. IEEE, 2020.

[27] T.I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley, 1994.

[28] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[29] António Pedro Aguiar and Joao Pedro Hespanha. Position tracking of underactuated vehicles. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 3, pages 1988–1993. IEEE, 2003.

[30] A Pedro Aguiar and Joao P Hespanha. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE transactions on automatic control*, 52(8):1362–1379, 2007.

[31] Roger Skjetne and Thor I Fossen. Nonlinear maneuvering and control of ships. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, volume 3, pages 1808–1815. IEEE, 2001.

[32] Khac D Do and Jie Pan. State-and output-feedback robust path-following controllers for underactuated ships using serret–frenet frame. *Ocean Engineering*, 31(5-6):587–613, 2004.

[33] KY Pettersen and Henk Nijmeijer. Tracking control of an underactuated surface vessel. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, volume 4, pages 4561–4566. IEEE, 1998.

[34] Alexey Pavlov, Håvard Nordahl, and Morten Breivik. Mpc-based optimal path following for underactuated vessels. *IFAC Proceedings Volumes*, 42(18):340–345, 2009.

[35] Andrea Alessandretti, A Pedro Aguiar, and Colin N Jones. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In *2013 european control conference (ecc)*, pages 1371–1376. IEEE, 2013.

[36] Mohamed Abdelaal, Martin Fränzle, and Axel Hahn. Nonlinear model predictive control for trajectory tracking and collision avoidance of underactuated vessels with disturbances. *Ocean Engineering*, 160:168–180, 2018.

[37] Andreas B Martinsen, Anastasios M Lekkas, and Sebastien Gros. Autonomous docking using direct optimal control. *IFAC-PapersOnLine*, 52(21):97–102, 2019.

[38] Nan Gu, Dan Wang, Zhouhua Peng, Tieshan Li, and Shaocheng Tong. Model-free containment control of underactuated surface vessels under switching topologies based on guiding vector fields and data-driven neural predictors. *IEEE Transactions on Cybernetics*, 2021.

[39] Charalampos P Bechlioulis, George C Karras, Shahab Heshmati-Alamdari, and Kostas J Kyriakopoulos. Trajectory tracking with prescribed performance for underactuated underwater vehicles under model uncertainties and external disturbances. *IEEE Transactions on Control Systems Technology*, 25(2):429–440, 2016.

[40] Christos K Verginis, Dimos V Dimarogonas, and Lydia E Kavraki. Kdf: Kinodynamic motion planning via geometric sampling-based algorithms and funnel control. *arXiv preprint arXiv:2104.11917*, 2021.

[41] Panagiotis Vlantis, Panos Marantos, Charalampos P Bechlioulis, and Kostas J Kyriakopoulos. Quadrotor landing on an inclined platform of a moving ground vehicle. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2202–2207. IEEE, 2015.

[42] Linnea Persson and Bo Wahlberg. Model predictive control for autonomous ship landing in a search and rescue scenario. In *AIAA Scitech 2019 Forum*, page 1169, 2019.

[43] Aleix Paris, Brett T Lopez, and Jonathan P How. Dynamic landing of an autonomous quadrotor on a moving platform in turbulent wind conditions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9577–9583. IEEE, 2020.

[44] Dženan Lapandić, Linnea Persson, Dimos V. Dimarogonas, and Bo Wahlberg. Aperiodic communication for mpc in autonomous cooperative landing. *IFAC-PapersOnLine*, 54(6):113–118, 2021. 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021.

[45] Linnea Persson and Bo Wahlberg. Variable prediction horizon control for cooperative landing on moving target. In *2021 IEEE Aerospace Conference*. IEEE, 2021.

[46] Robert Bereza, Linnea Persson, and Bo Wahlberg. Distributed model predictive control for cooperative landing. In *2020 Proceedings of IFAC World Congress*. IFAC, 2020.

[47] Panagiotis D Christofides, Riccardo Scattolini, David Munoz de la Pena, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[48] TamáS Keviczky, Francesco Borrelli, and Gary J Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

[49] William B Dunbar and Richard M Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[50] Matthias A Müller, Marcus Reble, and Frank Allgöwer. Cooperative control of dynamically decoupled systems via distributed model predictive control. *International Journal of Robust and Nonlinear Control*, 22(12):1376–1397, 2012.

[51] Hong Chen and Frank Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[52] WPMH Heemels, Karl Henrik Johansson, and Paulo Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3270–3285. IEEE, 2012.

[53] Kazumune Hashimoto, Shuichi Adachi, and Dimos V Dimarogonas. Distributed aperiodic model predictive control for multi-agent systems. *IET Control Theory & Applications*, 9(1):10–20, 2014.

[54] Kazumune Hashimoto, Shuichi Adachi, and Dimos V Dimarogonas. Event-triggered intermittent sampling for nonlinear model predictive control. *Automatica*, 81:148–155, 2017.

[55] Changxin Liu, Huiping Li, Yang Shi, and Demin Xu. Distributed event-triggered model predictive control of coupled nonlinear systems. *SIAM Journal on Control and Optimization*, 58(2):714–734, 2020.

[56] David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[57] David Q Mayne, Erric C Kerrigan, EJ Van Wyk, and Paola Falugi. Tube-based robust nonlinear model predictive control. *International journal of robust and nonlinear control*, 21(11):1341–1353, 2011.

[58] Shuyou Yu, Christoph Maier, Hong Chen, and Frank Allgöwer. Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems. *Systems & Control Letters*, 62(2):194–200, 2013.

[59] Brett T Lopez, Jean-Jacques E Slotine, and Jonathan P How. Dynamic tube mpc for nonlinear systems. In *2019 American Control Conference (ACC)*, pages 1655–1662. IEEE, 2019.

[60] Davide Falanga, Alessio Zanchettin, Alessandro Simovic, Jeffrey Delmerico, and Davide Scaramuzza. Vision-based autonomous quadrotor landing on a moving platform. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 200–207. IEEE, 2017.

[61] Yeongho Song, Hojin Lee, Cheolhyeon Kwon, Hyo-Sang Shin, and Hyondong Oh. Distributed estimation of stochastic multiagent systems for cooperative control with a virtual network. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022.

[62] Ioannis Lymperopoulos and John Lygeros. Adaptive aircraft trajectory prediction using particle filters. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 7387, 2008.

[63] Spyros Maniatopoulos, Dimos V Dimarogonas, and Kostas J Kyriakopoulos. A decentralized event-based predictive navigation scheme for air-traffic control. In *2012 American Control Conference (ACC)*, pages 2503–2508. IEEE, 2012.

[64] Panna Felsen, Patrick Lucey, and Sujoy Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 732–747, 2018.

[65] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning. In *2019 international conference on robotics and automation (ICRA)*, pages 6658–6664. IEEE, 2019.

[66] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12126–12134, 2019.

[67] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Advances in neural information processing systems*, 33:19783–19794, 2020.

[68] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. Advdo: Realistic adversarial attacks for trajectory prediction. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022.

[69] Stefano Di Cairano, Daniele Bernardini, Alberto Bemporad, and Ilya V Kolmanovsky. Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management. *IEEE Transactions on Control Systems Technology*, 22(3):1018–1031, 2013.

[70] Russ Tedrake. *Underactuated Robotics*. 2023.

[71] Hassan K Khalil. *Nonlinear control*, volume 406. Pearson New York, 2015.

[72] A. Bressan and B. Piccoli. *Introduction to the Mathematical Theory of Control*. AIMS on applied math. American Institute of Mathematical Sciences, 2007.

[73] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC, 1994.

[74] Samir Bouabdallah. Design and control of quadrotors with application to autonomous flying. Technical report, Epfl, 2007.

[75] Fredrik Ljungberg. *Estimation of nonlinear greybox models for marine applications*, volume 1880. Linköping University Electronic Press, 2020.

[76] Charalampos P Bechlioulis and George A Rovithakis. Robust adaptive control of feedback linearizable mimo nonlinear systems with prescribed performance. *IEEE Transactions on Automatic Control*, 53(9):2090–2099, 2008.

[77] Charalampos P. Bechlioulis and George A. Rovithakis. A low-complexity global approximation-free control scheme with prescribed performance for unknown pure feedback systems. *Automatica*, 50(4):1217–1226, 2014.

[78] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[79] Lars Grüne. Nmpc without terminal constraints. *IFAC Proceedings Volumes*, 45(17):1–13, 2012.

[80] Gabriele Pannocchia, James B Rawlings, and Stephen J Wright. Conditions under which suboptimal nonlinear mpc is inherently robust. *Systems & Control Letters*, 60(9):747–755, 2011.

[81] Shuyou Yu, Marcus Reble, Hong Chen, and Frank Allgöwer. Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control. *Automatica*, 50(9):2269–2280, 2014.

[82] Dženan Lapandić, Christos K Verginis, Dimos V Dimarogonas, and Bo Wahlberg. Robust trajectory tracking for underactuated quadrotors with prescribed performance. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 3351–3358. IEEE, 2022.

[83] Shaoping Chang and Wuxi Shi. Adaptive fuzzy time-varying sliding mode control for quadrotor uav attitude system with prescribed performance. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 4389–4394. IEEE, 2017.

[84] Gang Xu, Yuanqing Xia, Di-Hua Zhai, and Dailiang Ma. Adaptive prescribed performance terminal sliding mode attitude control for quadrotor under input saturation. *IET Control Theory & Applications*, 14(17):2473–2480, 2020.

[85] Changchun Hua, Jiannan Chen, and Xinping Guan. Adaptive prescribed performance control of quavs with unknown time-varying payload and wind gust disturbance. *Journal of the Franklin Institute*, 355(14):6323–6338, 2018.

[86] Tao Jiang, Jiangshuai Huang, and Bin Li. Composite adaptive finite-time control for quadrotors via prescribed performance. *Journal of the Franklin Institute*, 357(10):5878–5901, 2020.

[87] Kazuya Sasaki and Zi-Jiang Yang. Disturbance observer-based control of uavs with prescribed performance. *International Journal of Systems Science*, 51(5):939–957, 2020.

[88] Zhipeng Shen, Feng Li, Xiaoming Cao, and Chen Guo. Prescribed performance dynamic surface control for trajectory tracking of quadrotor uav with uncertainties and input constraints. *International Journal of Control*, 94(11):2945–2955, 2021.

[89] Christos K Verginis, Charalampos P Bechlioulis, Argiris G Soldatos, and Dimitris Tsipianitis. Robust trajectory tracking control for uncertain 3-dof helicopters with prescribed performance. *IEEE/ASME Transactions on Mechatronics*, 2022.

[90] Rita Cunha, David Cabecinhas, and Carlos Silvestre. Nonlinear trajectory tracking control of a quadrotor vehicle. In *2009 European Control Conference (ECC)*, pages 2763–2768. IEEE, 2009.

[91] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[92] Bruce Donald, Patrick Xavier, John Canny, and John Reif. Kinodynamic motion planning. *Journal of the ACM (JACM)*, 40(5):1048–1066, 1993.

[93] Ioan A Sucan and Lydia E Kavraki. Kinodynamic motion planning by interior-exterior cell exploration. *Algorithmic Foundation of Robotics VIII*, 57:449–464, 2009.

[94] Sertac Karaman and Emilio Frazzoli. Optimal kinodynamic motion planning using incremental sampling-based methods. In *49th IEEE conference on decision and control (CDC)*, pages 7681–7687. IEEE, 2010.

[95] Carl De Boor. *A practical guide to splines*. Applied mathematical sciences ; 27. Springer, New York, rev. ed edition, 2001.

[96] Boyu Zhou, Fei Gao, Luqi Wang, Chuhao Liu, and Shaojie Shen. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters*, 4(4):3529–3536, 2019.

[97] Jesus Tordesillas and Jonathan P How. Mader: Trajectory planner in multiagent and dynamic environments. *IEEE Transactions on Robotics*, 38(1):463–476, 2021.

[98] Ruben Van Parys and Goele Pipeleers. Distributed mpc for multi-vehicle systems moving in formation. *Robotics and Autonomous Systems*, 97:144–152, 2017.

[99] Kaihuai Qin. General matrix representations for b-splines. In *Proceedings Pacific Graphics' 98. Sixth Pacific Conference on Computer Graphics and Applications (Cat. No. 98EX208)*, pages 37–43. IEEE, 1998.

[100] Tamás Keviczky and Karl Henrik Johansson. A study on distributed model predictive consensus. *IFAC Proceedings Volumes*, 41(2):1516–1521, 2008.

[101] Daniel Limón, Ignacio Alvarado, Teodoro Alamo, and Eduardo F Camacho. Mpc for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, 2008.

[102] William B Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, 2007.

[103] Arthur Richards and Jonathan P How. Robust distributed model predictive control. *International Journal of control*, 80(9):1517–1531, 2007.

[104] Daniel Limón, Teodoro Alamo, Francisco Salas, and Eduardo F Camacho. On the stability of constrained mpc without terminal constraint. *IEEE transactions on automatic control*, 51(5):832–836, 2006.

[105] Johannes Köhler, Matthias A Müller, and Frank Allgöwer. A novel constraint tightening approach for nonlinear robust model predictive control. In *2018 Annual American Control Conference (ACC)*, pages 728–734. IEEE, 2018.

[106] David Mayne. An apologia for stabilising terminal conditions in model predictive control. *International Journal of Control*, 86(11):2090–2095, 2013.

[107] Noel Csomay-Shanklin, Andrew J Taylor, Ugo Rosolia, and Aaron D Ames. Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control lyapunov functions. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 3732–3739. IEEE, 2022.

[108] Ugo Rosolia, Andrew Singletary, and Aaron D Ames. Unified multirate control: From low-level actuation to high-level planning. *IEEE Transactions on Automatic Control*, 67(12):6627–6640, 2022.

[109] Olov Andersson, Patrick Doherty, Mårten Lager, Jens-Olof Lindh, Linnea Persson, Elin A Topp, Jesper Tordenlid, and Bo Wahlberg. Wara-ps: a research arena for public safety demonstrations and autonomous collaborative rescue robotics experimentation. *Autonomous Intelligent Systems*, 1(1):1–31, 2021.

[110] Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Nonlinear reference tracking with model predictive control: An intuitive approach. In *2018 European Control Conference (ECC)*, pages 1355–1360. IEEE, 2018.

[111] David Angeli. A lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

[112] Duc N Tran, Björn S Rüffer, and Christopher M Kellett. Incremental stability properties for discrete-time systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 477–482. IEEE, 2016.

[113] Andrea Boccia, Lars Grüne, and Karl Worthmann. Stability and feasibility of state constrained mpc without stabilizing terminal constraints. *Systems & control letters*, 72:14–21, 2014.

[114] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.