Advances in Model Predictive Flocking: Analysis and Design

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation (Monografie)

von
Philipp Hastedt

aus
Hamburg

2025

Vorsitzender des Prüfungsausschusses:

Prof. Dr.-Ing. Gerhard Bauch

Gutachter:

Prof. Dr. Herbert Werner

Prof.dr. Rudy R. Negenborn

Datum der mündlichen Prüfung:

22.05.2025

To my family.

Summary

This thesis addresses several aspects related to the analysis and design in the field of model predictive flocking (MPF), a powerful technique for the distributed and self-organized control of multi-agent systems.

Regarding the design of MPF algorithms, a novel framework is proposed, which results in an improved performance, particularly in complex scenarios involving group objectives and obstacle avoidance. The main advantage of this framework is the capability to independently tune attractive and repulsive forces between members of the swarm, enhancing the adaptability to different scenarios. This control strategy is referred to as asymmetric model predictive flocking (AMPF), and its performance is validated in simulation, demonstrating a superior performance compared to existing approaches. In contrast to the objective of classical flocking control, which is to converge towards a formation where all agents achieve a desired displacement to nearby members of the swarm, a novel concept aims to achieve elliptic formations, allowing agents to have distinct desired displacements along different directions. In this context, a novel framework for elliptic flocking (EF) is proposed and validated in simulation, extending the applicability of the approaches described in the existing literature. Moreover, a framework for elliptic model predictive flocking is proposed, which combines the asymmetric attractive and repulsive inter-agent forces with elliptical desired formations.

With regard to the stability analysis of model predictive flocking, the general challenges affecting the analysis are identified. While some of these challenges originate from the flocking objectives themselves, others result from the formulations of the involved optimization problems. Among others, these challenges include unknown setpoints, time-varying communication topologies, and information mismatch. It is then investigated how these challenges are addressed in the MPF literature. In this context, in the prevalent approaches based on compatibility constraints and geometric properties of the optimal state sequence, so-called N-paths, several shortcomings in the existing stability results are pointed out and their impact on the analysis is discussed. These errors can be traced to assumptions that are not generally valid, incorrect lemmas, and challenges in the MPF analysis that were not taken into account. In light of these findings, this work proposes a framework for analyzing centralized model predictive flocking that relies on the stability of sets of equilibria and avoids the shortcomings identified in the MPF literature. Furthermore, it is demonstrated that this analysis can also be extended to the proposed asymmetric and elliptic predictive frameworks. In the context of elliptic flocking, it is shown how analysis approaches in the related literature can be extended to the elliptic setting by formulating the EF problem as a generalization of well-studied flocking frameworks.

Contents

Acronyms and Abbreviations **xi**

1 Introduction **1**
1.1 Classification of Formation Control . . . 1
1.1.1 Flocking . . . 2
1.1.2 Model Predictive Flocking . . . 3
1.2 Contributions . . . 4
1.3 Thesis Outline . . . 5

2 Theoretical Background **7**
2.1 Notation . . . 7
2.2 Mathematical Preliminaries . . . 7
2.2.1 Sets in $\mathbb{R}^n$ . . . 8
2.2.2 Dynamic Multi-Agent Systems . . . 8
2.2.3 Graph Theory . . . 10
2.2.4 Stability . . . 12
2.3 Flocking Control . . . 13
2.3.1 α-Lattices and σ-Maps . . . 14
2.3.2 Flocking Protocols . . . 15
2.4 Model Predictive Control . . . 19
2.4.1 Model Predictive Control Architectures . . . 19
2.4.2 Model Predictive Control Stability Analysis . . . 24

3 Model Predictive Flocking and the Challenges in the Stability Analysis **27**
3.1 Model Predictive Flocking . . . 27
3.1.1 Centralized Model Predictive Flocking . . . 28
3.1.2 Distributed Model Predictive Flocking . . . 28
3.2 Challenges in the Stability Analysis of Model Predictive Flocking . . . 29
3.2.1 Non-Convex and not Continuously Differentiable Cost Functions . . . 29
3.2.2 Unknown Setpoint . . . 31
3.2.3 Time-Varying Communication Topology . . . 31
3.2.4 Prediction Mismatch in Distributed Optimization . . . 33
3.3 On the Analysis of Model Predictive Flocking Using N-Paths . . . 35
3.3.1 N-Paths . . . 35
3.3.2 Model Predictive Flocking Problem Statement and N-Path-Based Analysis . . . 37
3.3.3 Critical Discussion . . . 40

3.4 On the Analysis of Model Predictive Flocking Using Compatibility Constraints 43
3.4.1 Compatibility Constraints . . . 44
3.4.2 On the Analysis of Distributed Model Predictive Flocking in Hu et al. (2017) . . . 45
3.4.3 On the Analysis of Distributed Model Predictive Flocking in Hu et al. (2018) . . . 50
3.5 A Stability Analysis of Centralized Predictive Flocking Using Set Stability 56
3.5.1 Stability of Sets of Equilibria . . . 57
3.5.2 A Stability Analysis for Centralized Model Predictive Flocking with Terminal Equality Constraints . . . 59

4 Model Predictive Flocking with Asymmetric Interaction Forces 63
4.1 Asymmetric Interaction Forces . . . 64
4.2 Problem Statement . . . 65
4.2.1 Neighbor Trajectory Estimation . . . 66
4.2.2 α-Asymmetric Distributed Model Predictive Flocking . . . 66
4.2.3 α-γ-Asymmetric Distributed Model Predictive Flocking . . . 67
4.2.4 α-β-γ-Asymmetric Distributed Model Predictive Flocking . . . 68
4.2.5 Numerical Optimization . . . 69
4.3 Stability Analysis for Asymmetric Centralized Model Predictive Flocking . 70
4.4 Parameter Studies . . . 71
4.4.1 Trajectory Estimation and Prediction Horizon . . . 71
4.4.2 SQP Iterations . . . 77
4.5 Performance Comparison . . . 79
4.5.1 Setup . . . 79
4.5.2 Performance Measures . . . 79
4.5.3 Results . . . 79

5 Flocking with Ellipsoidal Level Sets 85
5.1 Mathematical Descriptions of Ellipses and Ellipsoids . . . 86
5.2 Elliptic Flocking Protocols . . . 89
5.2.1 Elliptic Proximity Graphs and ε-Lattices . . . 89
5.2.2 α-Elliptic Flocking . . . 90
5.2.3 α-γ-Elliptic Flocking . . . 93
5.2.4 α-β-γ-Elliptic Flocking . . . 93
5.3 Stability Analysis of Elliptic Flocking . . . 94
5.3.1 Analysis of Free Elliptic Flocking . . . 94
5.3.2 Analysis of Elliptic Flocking with Group Objectives . . . 96
5.3.3 Analysis of Elliptic Flocking with Obstacle Avoidance . . . 97
5.4 Simulation Results . . . 98
5.4.1 Performance Measures . . . 99
5.4.2 Scenario 1: 2D Performance Comparison . . . 99
5.4.3 Scenario 2: 3D Simulation Example . . . 103
5.4.4 Scenario 3: Time-Varying Desired Formations and Obstacle Avoidance 103

6 Elliptic Model Predictive Flocking **107**
6.1 Elliptic Model Predictive Flocking Framework . . . 107
6.1.1 Optimization and Neighbor Trajectory Estimation . . . 107
6.1.2 α-Elliptic Distributed Model Predictive Flocking . . . 108
6.1.3 α-γ-Elliptic Distributed Model Predictive Flocking . . . 109
6.1.4 α-β-γ-Elliptic Distributed Model Predictive Flocking . . . 109
6.2 Stability Analysis of Elliptic Centralized Model Predictive Flocking . . . 110
6.3 Numerical Example . . . 112
6.3.1 Setup . . . 112
6.3.2 Performance Measures . . . 113
6.3.3 Results . . . 113

7 Conclusion and Future Work **115**
7.1 Summary . . . 115
7.2 Outlook . . . 116

Bibliography **117**

Appendix **123**

A Simulation and Tuning Parameters **125**
A.1 Parameters for Chapter 4 . . . 125
A.2 Parameters for Chapter 5 . . . 126
A.3 Parameters for Chapter 6 . . . 127

Acronyms and Abbreviations

ACMPF	Asymmetric centralized model predictive flocking
ADMPF	Asymmetric distributed model predictive flocking
AMPF	Asymmetric model predictive flocking
CMPC	Centralized model predictive control
CMPF	Centralized model predictive flocking
DMPC	Distributed model predictive control
DMPF	Distributed model predictive flocking
ECMPF	Elliptic centralized model predictive flocking
EDMPF	Elliptic distributed model predictive flocking
EF	Elliptic flocking
EMPF	Elliptic model predictive flocking
LTI	Linear time-invariant
MAS	Multi-agent system
MPC	Model predictive control
MPF	Model predictive flocking
SQP	Sequential quadratic programming

1 Introduction

From environmental monitoring and public transportation to space exploration, multi-agent systems (MASs) have gained growing attention in recent years due to their potential to solve complex problems in various domains (Athanasiadis and Mitkas 2004; Burmeister et al. 1997; Truszkowski et al. 2006). Multi-agent systems consist of multiple interacting agents, each capable of performing actions autonomously while cooperating with others to achieve a shared goal. The agents in such systems can represent robots, vehicles, sensors, or even software entities, each equipped with local decision-making capabilities. One of the primary reasons for the importance of MASs as a research topic is their ability to handle tasks that exceed the capacity of a single agent or centralized systems. For example, in search and rescue operations following a natural disaster, a fleet of agents can be deployed to cover a large area, communicate with one another, and collaborate to perform their tasks more efficiently than an individual agent could on its own (Drew 2021).

1.1 Classification of Formation Control

Due to the vast amount of literature on multi-agent systems, it is more meaningful to discuss the general characteristics of MAS control strategies here rather than providing an exhaustive review of the existing literature on MASs. As a starting point to gain an understanding of MASs, consider Cai et al. (2022), Francis and Maggiore (2016), Ren and Cao (2011), and the references therein.

Control strategies for MASs can be broadly classified into two categories: centralized and decentralized. In a centralized control approach, a single entity, often referred to as a central controller, gathers information from all agents and computes their actions in a unified manner. The advantage of this approach is that the central controller has global knowledge of the system, which can lead to optimal decision-making. However, this approach is subject to a number of limitations, including scalability challenges caused by the increasing complexity for large numbers of agents, communication bandwidth, and a single point of failure (Cao et al. 2013). In contrast, a decentralized control approach involves agents making decisions based on local information and interactions with neighboring agents without a global perspective. This method offers several advantages, including enhanced scalability, reduced controller complexity, and higher robustness (Cao et al. 2013). An illustrative example of decentralized control is the coordination of autonomous vehicle traffic, where each vehicle must make decisions based on its local environment without relying on centralized instructions.

Following the classification introduced in Oh et al. (2015), control techniques for multi-agent systems can be characterized as position-, displacement-, and distance-based formation control, depending on how the desired formation is achieved. In position-based control, agents control their own position with respect to a global coordinate system to achieve the desired formation. In displacement-based control, the desired formation is specified by the desired displacements between agents with respect to a global coordinate frame whose orientation is known by the agents. The desired formation is then achieved by the agents controlling the displacement to their neighbors. In distance-based control, no knowledge of the global coordinate system is required by the agents. Instead, agents control their distance to neighboring agents to achieve the desired formation.

According to how the formation behaves, control approaches can also be categorized as leader-follower approach, behavioral approach, or virtual structure approach (Oh et al. 2015). In leader-follower formation control, at least one of the agents is considered the leader tracking a reference trajectory. The other agents then follow this leader in a position-, displacement-, or distance-based manner. In behavioral formation control, the movement of the MAS is governed by behavioral rules. In this approach, the desired formation is usually not specified explicitly but determined by the imposed rules. In the virtual structure approach, the overall formation of the agents is considered as a single object, a so-called virtual structure. The behavior of the individual agents is then determined from the desired behavior of the virtual structure.

1.1.1 Flocking

One important aspect of MASs is the ability to exhibit self-organized swarm behavior. The need for self-organized swarm behavior arises in scenarios where centralized control is either infeasible or undesirable, for example in large-scale scenarios involving the coordination of possibly thousands of agents. A specific type of self-organized swarm behavior that has received significant attention is flocking control. Inspired by the coordinated movement of biological swarms, such as bird flocks, fish schools, and ant colonies, flocking control refers to the process by which agents are governed by a simple set of rules derived from these biological swarms. In terms of the previously introduced classifications, flocking is considered a distance-based, behavioral control technique. One famous set of rules for flocking was stated in Reynolds (1987), leading to the first computer animation of flocking. The set of rules used are cohesion, separation, and alignment. In simple terms, cohesion refers to agents staying close to other members of the swarm, separation describes the avoidance of collisions with other agents, and alignment refers to the objective of matching the velocity of nearby agents. Based on these rules, many different mathematical frameworks for the control of MASs have been proposed. A good starting point for further reading on this topic are the studies by Barve and Nene (2013), Beaver and Malikopoulos (2021), and the references therein.

One well-known framework for flocking control is presented in Olfati-Saber (2006). Based on the rules of Reynolds, carefully shaped potential functions are used to achieve a formation where agents have a desired inter-agent distance to all of their neighboring

agents. The resulting formation is a hexagonal lattice structure, also referred to as an α-lattice. To increase scalability, agents furthermore only interact with their direct neighbors, resulting in a distributed control algorithm that is independent of the number of members of the swarm. Furthermore, obstacle avoidance and group objectives are considered in the flocking framework by considering them as virtual agents the members of the swarm interact with. Due to its flexibility, the general framework introduced in Olfati-Saber (2006) is also used as a starting point for many other flocking frameworks, for example including source seeking capabilities and the segmentation of heterogeneous swarms (Attallah et al. 2020; Kumar et al. 2010). For more applications of flocking control, see Barve and Nene (2013), Beaver and Malikopoulos (2021), Francis and Maggiore (2016), and the references therein. One limitation of the flocking frameworks mentioned above is that they are only applicable to scenarios where the desired formation is an α-lattice, resulting in a configuration where agents have the same distance to all their neighbors. This issue was addressed in recent publications, introducing a framework that allows for distinct inter-agent distances in different directions (Wang et al. 2022; Wang et al. 2023). One of the main applications of this technique is in scenarios where distinct safety distances are required in different directions, for example in lane driving scenarios (Wang et al. 2023). As the resulting formation of the MAS can be characterized as an elliptic lattice, this class of algorithms is referred to as elliptic flocking (EF) in this thesis.

1.1.2 Model Predictive Flocking

Motivated by the abilities of biological agents to predict the behavior of neighboring members of a swarm, one particular branch of flocking research focuses on incorporating such predictive mechanisms into flocking algorithms (Zhang et al. 2008). Commonly, such features are realized via model predictive control (MPC), and the resulting algorithms are referred to as model predictive flocking (MPF) algorithms. In MPF, at each sampling instance, agents determine an optimal control input with respect to a chosen performance measure by predicting the future behavior of themselves and neighboring agents. Additionally, MPC furthermore allows for the explicit consideration of performance and safety constraints, making MPF a suitable control technique for practical swarm control applications (Beaver and Malikopoulos 2021).

The first MPF algorithms were proposed in Zhan and Li (2011a,b, 2013). In these papers, the desired flocking behavior is achieved by designing MPC problems with cost functions capturing the flocking rules of cohesion, separation, alignment (Reynolds 1987). Over the last years, several results have been published in the field of MPF. In Yuan et al. (2017), the simulation results of Zhan and Li (2011a) were validated experimentally using swarms of multicopters. Furthermore, input constraints, the tracking of group objectives, and obstacle avoidance were integrated in the MPF framework (Hu et al. 2018, 2017; Huang et al. 2019; Zhang et al. 2015; Zhou and Li 2017).

While all of the aforementioned literature on MPF is based on the objective of agents achieving a desired inter-agent distance similar to the approach described in Olfati-Saber (2006), there also exist publications in the field of MPF that address the flocking problem

in a different way. For example, in Zhang et al. (2016), the authors take an approach centered around velocity alignment, and in Beaver and Malikopoulos (2020), the desired behavior is achieved via a set of collision avoidance and aggregation constraints. A good overview of different approaches to optimal and predictive flocking is provided in Beaver and Malikopoulos (2021). With the most commonly used approach in the related literature being based on the objective of forming an α-lattice, this is also the approach most of the discussions in this thesis focus on.

1.2 Contributions

As the title of this thesis suggests, the contributions are mainly related to the analysis and design of model predictive flocking (MPF) algorithms. In terms of design, there is still room for performance improvements, especially in complex scenarios involving group objectives and obstacle avoidance. Furthermore, the promising concept of elliptic flocking (EF) has so far not been implemented in predictive frameworks, and the performance of non-predictive elliptic flocking is also not pushed to its limit yet. The contributions of this thesis with respect to the design can therefore be summarized as follows:

1. A novel MPF framework with the capability of tuning attractive and repulsive inter-agent forces independently, referred to as asymmetric model predictive flocking (AMPF), is introduced. The proposed framework also includes reference tracking and obstacle avoidance and demonstrates superior performance compared to existing frameworks, particularly in obstacle avoidance scenarios.

2. In the field of non-predictive flocking, a framework for EF is proposed as a generalization of the flocking framework presented in Olfati-Saber (2006). The proposed algorithm also allows for inclusion of elliptic obstacle avoidance, and in comparison to existing EF frameworks, achieves superior performance in simulation.

3. A framework for elliptic model predictive flocking (EMPF) is proposed, combining the advantages of EF with the presented asymmetric model predictive flocking framework, including reference tracking and elliptic obstacle avoidance.

When skimming through the literature in the field of model predictive flocking, at first glance, it seems that the stability analysis of MPF has also been satisfactorily addressed. In the papers that provide an analysis, mainly one of two approaches is used. The first approach exploits geometric properties of the optimal state sequence, so-called N-paths (Zhan and Li 2013; Zhang et al. 2015; Zhou and Li 2017). The second approach imposes compatibility constraints on the optimization problem (Hu et al. 2018, 2017). However, due to the nature of the MPF problem, the analysis is challenging, especially in the context of distributed MPF implementations. Upon closer examination of the analysis results presented in the related literature, it becomes evident that these challenges are not addressed properly, resulting in some of the existing theoretical results being invalid. This thesis addresses the analysis of model predictive flocking in detail. The contributions related to the analysis of model predictive flocking can be summarized as follows:

1. The challenges arising in the analysis of MPF algorithms are identified. These includes challenges related to the predictive flocking problem in general as well as challenges related to commonly used implementations.

2. Analysis results in the MPF literature are discussed, pointing out several errors in the presented lines of reasoning. This includes the approaches based on N-paths in Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017), as well as the analysis based on compatibility constraints in Hu et al. (2018, 2017). These errors can mainly be traced back to not properly addressing the challenges arising from the MPF problem formulation, assumptions that do not hold in general, and incorrect lemmas.

3. A result for the analysis of centralized model predictive flocking based on the stability of sets of equilibria is proposed (Theorem 3.12), which can also be extended to AMPF (Theorem 4.1) and EMPF (Theorem 6.2).

4. With respect to non-predictive flocking, the analysis in Olfati-Saber (2006) is extended to the proposed EF algorithms (Theorem 5.2, 5.3, 5.4).

1.3 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 contains the theoretical background required for this thesis, including mathematical preliminaries as well as introduction to the general frameworks of MPC and flocking control.

In Chapter 3, the general concept of MPF is reviewed followed by a detailed investigation of the stability analysis of MPF algorithms. This includes a study of the general challenges that arise from the MPF problem formulation, as well as a critical discussion of the stability analyses presented in the related literature. Furthermore, a framework for the general analysis of CMPF is presented.

In Chapter 4, the AMPF framework allowing for an individual tuning of attractive and repulsive inter-agent forces is presented. This includes the problem formulations for free flocking, flocking with group objective, and flocking with obstacle avoidance. Furthermore, a stability analysis based on the framework derived in Chapter 3 is presented. Finally, the performance of the AMPF framework is validated in simulation, including a comparison to predictive and non-predictive flocking algorithms in the literature.

In Chapter 5, a framework for EF is presented as a generalization of the flocking framework presented in Olfati-Saber (2006), including the pursuit of group objectives and elliptic obstacle avoidance. It is furthermore demonstrated how the analysis presented in Olfati-Saber (2006) can be extended to the elliptic flocking algorithms. The performance of the proposed scheme is validated in simulation and compared to an existing EF framework.

In Chapter 6, the results of the previous chapters are combined by formulating a framework for EMPF with asymmetric interaction forces, the pursuit of group objectives, and elliptic obstacle avoidance. Furthermore, the analysis presented in Chapter 3 is extended to Chapter 6. The performance of the proposed scheme is validated in simulation.

Chapter 7 summarizes the results presented in this thesis and outlines possible future directions of research in the field of MPF.

The content dependencies of the main chapters of this thesis are visualized in Fig. 1.1.

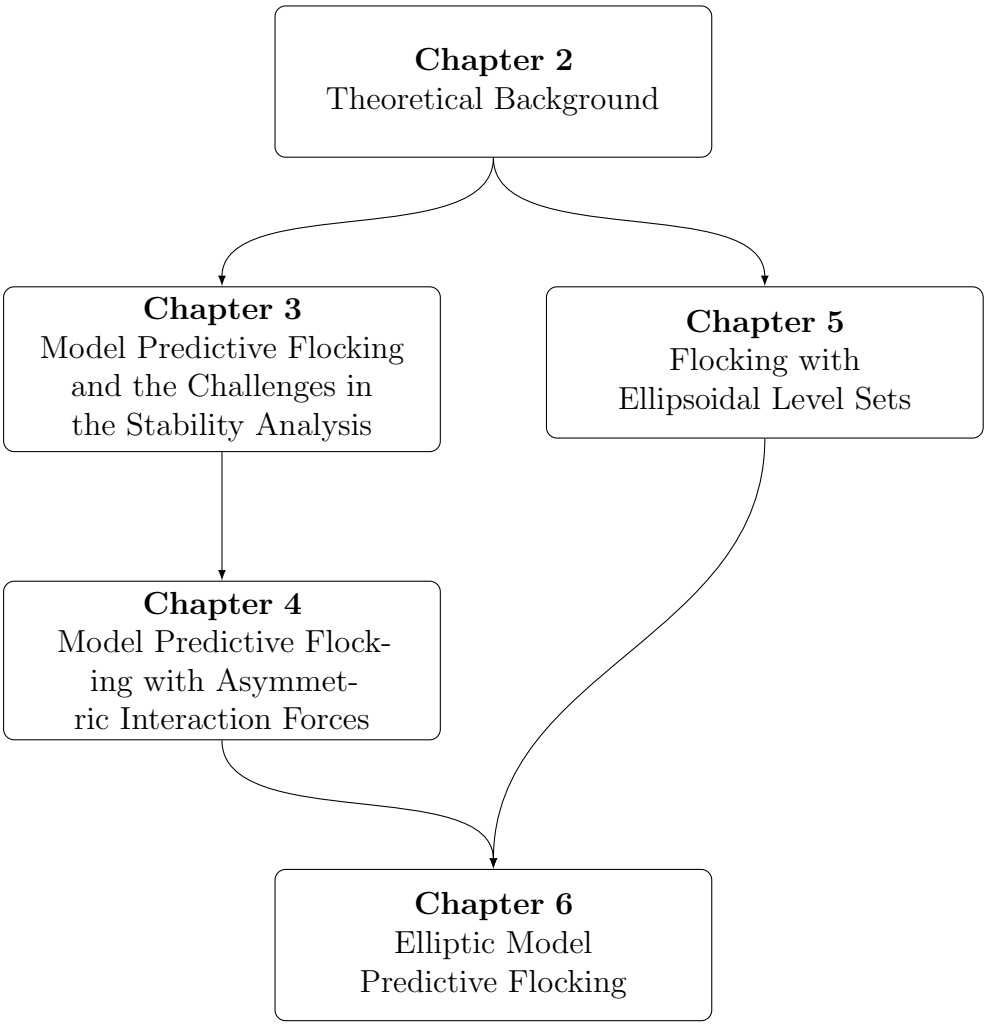


Figure 1.1: Dependencies of the main chapters.

2 Theoretical Background

This chapter introduces the notation, mathematical background, and frameworks used throughout this thesis. This includes the required mathematical preliminaries and the fundamental frameworks for flocking and model predictive control.

2.1 Notation

Let $\mathbb{R}$ be the set of real numbers, with $\mathbb{R}_{\geq 0}$ denoting the non-negative real numbers. $\mathbb{N}$ denotes the set of natural numbers including zero. The index set $\mathbb{I}$, ranging from i_1 to i_2, is defined as $\mathbb{I}_{i_1:i_2} = \{i \in \mathbb{N} : i_1 \leq i \leq i_2\}$. The symbols 0 and I represent zero and identity matrices with compatible dimensions, unless specified explicitly. Zero and one vectors of dimension $n \times 1$ are denoted by 0_n and 1_n, respectively. For a matrix X, $X^\top$ denotes its transpose and X^{-1} its inverse. The entry of a matrix in the i^{th} row and j^{th} column is denoted by $[X]_{ij}$. If X is real and symmetric, $X \succ (\succeq)\, 0$ denotes that X is positive definite (semi-definite) and $X \prec (\preceq)\, 0$ represents that X is negative definite (semi-definite). For $x_i \in \mathbb{R}^{n_i}$, $i \in \mathbb{I}_{1:N}$, the stacked vector of x_i's is defined as

$$\operatorname{col}(x_1, \ldots, x_N) = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}.$$

For $x_i \in \mathbb{R}$, $i \in \mathbb{I}_{1:N}$, the diagonal matrix with the x_i on the diagonal is denoted by $\operatorname{diag}(x_1, \ldots, x_N)$. Similarly, for matrices $X_i \in \mathbb{R}^{n_i \times m_i}$, $i \in \mathbb{I}_{1:N}$, the block-diagonal matrix with the X_i as blocks on the diagonal is denoted by $\operatorname{blkdiag}(X_1, \ldots, X_N)$. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\|$, and $\|x\|_M = \sqrt{x^\top M x}$ denotes the M-weighted norm with M being real and positive definite. The Kronecker product is denoted by $\otimes$, and $\oplus$ is the Minkowski sum. In Cartesian coordinate systems, the coordinates are denoted by ξ_i, with $i \in \mathbb{I}_{1:2}$ in two dimensions (2D) and $i \in \mathbb{I}_{1:3}$ in three dimensions (3D).

2.2 Mathematical Preliminaries

This section presents the background and notation on set theory, dynamic systems, graph theory, and notions of stability that are fundamental to this thesis.

2.2.1 Sets in $\mathbb{R}^n$

For $x \in \mathbb{R}^n$, let $\mathcal{B}_r^{\mathrm{o}}(x) = \{y \in \mathbb{R}^n : \|y - x\| < r\}$ define an open ball of radius r around x. Analogously, $\mathcal{B}_r^{\mathrm{c}}(x) = \{y \in \mathbb{R}^n : \|y - x\| \leq r\}$ defines the closed ball. A set $X \subset \mathbb{R}^n$ is said to be open if for every $x \in X$, there exists a ball of radius $r > 0$ such that $\mathcal{B}_r^{\mathrm{c}}(x)$ is in X. Moreover, a set $X \subset \mathbb{R}^n$ is closed if its complement $Y = \{x \in \mathbb{R}^n : x \notin X\}$ is open. Furthermore, $X \subset \mathbb{R}^n$ is said to be bounded if there exists $\rho < \infty$ such that $\|x\| < \rho$ for all $x \in X$. If a set is both, closed and bounded, it is said to be compact. A set $X \subset \mathbb{R}^n$ is referred to as polyhedral if it can be represented as an intersection of m half-spaces, $X = \{x \in \mathbb{R}^n : Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Given a set $X \subset \mathbb{R}^n$, the distance between a point $y \in \mathbb{R}^n$ and X is defined as $|y|_X = \inf_{x \in X} \|x - y\|$. The open and closed sets containing all points within a distance $r > 0$ from $X \subset \mathbb{R}^n$ are given by $\mathcal{B}_r^{\mathrm{o}}(X) = \{y \in \mathbb{R}^n : |y|_X < r\}$ and $\mathcal{B}_r^{\mathrm{c}}(X) = \{y \in \mathbb{R}^n : |y|_X \leq r\}$, respectively.

2.2.2 Dynamic Multi-Agent Systems

Consider a multi-agent system (MAS) consisting of N dynamically decoupled agents that are moving in an n_{d}-dimensional space, where n_{d} is typically 2 or 3. To describe the motion of the MAS, the dynamics of a single agent are introduced first, followed by the collective swarm dynamics.

Single Agent Dynamics

The constrained dynamics $f_i^{\mathrm{c}} : \mathbb{X}_i \times \mathbb{U}_i \to \mathbb{X}_i$ of a single agent with index $i \in \mathbb{I}_{1:N}$ are described by the differential equation

$$\dot{x}_i(t) = f_i^{\mathrm{c}}(x_i(t), u_i(t)), \tag{2.1}$$

with state $x_i \in \mathbb{R}^{n_{x,i}}$ and input $u_i \in \mathbb{R}^{n_{u,i}}$. The sets $\mathbb{X}_i$ and $\mathbb{U}_i$ denote the admissible states and inputs given by

$$\mathbb{X}_i = \{x_i \in \mathbb{R}^{n_{x,i}} : h_{i,x}^{\mathrm{eq}}(x_i) = 0, h_{i,x}^{\mathrm{ieq}}(x_i) \leq 0\}, \tag{2.2}$$

$$\mathbb{U}_i = \{u_i \in \mathbb{R}^{n_{u,i}} : h_{i,u}^{\mathrm{eq}}(u_i) = 0, h_{i,u}^{\mathrm{ieq}}(u_i) \leq 0\}. \tag{2.3}$$

Here, $h_{i,x}^{\mathrm{eq}} : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{c_{i,x}^{\mathrm{eq}}}$, $h_{i,x}^{\mathrm{ieq}} : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{c_{i,x}^{\mathrm{ieq}}}$ $h_{i,u}^{\mathrm{eq}} : \mathbb{R}^{n_{u,i}} \to \mathbb{R}^{c_{i,u}^{\mathrm{eq}}}$, and $h_{i,u}^{\mathrm{ieq}} : \mathbb{R}^{n_{u,i}} \to \mathbb{R}^{c_{i,u}^{\mathrm{ieq}}}$ are vector-valued equality and inequality constraint functions for the states and inputs, respectively. The numbers of the corresponding constraints are given by $c_{i,x}^{\mathrm{eq}}$, $c_{i,x}^{\mathrm{ieq}}$, $c_{i,u}^{\mathrm{eq}}$, and $c_{i,u}^{\mathrm{ieq}}$. In the context of discrete-time systems, the dynamics of agent i are written as

$$x_i(k+1) = f_i^{\mathrm{d}}(x_i(k), u_i(k)), \tag{2.4}$$

$f_i^{\mathrm{d}} : \mathbb{X}_i \times \mathbb{U}_i \to \mathbb{X}_i$, where $k \in \mathbb{N}$ is a non-negative integer denoting the sample number. The time and sample number are connected by the relation $t = \tau k$, where τ is the sampling time. For discrete quantities at time k, for example $x_i(k)$, the abbreviation $x_{i,k}$ is used. If all agents are governed by the same dynamics, the MAS is referred to as homogeneous. Otherwise, it is called heterogeneous.

One special class of dynamic systems are linear time-invariant (LTI) systems, for which the dynamics can be written as

$$\dot{x}_i(t) = A_i^{\mathrm{c}} x_i(t) + B_i^{\mathrm{c}} u_i(t), \tag{2.5}$$

where $A_i^{\mathrm{c}} \in \mathbb{R}^{n_{x,i} \times n_{x,i}}$ is the system matrix and $B_i^{\mathrm{c}} \in \mathbb{R}^{n_{x,i} \times n_{u,i}}$ is the input matrix. The corresponding discrete-time dynamics are given by

$$x_i(k+1) = A_i^{\mathrm{d}} x_i(k) + B_i^{\mathrm{d}} u_i(k), \tag{2.6}$$

where A_i^{d} and B_i^{d} are the discrete-time equivalents of the model matrices in (2.5). In the remainder of this thesis, the superscripts c and d indicating continuous- and discrete-time functions and model matrices are dropped to increase readability.

To describe the agents' positions and velocities in the n_{d}-dimensional space, let $q_i, p_i \in \mathbb{R}^{n_{\mathrm{d}}}$ describe the translational position and velocity, respectively.

Two types of agent dynamics commonly used in this thesis are those of single- and double-integrator dynamics with $x = q$ and $x = [q\ p]^\top$, respectively. In continuous time, the n_{d}-dimensional dynamics can be stated as

$$\dot{q}_i(t) = u_i(t) \tag{2.7}$$

for single-integrator dynamics and

$$\begin{aligned} \dot{q}_i(t) &= p_i(t), \\ \dot{p}_i(t) &= u_i(t), \end{aligned} \tag{2.8}$$

for double-integrator dynamics. The corresponding state-space representations are given by

$$\dot{x}_i(t) = 0 x_i(t) + I_{n_{\mathrm{d}}} u_i(t)$$

and

$$\dot{x}_i(t) = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes I_{n_{\mathrm{d}}} \right) x_i(t) + \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes I_{n_{\mathrm{d}}} \right) u_i(t),$$

respectively. Using the forward-Euler discretization with sampling time τ, the discrete-time single- and double-integrator dynamics are given by

$$q_i(k+1) = q_i(k) + \tau u_i(k) \tag{2.9}$$

and

$$\begin{aligned} q_i(k+1) &= q_i(k) + \tau p_i(k), \\ p_i(k+1) &= p_i(k) + \tau u_i(k), \end{aligned} \tag{2.10}$$

with the corresponding state-space models

$$x_i(k+1) = I_{n_{\mathrm{d}}} x_i(k) + \tau I_{n_{\mathrm{d}}} u_i(k)$$

and

$$x_i(k+1) = \left(\begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \otimes I_{n_{\mathrm{d}}} \right) x_i(k) + \left(\begin{bmatrix} 0 \\ \tau \end{bmatrix} \otimes I_{n_{\mathrm{d}}} \right) u_i(k),$$

respectively.

Collective Dynamics

To describe the motion of the overall MAS, let $x = \text{col}(x_1, \ldots, x_N) \in \mathbb{R}^{n_x}$ and $u = \text{col}(u_1, \ldots, u_N) \in \mathbb{R}^{n_u}$ denote the collective state and input with dimensions $n_x = \sum_{i=1}^{N} n_{x,i}$ and $n_u = \sum_{i=1}^{N} n_{u,i}$, respectively. For homogeneous MASs, the dimensions become $n_x = n_{x,i}N$ and $n_u = n_{u,i}N$. The collective continuous-time and discrete-time dynamics for general nonlinear systems are then given by $f : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ with

$$f(x(t), u(t)) = \begin{bmatrix} f_1(x_1(t), u_1(t)) \\ \vdots \\ f_N(x_N(t), u_N(t)) \end{bmatrix}, f(x(k), u(k)) = \begin{bmatrix} f_1(x_1(k), u_1(k)) \\ \vdots \\ f_N(x_N(k), u_N(k)) \end{bmatrix}. \tag{2.11}$$

Here, the sets of admissible states and inputs are given by

$$\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_N, \tag{2.12}$$

$$\mathbb{U} = \mathbb{U}_1 \times \cdots \times \mathbb{U}_N, \tag{2.13}$$

with $\mathbb{X}_i$ and $\mathbb{U}_i$ as defined in (2.2) and (2.3), respectively.

For linear systems, the collective model matrices become

$$A = \text{blkdiag}(A_1, \ldots, A_N), \quad B = \text{blkdiag}(B_1, \ldots, B_N).$$

The collective positions and velocities are denoted as $q = \text{col}(q_1, \ldots, q_N) \in \mathbb{R}^{n_\text{d} N}$ and $p = \text{col}(p_1, \ldots, p_N) \in \mathbb{R}^{n_\text{d} N}$, respectively.

2.2.3 Graph Theory

In the context of multi-agent systems, graphs can be used to describe the interactions between the individual agents. Following the definitions in Diestel (2017) and Mesbahi and Egerstedt (2010), a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by its vertex set $\mathcal{V} = \{v_1, ...v_N\}$ and the edge set $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V},\ i \neq j\}$. Multi-agent systems can be represented by a graph by considering the agents as vertices and the interactions as the edges. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is undirected if $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E}$, $\forall i, j \in \mathcal{V}$. It is said to be directed otherwise. In the remainder of this thesis, undirected graphs are considered if not explicitly stated otherwise.

A path $\mathcal{P} = (\mathcal{V}, \mathcal{E})$ is a non-empty graph of the form $\mathcal{V} = \{v_1, \ldots, v_k\}$ and $\mathcal{E} = \{(1, 2), \ldots, (k-1, k)\}$ with distinct vertices v_i linking v_1 and v_k. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called connected if it is non-empty and any two of its vertices $v_i, v_j \in \mathcal{V}$ are linked by a path in $\mathcal{G}$. It is called disconnected otherwise. Examples for connected and disconnected graphs are provided in Fig. 2.1. While the graph in Fig. 2.1a is connected, the graph in Fig. 2.1b is not since there is no path connecting v_1 to any of the other vertices. Let

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\} \tag{2.14}$$

denote the neighborhood of vertex v_i. The number of neighboring vertices is given by $|\mathcal{N}_i|$, the cardinality of the neighbor set.

Proximity Graphs

In the context of the dynamic multi-agent systems considered in this thesis, it is useful to introduce the concept of proximity graphs, also referred to as disk graphs. Let $r > 0$ be a sensing or interaction range. A proximity graph $\mathcal{G}(q) = (\mathcal{V}, \mathcal{E}(q))$ is defined by the position-dependent edge-set $\mathcal{E}(q) = \{(i,j) \in \mathcal{V} \times \mathcal{V} : \|q_j - q_i\| < r,\ i \neq j\}$. The neighborhood of a vertex is then defined analogous to (2.14) as

$$\mathcal{N}_i(q) = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}(q)\}, \tag{2.15}$$

with $|\mathcal{N}_i(q)|$ denoting its cardinality. For dynamic agents with $q = q(t)$, the edge set of the proximity graph can change over time depending of the dynamics of the agents, i.e. $\mathcal{E}(q) = \mathcal{E}(q(t))$ for continuous time MASs or $\mathcal{E}(q) = \mathcal{E}(q(k))$ in the discrete-time case.

Graphs and Matrices

Some properties of graphs can also be expressed in terms of matrices. In the context of this thesis, the most important of these properties are the adjacency matrix $A(\mathcal{G})$, the degree matrix $\Delta(\mathcal{G})$, and the graph Laplacian $L(\mathcal{G})$.

The adjacency matrix $A(\mathcal{G})$ is a $N \times N$ matrix describing the adjacency relationships in the graph with its elements defined as

$$[A(\mathcal{G})]_{ij} = \begin{cases} a_{ij}, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases} \tag{2.16}$$

with $a_{ij} \in \mathbb{R}_{\geq 0}$. A graph is said to be unweighted if $a_{ij} = 1$ for all $(i,j) \in \mathcal{E}$. It is said to be weighted otherwise. An example for a weighted graph is provided in Fig. 2.1b where the numbers next to each edge denote the corresponding weights. On the other hand, the graph in Fig. 2.1a is unweighted.

The degree matrix $\Delta(\mathcal{G}) = \mathrm{diag}(d(v_1), \ldots, d(v_N))$ is a diagonal matrix containing the degree of each vertex given by

$$d(v_i) = \sum_{j \in \mathcal{N}_i} a_{ij}. \tag{2.17}$$

(a) Connected graph (unweighted). (b) Disconnected graph (weighted).

Figure 2.1: Examples of graphs.

In the case of unweighted graphs, the degree of a vertex is equal to the cardinality of its neighbor set.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can also be represented by the graph Laplacian $L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G})$. For undirected graphs, this matrix has the property that the individual rows and columns sum up to zero. The Laplacian lifted to higher dimensions is denoted as $L_{(n)} = L \otimes 1_n$. For the unweighted and connected graph in Fig. 2.1a, the adjacency matrix, degree matrix, and Laplacian are

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \; \Delta(\mathcal{G}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \; L(\mathcal{G}) = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix},$$

and for the weighted and disconnected graph in Fig. 2.1b, they are given by

$$A(\mathcal{G}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1.5 \\ 0 & 1.5 & 0 & 3 \\ 0 & 2 & 3 & 0 \end{bmatrix}, \; \Delta(\mathcal{G}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}, \; L(\mathcal{G}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & -2 & -1 \\ 0 & -2 & 5 & -3 \\ 0 & -1 & -3 & 4 \end{bmatrix}.$$

For proximity graphs, $\Delta(\mathcal{G}(q))$, $A(\mathcal{G}(q))$, and $L(\mathcal{G}(q))$ are defined analogously.

2.2.4 Stability

This section provides definitions and conditions for establishing stability used throughout this thesis, following the notation and definitions in Rawlings et al. (2020). Consider the autonomous system

$$x(k+1) = f(x(k)), \tag{2.18}$$

with initial state $x(0) = x_0$, $f : \mathbb{X} \to \mathbb{R}^n$ being locally bounded ($\forall x_0 \in \mathbb{X} \; \exists r, R > 0 : \|f(x)\| < R \; \forall x \in \mathcal{B}^{\circ}_{x_0}(r)$), and $\mathbb{X} \subseteq \mathbb{R}^n$. Furthermore, let

$$x(k) = \phi(k; x(0)), \quad k \in \mathbb{N},$$

denote the solution of (2.18) at time k.

To define stability, consider the following definitions of positive-invariant sets in Definition 2.1 and the comparison functions in Definition 2.2.

Definition 2.1 (Positive-invariant set)**.** A set $\mathcal{A} \subseteq \mathbb{R}^n$ is positive invariant under the autonomous system $x(k+1) = f(x(k))$ if $x(k) \in \mathcal{A}$ implies $\phi(k', x(k)) \in \mathcal{A}$, $\forall k' \in \mathbb{I}_{k:\infty}$.

Definition 2.2 (Comparison functions)**.**

a) A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class $\mathcal{PD}$ if it is continuous, $\alpha(0) = 0$, and $\alpha(r) > 0$ for all $r \in \mathbb{R}_{\geq 0} \setminus \{0\}$.

b) A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class $\mathcal{K}$ if it is in class $\mathcal{PD}$ and is strictly increasing.

c) A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ belongs to class $\mathcal{K}_\infty$ if it is in class $\mathcal{K}$ and $\lim_{r\to\infty} \alpha(r) = \infty$.

d) A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ belongs to class $\mathcal{KL}$ if it is continuous, $\beta(\cdot, k)$ belongs to class $\mathcal{K}$ for all $k \in \mathbb{N}$, and $\beta(r, \cdot)$ is non-increasing with $\lim_{k\to\infty} \beta(r, k) = 0$ for all $r \in \mathbb{R}_{\geq 0}$.

Note that $\mathcal{K}_\infty \subset \mathcal{K} \subset \mathcal{PD}$. Using the above definitions, asymptotic stability of a positive invariant set can be defined according to Rawlings et al. (2020, Appendix B). Recall that $|x|_\mathcal{A}$ denotes the distance from a point x to a set $\mathcal{A}$ (see Section 2.2.1).

Definition 2.3 (Asymptotic stability)**.** Let $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathbb{X}$ be closed and positive invariant for $x(k+1) = f(x(k))$. Then, $\mathcal{A}$ is asymptotically stable for $x(k+1) = f(x(k))$ in $\mathbb{X}$ if there exists a function $\beta \in \mathcal{KL}$ such that for each $x(0) \in \mathbb{X}$

$$|\phi(k; x(0))|_\mathcal{A} \leq \beta(|x(0)|_\mathcal{A}, k), \quad \forall k \in \mathbb{N}.$$

If $\mathbb{X} = \mathbb{R}^n$, $\mathcal{A}$ is globally asymptotically stable for $x(k+1) = f(x(k))$.

In general, showing the existence of the class $\mathcal{KL}$ function in Definition 2.3 is not trivial. However, if there exists a Lyapunov function according to Definition 2.4, local and global asymptotic stability of the system $x(k+1) = f(x(k))$ can be established via the following theorem from Rawlings et al. (2020, Theorem B.15 and B.18).

Definition 2.4 (Lyapunov function)**.** Suppose that $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathbb{X}$ are closed and positive invariant for $x(k+1) = f(x(k))$, with f being locally bounded. The function $V : \mathbb{X} \to \mathbb{R}_{\geq 0}$ is said to be a Lyapunov function in $\mathbb{X}$ for $x(k+1) = f(x(k))$ and $\mathcal{A}$ if there exist $\alpha_1, \alpha_2 \in \mathcal{K}$ and $\alpha_3 \in \mathcal{PD}$, such that, for any $x \in \mathbb{X}$,

$$V(x) \geq \alpha_1(|x|_\mathcal{A}), \tag{2.19}$$

$$V(x) \leq \alpha_2(|x|_\mathcal{A}), \tag{2.20}$$

$$V(x) - V(f(x)) \geq \alpha_3(|x|_\mathcal{A}). \tag{2.21}$$

Theorem 2.1. *Suppose that $\mathbb{X} \subseteq \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathbb{X}$ are closed and positive invariant for $x(k+1) = f(x(k))$, with f being locally bounded. If $V(x)$ is a Lyapunov function in $\mathbb{X}$ for $x(k+1) = f(x(k))$ and $\mathcal{A}$, then $\mathcal{A}$ is asymptotically stable in $\mathbb{X}$. If $\mathbb{X} = \mathbb{R}^n$, $\mathcal{A}$ is globally asymptotically stable.*

The proofs are provided in Rawlings et al. (2020).

Remark. From the definitions above, stability of a single equilibrium x^* can be established by defining $\mathcal{A} = x^*$, resulting in $|x|_\mathcal{A} = \|x - x^*\|$.

2.3 Flocking Control

This section provides an introduction to flocking control as presented in Olfati-Saber (2006). Agent interactions are modeled via a proximity graph $\mathcal{G}(q) = (\mathcal{V}, \mathcal{E}(q))$ with an

interaction range $r > 0$. In Olfati-Saber (2006), the primary control objective is for the agents to form a cohesive flock according to the definitions below.

Definition 2.5 (Flock)**.** A group of agents is called a flock over the time interval $[t_0, t_f)$, $t_f \geq t_0$, if the proximity graph $\mathcal{G}(q(t))$ is connected over $[t_0, t_f)$.

Definition 2.6 (Cohesion)**.** A group of agents is cohesive over the time interval $[t_0, t_f)$, $t_f \geq t_0$, if there exists an $R > 0$ such that, for all $t \in [t_0, t_f)$,

$$\left\| q(t) - 1_N \otimes \left(\frac{1}{N} \sum_{i=1}^{N} q(t) \right) \right\| \leq R.$$

This objective is achieved by designing a distributed control law implementing Reynold's rules of cohesion, separation, and alignment (Reynolds 1987). In Olfati-Saber (2006), the author considers N continuous-time agents with n_d-dimensional double-integral behavior as defined in (2.8). The collective dynamics are then given by

$$\begin{aligned} \dot{q}(t) &= p(t), \\ \dot{p}(t) &= u(t). \end{aligned} \tag{2.22}$$

2.3.1 α-Lattices and σ-Maps

The goal in Olfati-Saber's flocking framework is for the swarm to achieve a configuration where all agents have the same desired inter-agent distance $d > 0$ to all of their neighbors. This desired formation is referred to as α-lattice. Configurations that are close to an α-lattice are referred to as quasi α-lattices.

Definition 2.7 (α-lattice)**.** Given a proximity graph $\mathcal{G}(q) = (\mathcal{V}, \mathcal{E}(q))$ with $r > d > 0$, an α-lattice is a configuration q satisfying the constraint $\|q_j - q_i\| = d$, $\forall (i,j) \in \mathcal{E}(q)$.

Definition 2.8 (Quasi α-lattice)**.** Given a proximity graph $\mathcal{G}(q) = (\mathcal{V}, \mathcal{E}(q))$ with $r > d > 0$ and a small perturbation $0 < \eta \ll d$, a quasi α-lattice is a configuration q satisfying $-\eta < \|q_j - q_i\| - d < \eta$, $\forall (i,j) \in \mathcal{E}(q)$.

Note that these definitions do not require the underlying graph to be connected. To ensure that the resulting lattice is planar, the ratio of the desired distance d and the interaction range r has to be chosen carefully. It can be shown that planar lattices are formed for $r < \sqrt{2}d$ in 2D and $r < \sqrt{3}d$ in 3D (Olfati-Saber 2006).

To quantify the deviation from the desired configuration, define

$$\delta(q_i, q_j) = \delta_{ij} = \|q_j - q_i\| - d. \tag{2.23}$$

Abbreviated notations $q_{ij} = q_j - q_i$ and $p_{ij} = p_j - p_i$ are also used to simplify the notation. As δ involves the Euclidean norm, it is not differentiable at $q_{ij} = 0$. To obtain a differentiable measure for the deviation from the desired distance, in Olfati-Saber (2006), the author replaces the norm by the σ-map $\sigma_s : \mathbb{R}^{n_\mathrm{d}} \to \mathbb{R}_{\geq 0}$,

$$\sigma_s(z) = \frac{\sqrt{1 + s\|z\|^2} - 1}{s}, \tag{2.24}$$

where $s > 0$ is a tuning parameter of the σ-map. The derivative of the σ-map is given by

$$\nabla \sigma_s(z) = \frac{z}{\sqrt{1 + s\|z\|^2}}. \tag{2.25}$$

Even though this σ-map is sometimes also referred to as σ-norm, it is not a norm, as it does not satisfy the properties of scalar multiplication and the triangle inequality. Figure 2.2 provides a comparison of the σ-map, the Euclidean norm, and their derivatives for different values of the parameter s.

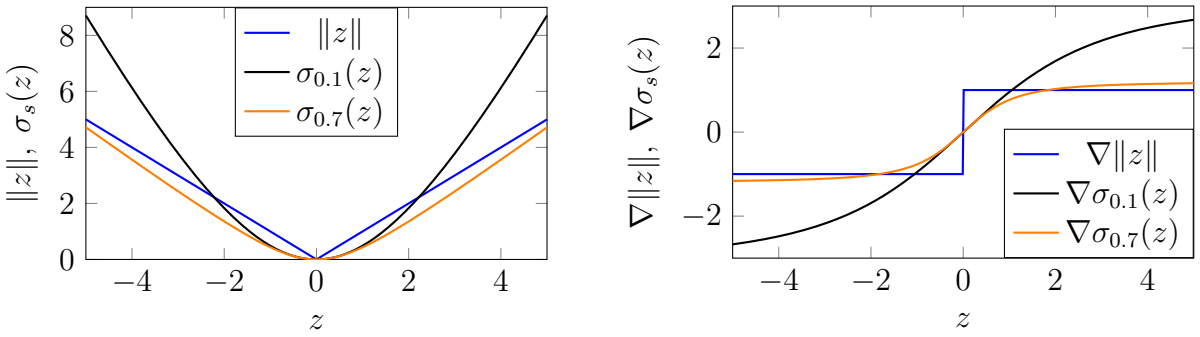
2.3.2 Flocking Protocols

In the framework presented in Olfati-Saber (2006), the flocking protocols are described utilizing different kinds of agents: the agents of the swarm (α-agents), obstacles (β-agents), and group objectives (γ-agents). Depending on which kinds of agents are considered, different flocking protocols are derived. In free flocking (α-flocking), only interactions between α-agents are considered. Flocking with group objectives (α-γ-flocking) additionally considers α-γ-interactions. Finally, in flocking with obstacle avoidance (α-β-γ-flocking), interactions between α- and β-agents are considered as well. In the following, these protocols will be introduced.

α-Flocking

In Olfati-Saber (2006), the key ingredient for designing a control law exhibiting the desired behavior of cohesion and separation is a smooth collective potential function $V_\alpha : \mathbb{R}^{n_\mathrm{d} N} \to \mathbb{R}_{\geq 0}$. To obtain a potential with a finite cutoff, the edges of the graph are weighted depending on the distances between agents. Let $\rho_h : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the continuously differentiable bump function

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h), \\ \frac{1}{2}\left(1 + \cos\left(\pi \frac{z-h}{1-h}\right)\right), & z \in [h, 1), \\ 0, & \text{otherwise}, \end{cases} \tag{2.26}$$



(a) Comparison of $\|z\|$ and $\sigma_s(z)$.

(b) Comparison of $\nabla\|z\|$ and $\nabla\sigma_s(z)$.

Figure 2.2: Comparison of Euclidean norm and σ-map for different parameters.

with shaping parameter $h \in (0,1)$. The elements of the adjacency matrix are then assigned as

$$a_{ij}\Big(\sigma_s\big(q_{ij}(t)\big)\Big) = \rho_{h_\alpha}\left(\frac{\sigma_s\big(q_{ij}(t)\big)}{r_\alpha}\right), \tag{2.27}$$

where h_α is the shaping parameter for agent-agent interactions and r_α denotes the σ-map of the interaction range, $r_\alpha = \sigma_s(r)$.

To define the collective potential V_α, consider first the action function $\phi_\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}$,

$$\phi_\alpha(z) = \rho_{h_\alpha}\left(\frac{z}{r_\alpha}\right)\frac{z - d_\alpha}{\sqrt{1 + \|z - d_\alpha\|^2}} = \rho_{h_\alpha}\left(\frac{z}{r_\alpha}\right)\nabla\sigma_1(z - d_\alpha), \tag{2.28}$$

with d_α denoting the σ-map of the desired inter-agent distance, where $d_\alpha = \sigma_s(d)$. By defining a pairwise inter-agent potential $\psi_\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ as

$$\psi_\alpha(z) = \int_{d_\alpha}^{z} \phi_\alpha(y)\,\mathrm{d}y, \tag{2.29}$$

the collective potential function is then constructed by summing the pairwise potentials of all agents

$$V_\alpha(q(t)) = \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}, j\neq i}\psi_\alpha\left(\sigma_s(q_{ij}(t))\right). \tag{2.30}$$

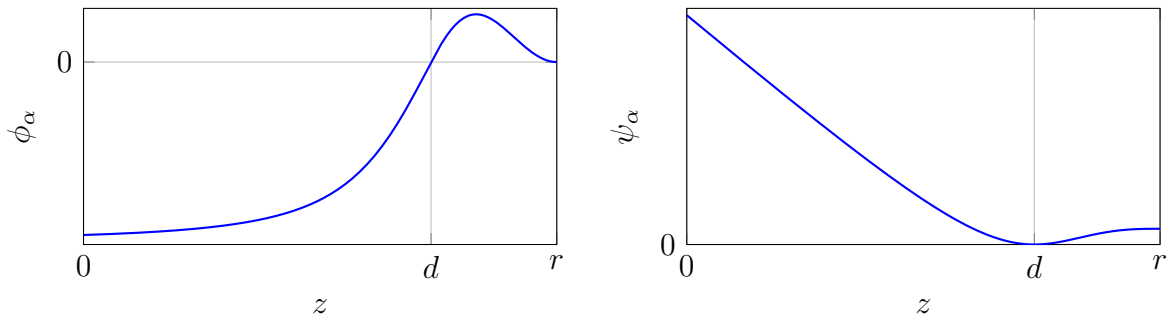
The qualitative shapes of ψ_α and ϕ_α are depicted in Fig. 2.3.

The protocol for free flocking is then given by

$$u_i^\alpha(t) = -c_1^\alpha \sum_{j\in\mathcal{N}_i(q(t))}\nabla\psi_\alpha\Big(\sigma_s\big(q_{ij}(t)\big)\Big) + c_2^\alpha \sum_{j\in\mathcal{N}_i(q(t))} a_{ij}\Big(\sigma_s\big(q_{ij}(t)\big)\Big)p_{ij}(t), \tag{2.31}$$

where $c_1^\alpha, c_2^\alpha > 0$ are the tuning parameters. The first term corresponds to the cohesion and separation rules, and the second term corresponds to the velocity alignment. The derivative of the agent-wise potential with respect to the positions is given by

$$\nabla\psi_\alpha\Big(\sigma_s\big(q_{ij}(t)\big)\Big) = -\phi_\alpha\big(\sigma_s(q_{ij}(t)\big)\frac{q_{ij}}{\sqrt{1 + s\|q_{ij}\|^2}}. \tag{2.32}$$



(a) Action function ϕ_α.

(b) Pairwise potential ψ_α.

Figure 2.3: Olfati-Saber pairwise potential and action function.

In terms of the collective dynamics, the control law becomes

$$u^\alpha(t) = -c_1^\alpha \nabla V_\alpha(q(t)) - c_2^\alpha L_{(n_d)}\big(q(t)\big)p(t). \tag{2.33}$$

Remark. In Olfati-Saber (2006), the action function ψ_α in (2.28) is defined using additional shaping parameters a, b, and c. The definition in (2.28) corresponds to the case $a = b$ where the scaling is absorbed in the weighting parameter c_1^α.

α-γ-Flocking

In Olfati-Saber (2006), group objectives are modeled in the form of virtual γ-agents with reference position $q_{\mathrm{r}_i}(t) \in \mathbb{R}^{n_\mathrm{d}}$ and reference velocity $p_{\mathrm{r}_i}(t) \in \mathbb{R}^{n_\mathrm{d}}$. The γ-flocking component can then be modeled as a function of the states of the agent and the γ-agent,

$$u_i^\gamma = f_i^\gamma(q_i, p_i, q_{\mathrm{r}_i}, p_{\mathrm{r}_i}).$$

The collective control law is then given by $u^\gamma = \mathrm{col}(u_1^\gamma, \ldots, u_N^\gamma)$. Then, the agent-wise and collective flocking protocols are given by

$$u_i(t) = u_i^\alpha(t) + u_i^\gamma(t) \tag{2.34}$$

and

$$u(t) = u^\alpha(t) + u^\gamma(t), \tag{2.35}$$

respectively.

α-β-γ-Flocking

Consider N^β static, spherical obstacles characterized by their center states $x_o^\beta = \mathrm{col}(q_o^\beta, 0) \in \mathbb{R}^{2n_\mathrm{d}}$ and radii $r_o^\beta > 0$, $o \in \mathbb{I}_{1:N^\beta}$. The set of all obstacles is defined as $\mathcal{V}^\beta = \{v_1^\beta, \ldots, v_{N_\beta}^\beta\}$. For each agent i and obstacle o, a virtual β-agent is generated on the boundary of the obstacle at the position $q_o^{\beta_i}(q_i(t))$ closest to agent i. This concept is depicted in Fig. 2.4 where the generated β-agents are depicted for two agents and two obstacles. According to Olfati-Saber (2006, Lemma 4), the position and velocity of the β-agent can be computed as

$$q_o^{\beta_i} = \mu_o^i q_i + (1 - \mu_o^i) q_o^\beta, \quad p_o^{\beta_i} = \mu_o^i (I - a_o^i a_o^{i\top}) p_i \tag{2.36}$$

with

$$\mu_o^i = \frac{r_o^\beta}{\|q_i - q_o^\beta\|}, \quad a_o^i = \frac{q_i - q_o^\beta}{\|q_i - q_o^\beta\|}.$$

Defining $q_{io}(t) = q_o^{\beta_i}(t) - q_i(t)$ and $p_{io}(t) = p_o^{\beta_i}(t) - p_i(t)$, obstacle avoidance can be integrated in the flocking protocol by defining an obstacle-avoidance potential function (Olfati-Saber 2006).

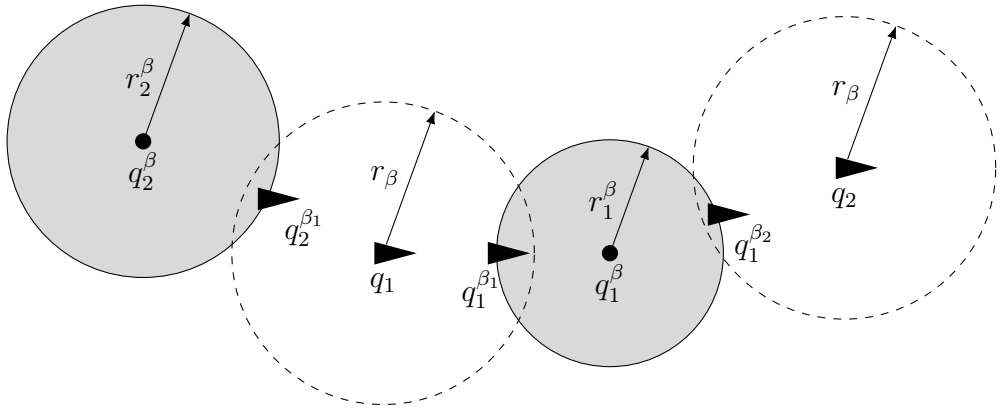


Figure 2.4: Two obstacles detected by two agents and corresponding β-agents.

For each agent i, define the set of β-neighbors as

$$\mathcal{N}_i^\beta(q_i) = \{o \in \mathcal{V}^\beta : \|q_{io}\| < r_o\}, \tag{2.37}$$

where $r_o > 0$ is the obstacle detection range. Let $d_o < r_o$ be the desired obstacle separation. The σ-maps of r_o and d_o are given by $r_\beta = \sigma_s(r_o)$ and $d_\beta = \sigma_s(d_o)$. The collective β-potential $V_\beta : \mathbb{R}^{n_\mathrm{d} N} \to \mathbb{R}_{\geq 0}$, the pairwise potential $\psi_\beta : \mathbb{R}^{n_\mathrm{d}} \to \mathbb{R}_{\geq 0}$, and the action function $\phi_\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}$ are defined analogous to (2.28)-(2.29) as

$$\phi_\beta(z) = \rho_{h_\beta}\left(\frac{z}{d_\beta}\right)\left(\nabla\sigma_1(z - d_\beta) - 1\right), \tag{2.38}$$

$$\psi_\beta(z) = \int_{d_\beta}^{z} \phi_\beta(y)\, \mathrm{d}y, \tag{2.39}$$

and

$$V_\beta(q(t)) = \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{o\in\mathcal{V}^\beta} \psi_\beta\left(\sigma_s(q_{io}(t))\right). \tag{2.40}$$

Here, h_β denotes the shaping parameter for the bump function for agent-obstacle interactions. The qualitative behavior of ψ_β and ϕ_β is depicted in Fig. 2.5. The agent-wise and collective flocking protocols with obstacle avoidance are then given by

$$u_i(t) = u_i^\alpha(t) + u_i^\beta(t) + u_i^\gamma(t) \tag{2.41}$$

and

$$u(t) = u^\alpha(t) + u^\beta(t) + u^\gamma(t), \tag{2.42}$$

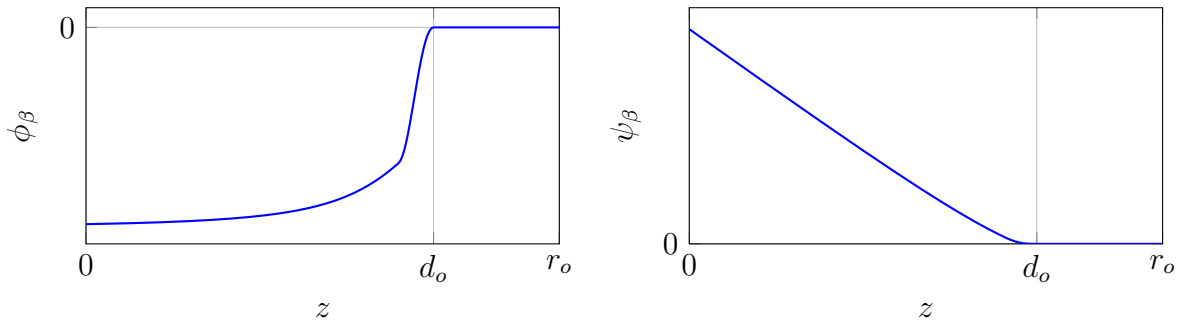
respectively, with

$$u_i^\beta(t) = -c_1^\beta \sum_{o\in\mathcal{N}_i^\beta(t)} \nabla\psi_\beta\big(\sigma_s(q_{io}(t))\big) + c_2^\beta \sum_{o\in\mathcal{N}_i^\beta(t)} \rho_{h_\beta}\left(\frac{\sigma_s(q_{io}(t))}{d_\beta}\right) p_{io}(t) \tag{2.43}$$

and

$$u^\beta(t) = -c_1^\beta \nabla V_\beta(q(t)) - c_2^\beta \sum_{i\in\mathcal{V}}\sum_{o\in\mathcal{V}^\beta} \rho_{h_\beta}\left(\frac{\sigma_s(q_{io}(t))}{d_\beta}\right) p_{io}(t). \tag{2.44}$$

Here, $c_1^\beta, c_2^\beta > 0$ are the obstacle-avoidance tuning parameters.



(a) Action function ϕ_β.

(b) Pairwise potential ψ_β.

Figure 2.5: Olfati-Saber pairwise potential and action function for obstacle avoidance.

2.4 Model Predictive Control

This section introduces the general concepts and notations for model predictive control (MPC) used throughout this thesis, including problem formulations for different MPC architectures in the context of MASs. Furthermore, a general line of reasoning for the MPC stability analysis is presented.

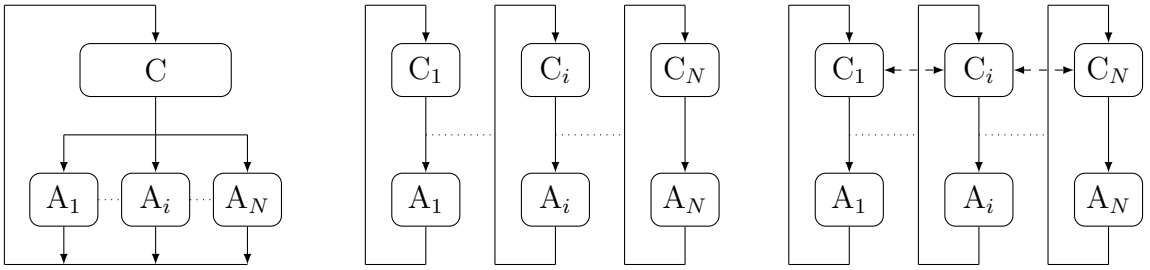
In MPC, at each time k, the current system state $x(k)$ and a model of the system are used to predict the future system behavior. The number of steps the system behavior is predicted in the future is referred to as the prediction horizon M. This prediction of the system behavior is then utilized to obtain the best control input for time k. Here, "best" refers to the solution of an optimization problem with a cost function that penalizes deviations from the desired system behavior subject to additional state and input constraints. The optimality with respect to a chosen performance index while considering constraints is one of the main advantages of MPC over other control techniques. The notation used in this thesis follows the notation and definitions in Rawlings et al. (2020). If not stated otherwise, this section considers homogeneous MASs with all agents subject to the same dynamics and constraints.

2.4.1 Model Predictive Control Architectures

In the context of predictive control for multi-agent systems, one important property is the underlying controller architecture. In this thesis, three different approaches are distinguished: centralized control, decentralized control, and distributed control. The general structure of these architectures is depicted in Fig. 2.6, where the controllers are denoted by C and the agents by A. A comprehensive study of the features of different MPC architectures is provided in Negenborn and Maestre (2014).

Centralized Model Predictive Control

In centralized model predictive control (CMPC), as depicted in Fig. 2.6a, a single predictive controller C is designed for all agents A_i. This controller then solves one large optimization



(a) Centralized architecture. (b) Decentralized architecture. (c) Distributed architecture.

Figure 2.6: MPC control architectures.

problem to obtain the optimal inputs for all agents simultaneously. As all information is available to the centralized controller, this architecture does not require information exchange between individual agents.

Consider a MAS governed by the general collective dynamics in (2.11). Let

$$U(k) = \mathrm{col}\big(u(k|k), \dots, u(k+M-1|k)\big) \in \mathcal{U} \tag{2.45}$$

be the collective control input over the prediction horizon, where the constraint set for the input sequence is given by $\mathcal{U} = \mathbb{U} \times \cdots \times \mathbb{U} = \mathbb{U}^M$. The notations $*(k+m|k)$ and $*_{k+m|k}$ are used to denote a prediction for time $k+m$ with information available up to time k. Given an initial state $x(k|k) = x(k)$ and an input sequence $U(k)$, the solution of the collective dynamics (2.11) at time $k+m$ is given by

$$x(k+m|k) = \phi(m; x(k), U(k)), \quad m \in \mathbb{I}_{0:M}. \tag{2.46}$$

The predicted sequence of length M starting at $x(k)$ is then denoted as

$$X(k) = \mathrm{col}(x(k), \dots, x(k+M-1|k)) \in \mathcal{X}. \tag{2.47}$$

The corresponding constraint set is given by $\mathcal{X} = \mathbb{X} \times \cdots \times \mathbb{X} = \mathbb{X}^M$.

With the above definitions, the centralized MPC problem is stated.

Problem 2.1 (Centralized MPC)**.**

$$\begin{aligned} V^*(x(k)) = &\min_{U(k)} V(x(k), U(k)) \\ &\text{s.t. } x(k+m) = \phi(m; x(k), U(k)), \quad m \in \mathbb{I}_{0:M}, \\ &\quad X(k) \in \mathcal{X}, \\ &\quad U(k) \in \mathcal{U}, \\ &\quad x(k+M) \in \mathbb{X}_f. \end{aligned}$$

Here, the function $V : \mathbb{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ is the cost function,

$$V(x(k), U(k)) = \sum_{m=0}^{M-1} \ell(x(k+m|k), u(k+m|k)) + V_f(x(k+M|k)), \tag{2.48}$$

with stage cost $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ and terminal cost $V_f : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$. The set $\mathbb{X}_f \subseteq \mathbb{X}$ denotes the terminal set. The terminal cost and set are related to last predicted state $x(k+M|k)$ and play an important role in the stability analysis presented in Section 2.4.2. Analogous to (2.2) and (2.12), the terminal constraint is given by

$$\mathbb{X}_f = \mathbb{X}_{1,f} \times \cdots \times \mathbb{X}_{N,f}, \tag{2.49}$$

with

$$\mathbb{X}_{i,f} = \{x_i \in \mathbb{R}^{n_{x,i}} : h_{i,f}^{\text{eq}}(x_i) = 0, h_{i,f}^{\text{ieq}}(x_i) \leq 0\} \subseteq \mathbb{X}_i. \tag{2.50}$$

Here, $h_{i,f}^{\text{eq}} : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{c_{i,f}^{\text{eq}}}$ and $h_{i,f}^{\text{ieq}} : \mathbb{R}^{n_{x,i}} \to \mathbb{R}^{c_{i,f}^{\text{ieq}}}$ denote the equality and inequality constraint functions, with $c_{i,f}^{\text{eq}}$ and $c_{i,f}^{\text{ieq}}$ being the numbers of constraints.

Let $U^*(k) = \text{col}(u^*(k|k), \ldots, u^*(k+M-1|k))$ denote the optimal solution to Problem 2.1 at time k. The centralized MPC control law is then given by

$$u(k) = \kappa(x(k)) = u^*(k|k). \tag{2.51}$$

The corresponding optimal cost $V^*(x(k)) = V(x(k), U^*(k))$ is also referred to as the value function.

As the centralized controller has access to the information of all the individual agents, CMPC can obtain the globally optimal control inputs. However, depending on how the states and inputs are coupled via the cost and constraints, and considering the large number of decision variables, solving the CMPC problem can be computationally expensive.

Decentralized Model Predictive Control

In decentralized MPC, as depicted in Fig. 2.6b, each agent A_i is equipped with a local controller C_i. Consequently, instead of solving one large optimization problem, each agent solves a local optimization problem considering only its own states and decision variables. Let the individual agents be governed by the dynamics in (2.4). The predicted state and input sequences $U_i \in \mathcal{U}_i$ and $X_i \in \mathcal{X}_i$ are defined analogous to (2.45) and (2.47) with the solution at time $k+m$ given by

$$x_i(k+m|k) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \tag{2.52}$$

and

$$\mathcal{U}_i = \mathbb{U}_i^M, \quad \mathcal{X}_i = \mathbb{X}_i^M,$$

denote the corresponding state and input constraint sets. The decentralized MPC problem can then be stated as follows:

Problem 2.2 (Decentralized MPC)**.**

$$\begin{aligned} V_i^*(x(k)) = \min_{U_i(k)} \; & V_i(x_i(k), U_i(k)) \\ \text{s.t. } & x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ & X_i(k) \in \mathcal{X}_i, \\ & U_i(k) \in \mathcal{U}_i, \\ & x_i(k+M) \in \mathbb{X}_{i,f}. \end{aligned}$$

Here, the function $V : \mathbb{X}_i \times \mathcal{U}_i \to \mathbb{R}_{\geq 0}$ is the local cost function,

$$V_i(x_i(k), U_i(k)) = \sum_{m=0}^{M-1} \ell_i(x_i(k+m|k), u_i(k+m|k)) + V_{i,f}(x_i(k+M|k)), \tag{2.53}$$

with the local stage cost $\ell_i : \mathbb{X}_i \times \mathbb{U}_i \to \mathbb{R}_{\geq 0}$, terminal cost $V_{i,f} : \mathbb{X}_{i,f} \to \mathbb{R}_{\geq 0}$, and terminal set $\mathbb{X}_{i,f} \subseteq \mathbb{X}_i$. Let $U_i^*(k) = \mathrm{col}(u_i^*(k|k), \ldots, u_i^*(k+M-1|k))$ denote the optimal solution to Problem 2.2 for agent i at time k. The corresponding value function is denoted as $V_i^*(x_i(k)) = V_i(x_i(k), U_i^*(k))$.

The decentralized MPC control law for agent i is then given by

$$u_i(k) = \kappa_i(x(k)) = u_i^*(k|k). \tag{2.54}$$

Unlike the CMPC problem in Problem 2.1, the decentralized optimization problems only have the size of a single agent, reducing the computational complexity. On the other hand, as couplings in the costs and constraints of the individual agents are not considered in the decentralized framework, the obtained solution is only locally optimal and not globally optimal as the solution of the CMPC problem.

Distributed Model Predictive Control

As in the decentralized case, in the distributed model predictive control (DMPC) architecture, each agent A_i is equipped with a local controller C_i. This is depicted in Fig. 2.6c. Contrary to decentralized MPC, in the DMPC framework, the local controllers can exchange information, allowing for the modeling of couplings in costs and constraints. Let the individual agents be governed by the dynamics in (2.4) with agent interactions being described by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{N}_i = \{j_1^i, \ldots, j_{|\mathcal{N}_i|}^i\}$ denote the set of neighbors of agent i. The state vector of the neighbors of an agent i is then denoted as

$$x_{\bar{i}} = \mathrm{col}(x_{j_1^i}, \ldots, x_{j_{|\mathcal{N}_i|}^i}) \in \mathbb{X}_{\bar{i}},$$

with $\mathbb{X}_{\bar{i}} = \mathbb{X}_{j_1^i} \times \cdots \times \mathbb{X}_{j_{|\mathcal{N}_i|}^i}$. Together with the state of agent i, the vector

$$\bar{x}_i = \mathrm{col}(x_i, x_{\bar{i}}) \in \bar{\mathbb{X}}_i$$

denotes the state of the i^{th} subsystem with

$$\bar{\mathbb{X}}_i \subseteq \mathbb{X} \times \mathbb{X}_{\bar{i}}. \tag{2.55}$$

Neighborhood and subsystem inputs ($u_{\bar{i}}$, $\bar{u}_i$), positions ($q_{\bar{i}}$, $\bar{q}_i$), and velocities ($p_{\bar{i}}$, $\bar{p}_i$) are defined analogously. The corresponding state sequences for the neighborhood and subsystem of agent i are denoted as $X_{\bar{i}}$ and $\bar{X}_i$, respectively, and are defined analogous to (2.47).

In general, MASs can be coupled through their states, inputs, outputs, cost, and/or constraints. Here, it is assumed that the considered MASs consist of autonomous entities

that are not coupled in their dynamics but only through the cost and state constraints. In (2.55), this is indicated by the $\subseteq$ relation. Let

$$X_{\bar{i}}(k) = \operatorname{col}(x_{\bar{i}}(k|k), \dots, x_{\bar{i}}(k+M-1|k))$$

and

$$X_{\bar{i}}^{M}(k) = \operatorname{col}(x_{\bar{i}}(k|k), \dots, x_{\bar{i}}(k+M|k))$$

denote the sequences of neighbor states, where $X_{\bar{i}}^{M}$ additionally contains the terminal neighbor state at time $k+m$. The constraint set for the predicted state sequence is given by

$$\bar{\mathcal{X}}_i = \bar{\mathbb{X}}_i^M. \tag{2.56}$$

With these definitions, the distributed MPC problem can then be stated as follows:

Problem 2.3 (Distributed MPC)**.**

$$\begin{aligned} V_i^*(x_i(k), X_{\bar{i}}^{M}(k)) = \min_{U_i(k)} \; & V_i(x_i(k), X_{\bar{i}}^{M}(k), U_i(k)) \\ \text{s.t. } & x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ & (X_i(k), X_{\bar{i}}(k)) \in \bar{\mathcal{X}}_i, \\ & U_i(k) \in \mathcal{U}_i, \\ & (x_i(k+M), x_{\bar{i}}(k+M)) \in \bar{\mathbb{X}}_{i,f} \end{aligned}$$

With the dependence on the states of agents in the neighborhood, each agent's cost function is given by

$$\begin{aligned} V_i(x_i(k), X_{\bar{i}}^{M}(k), U_i(k)) = & \sum_{m=0}^{M-1} \ell_i\big(x_i(k+m|k), x_{\bar{i}}(k+m), u_i(k+m|k)\big) \\ & + V_{i,f}\big(x_i(k+M|k), x_{\bar{i}}(k+M)\big). \end{aligned} \tag{2.57}$$

The coupled terminal set is given by $\bar{\mathbb{X}}_{i,f} \subseteq \mathbb{X} \times \mathbb{X}_{\bar{i}}$.

The distributed MPC control law for agent i is then given by

$$u_i(k) = \kappa_i(x(k), X_{\bar{i}}^{M}(k)) = u_i^*(k|k), \tag{2.58}$$

with value function $V_i^*(x_i(k), X_{\bar{i}}^{M}(k)) = V_i(x_i(k), X_{\bar{i}}^{M}(k), U_i^*(k))$.

Contrary to decentralized MPC, distributed MPC can consider couplings between the agents' costs and constraints at the price of requiring an exchange of information between agents. Although the solution of the DMPC problem is not necessarily equal to the globally optimal solution, the additional information considered in the optimization problems can reduce the mismatch between the locally and globally optimal solutions compared to the decentralized architecture (Stewart et al. 2010). For some combination of DMPC problem formulations and optimization algorithms, the solution can be shown to converge towards the globally optimal one. A more detailed overview of MPC architectures, algorithms, and their features, can be found in Rawlings et al. (2020, Chapter 6), Negenborn and Maestre (2014), and the references therein.

2.4.2 Model Predictive Control Stability Analysis

Aside from practical performance criteria, theoretical stability guarantees play an important role when comparing different control algorithms. In the case of MPC, stability guarantees are commonly established by the following two steps:

1. Showing that the problem is recursively feasible.
2. Showing that the value function is a Lyapunov function and that Theorem 2.1 can be applied.

Recursive feasibility refers to the property that, if the problem is feasible at time $k = 0$, it is also feasible for all $k \in \mathbb{N}$. Consider the CMPC problem (Problem 2.1), and let $U^*(0) = \mathrm{col}(u^*(0|0), \dots, u^*(M-1|0))$ denote the optimal solution at time $k = 0$. For time $k = 1$, consider the warm start sequence based on the optimal solution at the previous step given by

$$\tilde{U}(k|k-1) = \mathrm{col}\big(u^*(k|k-1), \dots, u^*(k+M-2|k-1), \kappa_M(x(k+M-1|k-1))\big), \quad (2.59)$$

where $\kappa_M : \mathbb{X} \to \mathbb{U}$ is referred to as the terminal control law. Together, the terminal cost, terminal set, and terminal control law are also referred to as the terminal ingredients. By designing these ingredients such that the terminal set is positive-invariant under the terminal control law, the satisfaction of the constraints in Problem 2.1 can be guaranteed for all $k > 0$ if the solution exists at time $k = 0$. For general nonlinear optimization problems with nonlinear constraint sets, the design of these terminal ingredients is a challenging task (Rawlings et al. 2020).

One special set of terminal ingredients is characterized by the terminal equality constraint $\mathbb{X}_f = x^{\mathrm{e}}$. Here, x^{e} is the desired equilibrium state, $f(x^{\mathrm{e}}, u^{\mathrm{e}}) = x^{\mathrm{e}}$, and u^{e} is the corresponding equilibrium input. This constraint enforces that the target state is reached within M steps and allows for setting $V_f = 0$ and $\kappa_M = u^{\mathrm{e}}$. While drastically simplifying the design of the terminal ingredients, imposing a terminal equality constraint reduces the set of feasible initial states to those that can reach the target state within M steps. In general, the set of all feasible states at time $k = 0$ is referred to as the region of attraction $\mathbb{X}_M \subseteq \mathbb{X}$.

For the second part, showing that the value function is a valid Lyapunov function requires showing the existence of α_1, α_2, and α_3 in (2.19)-(2.21). Then, asymptotic stability follows directly from Theorem 2.1. However, for constrained, nonlinear problems, the value function and the control law are not necessarily continuous, making it challenging to show the existence of the comparison functions in (2.19) and (2.20) (Grimm et al. 2004; Rawlings et al. 2020).

In Rawlings et al. (2020), the following assumptions are made to ensure that the value function satisfies the prerequisites in Definition 2.4 (Rawlings et al. 2020, Assumptions 2.2, 2.3). Assume without loss of generality that the desired equilibrium is given by $(x^{\mathrm{e}}, u^{\mathrm{e}}) = (0, 0)$.

Assumption 2.1. The functions $f : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$, $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$, and $V_f : \mathbb{X}_f \to \mathbb{R}_{\geq 0}$ are continuous.

Assumption 2.2. It holds that $f(0,0) = 0$, $\ell(0,0) = 0$, and $V_f(0) = 0$.

Assumption 2.3. The sets $\mathbb{X}$, $\mathbb{X}_f$, and $\mathbb{U}$ contain the origin with $\mathbb{X}$ closed, and $\mathbb{X}_f$ and $\mathbb{U}$ compact.

Note that the extension of Assumption 2.2 and Assumption 2.3 to equilibria other than $(0,0)$ is straightforward. Furthermore, the following assumption is made on the value function (Rawlings et al. 2020, Assumption 2.17).

Assumption 2.4. There exists a function $\alpha \in \mathcal{K}_\infty$ such that

$$V^*(x) \leq \alpha(\|x\|), \quad \forall x \in \mathbb{X}_M.$$

Loosely speaking, Assumption 2.4 ensures that all states that can reach the terminal set within M steps can do so without an excessive cost. For some classes of problems, for example linear systems with polyhedral state constraints, Assumption 2.4 is trivially satisfied by continuity of the value function (Rawlings et al. 2020, Proposition 2.4). The terminal ingredients are related to the stability arguments by the following assumption (Rawlings et al. 2020, Assumption 2.4).

Assumption 2.5. For all $x \in \mathbb{X}_f$, there exists a $u \in \mathbb{U}$ satisfying $f(x,u) \in \mathbb{X}_f$ and

$$V_f(f(x,u)) - V_f(x) \leq -\ell(x,u). \tag{2.60}$$

Furthermore, there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ satisfying

$$\ell(x,u) \geq \alpha_1(\|x\|), \quad \forall x \in \mathbb{X}_M,\ u \in \mathbb{U}, \tag{2.61}$$
$$V_f(x) \leq \alpha_2(\|x\|), \quad \forall x \in \mathbb{X}_f. \tag{2.62}$$

The first part of this assumption ensures that the terminal set is positive invariant under the terminal control law and the second part ensures that the cost is non-increasing in the terminal set. Stability of the origin is then established by the following result.

Theorem 2.2 (Rawlings et al. (2020, Theorem 2.19))**.** *Suppose Assumptions 2.1, 2.2, 2.3, 2.4, and 2.5 are satisfied. Then, the origin is asymptotically stable in $\mathbb{X}_M$ under the control law* (2.51).

The proof can be found in Rawlings et al. (2020) and is based on showing that the value function of the considered MPC problem is a valid Lyapunov function and that Theorem 2.1 can be applied.

The concepts discussed above can also be applied to the decentralized MPC (Problem 2.2) and distributed MPC (Problem 2.3) by choosing either the local value functions V_i^* or their sum $\sum_{i=1}^N V_i^*$ as Lyapunov function candidates. However, depending on the nature of the coupling between the individual agents, the design of the terminal ingredients is not straightforward in the distributed case and can require a coupled design of the terminal ingredients.

3 Model Predictive Flocking and the Challenges in the Stability Analysis

When combining flocking algorithms with MPC, the resulting algorithms inherit the benefits of both frameworks: an optimal, distributed control law for large, self-organized swarms of autonomous agents, capable of handling state and input constraints. The combination of these two concepts is also referred to as model predictive flocking (MPF). Since all approaches to MPF in the related literature use different setups and notations, a unified framework and notation for MPF are presented in Section 3.1. Afterwards, this chapter addresses the stability analysis of MPF, including the general challenges (Section 3.2), and problems in approaches in the related literature (Sections 3.3, 3.4). Based on these findings, a novel approach to the MPF stability analysis is presented in Section 3.5.

3.1 Model Predictive Flocking

As in non-predictive flocking (Section 2.3), the goal of MPF algorithms is to formulate a control law based on the rules of cohesion, separation, and alignment (Reynolds 1987). To achieve this goal, most of the MPF literature is closely related to the flocking protocol in Olfati-Saber (2006), where the objective of the agents is to form an α-lattice (cohesion and separation) while converging towards the same velocity (alignment). While there also exist approaches that do not require the formation of an α-lattice, the majority of contributions in MPF, this chapter focuses on Olfati-Saber modeling of flocking behavior (cf. Zhang et al. (2016), Beaver et al. (2020), Beaver and Malikopoulos (2020)). The MPF results presented in this and the following chapters mainly focus on the first approach. In such a framework, the primary challenge is to cast the flocking problem in the form of the optimization problems presented in Section 2.4.1, where the common approach is to penalize the deviations from the desired inter-agent distances in the cost function. Depending on the chosen MPC architecture, the predictive flocking problem is then formulated as a centralized model predictive flocking (CMPF) or distributed model predictive flocking (DMPF) problem. In the following, unified frameworks for CMPF and DMPF are introduced.

3.1.1 Centralized Model Predictive Flocking

Consider a homogeneous MAS with collective state vector $x = \mathrm{col}(q, p)$, and let $p_{\mathrm{r}_i} \in \mathbb{R}^{n_\mathrm{d}}$ denote the reference velocity for a single agent. Assuming the same reference of all agents, the collective reference velocity is given by $p_\mathrm{r} = 1_N \otimes p_{\mathrm{r}_i}$. In the centralized case, for a given configuration q, the set of all possible configurations satisfying $\delta_{ij} = 0$ for all $(i,j) \in \mathcal{E}(q)$ while also satisfying $p = p_\mathrm{r}$ is given by

$$\mathcal{A} = \{x \in \mathbb{X} : \ \|q_{ij}\| = d \ \forall (i,j) \in \mathcal{E}(q), \ p = p_\mathrm{r}\}. \tag{3.1}$$

This set is also referred to as the set of all admissible α-lattice configurations. The general CMPF problem can then be stated in the form of Problem 2.1 with stage cost

$$\ell(x_{k+m|k}, u_{k+m|k}) = \|u_{k+m|k}\|_{R_u}^2 + \|p_{k+m|k} - p_\mathrm{r}\|_{R_p}^2 + \sum_{(i,j)\in\mathcal{E}(q_k)} \|\delta_{ij,k+m|k}\|_{R_\delta}^2. \tag{3.2}$$

Here, $R_u \succ 0$ is a positive definite weighting matrix, and $R_p, R_\delta \succeq 0$ are positive semi-definite weighting matrices. For the stage cost (3.2), the following assumption is introduced:

Assumption 3.1. The communication topology for the optimization problem at time k is considered to be constant over the prediction horizon with

$$\mathcal{G}\big(\mathcal{V}, \mathcal{E}(q_{k+m|k})\big) = \mathcal{G}\big(\mathcal{V}, \mathcal{E}(q_k)\big), \quad \forall m \in \mathbb{I}_{0:M}.$$

This assumption, although rarely stated explicitly, is made implicitly in most of the literature on MPF. Without it, the MPF problem would have to be solved iteratively since the problem formulation would depend on the solution of the problem.

In most of the frameworks in the MPF literature, no state constraints are considered,

$$\mathcal{X} = \mathbb{R}^{Mn_x}, \tag{3.3}$$

and input constraints are expressed in the form

$$\mathcal{U} = \left\{ U \in \mathbb{R}^{Mn_u} : \ \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes I_{Mn_u} U - \begin{bmatrix} u_\mathrm{max} \\ -u_\mathrm{min} \end{bmatrix} \otimes 1_{Mn_u} \leq 0 \right\}, \tag{3.4}$$

with maximum and minimum admissible inputs u_max and u_min, respectively.

3.1.2 Distributed Model Predictive Flocking

Similar to the CMPF problem, the general DMPF problem can be stated in the form of Problem 2.3 with local stage cost

$$\ell_i(x_{i,k+m|k}, x_{\bar{i},k+m|k}, u_{i,k+m|k}) = \|u_{i,k+m|k}\|_{R_u}^2 + \|p_{i,k+m|k} - p_{\mathrm{r}_i}\|_{R_p}^2 + \sum_{j\in\mathcal{N}_i(q_k)} \|\delta_{ij,k+m|k}\|_{R_\delta}^2, \tag{3.5}$$

where $p_{\mathrm{r}_i} \in \mathbb{R}^{n_\mathrm{d}}$ is the reference velocity for a single agent. Note that Assumption 3.1 is made again. For the DMPF problem, the constraint sets corresponding to (3.3) and (3.4) are given by

$$\mathcal{X}_i = \mathbb{R}^{Mn_{x,i}}, \tag{3.6}$$

$$\mathcal{U}_i = \left\{ U_i \in \mathbb{R}^{Mn_{u,i}} : \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes I_{Mn_{u,i}} U_i - \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix} \otimes 1_{Mn_{u,i}} \leq 0 \right\}. \tag{3.7}$$

Similar to (3.1), the set of all desired configurations for agent i is given by

$$\mathcal{A}_i(x_{\bar{i}}) = \{ x_i \in \mathbb{X}_i : \ \|q_{ij}\| = d \ \forall (i,j) \in \mathcal{E}(q), \ p_i = p_{\mathrm{r}_i} \}. \tag{3.8}$$

Given how the objective of flocking control is formulated, each agent's stage cost also depends on the predicted future states of its neighbors. However, depending on how the optimization algorithm is set up, the exact future states of the neighboring agents may be unknown. In that case, the neighbor states $x_{\bar{i},k+m|k}$ must be estimated. This estimation, based on information available at time k, is denoted as $\hat{x}_{\bar{i},k+m|k}$.

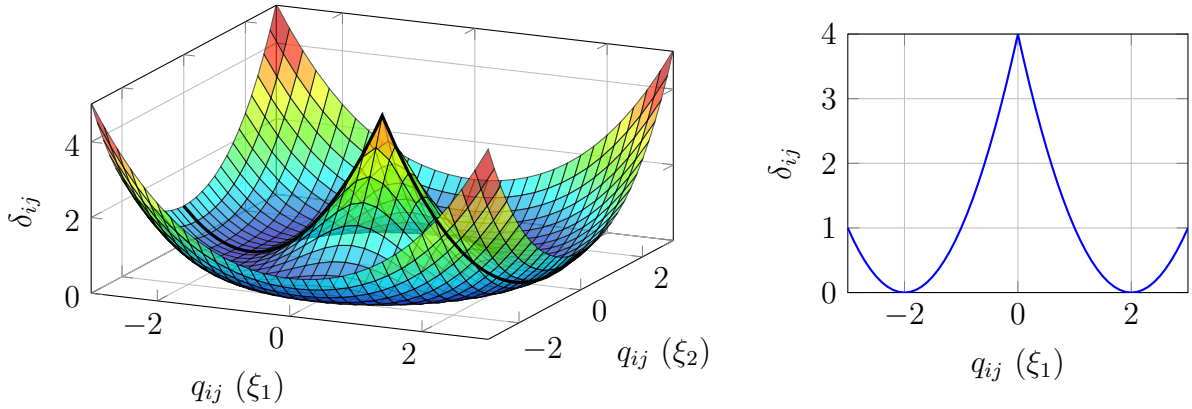
3.2 Challenges in the Stability Analysis of Model Predictive Flocking

Due to the nature of the desired flocking configuration and the assumptions made in the MPF problem formulation, analyzing the stability of the presented MPF schemes can be a challenging task. Three of the prevalent challenges for establishing stability of CMPF and DMPF schemes are the non-convex nature of the cost function (Section 3.2.1), the unknown setpoint (Section 3.2.2), and the time-varying communication topology (Section 3.2.3). In the following, these challenges and their effect on the stability analysis are discussed in detail.

3.2.1 Non-Convex and not Continuously Differentiable Cost Functions

In the CMPF and DMPF cost functions in (3.2) and (3.5), the objective of forming an α-lattice is included in the optimization problem by penalizing $\delta(q_i, q_j)$, the deviations from the desired configuration. However, by definition, this function, and therefore the MPF cost function, are not convex since $\delta(q_i, q_j)$ is non-convex in the agents' positions (for information on convex sets and functions see Luenberger and Ye (2016)). This is visualized in Fig. 3.1, where the squared deviations are plotted for two agents ($n_\mathrm{d} = 2$) over the inter-agent distances in ξ_1- and ξ_2-directions (Fig. 3.1a). Recall that ξ_1 and ξ_2 denote the coordinates in Cartesian coordinate systems. Figure 3.1b furthermore provides a projection with q_{ij} in ξ_2-direction fixed to zero (indicated by the thick black line in Fig. 3.1a).

For convex optimization problems, every minimum is a global minimum. Therefore, this class of problems can generally be solved numerically very efficiently (Boyd and



(a) Squared deviations plotted over q_{ij} in ξ_1- and ξ_2-direction.

(b) Projection of δ_{ij}^2 for q_{ij} in ξ_2-direction fixed to zero.

Figure 3.1: Squared deviation from desired separation plotted over the inter-agent distance q_{ij} for two agents with $d = 2$.

Vandenberghe 2004). On the other hand, this is normally not true for nonlinear programs. In general, there are no effective methods for solving this class of problem but there exist several different approaches, each of which involves some compromise (Boyd and Vandenberghe 2004). For more information on this topic, the reader is referred to Boyd and Vandenberghe (2004) and Rawlings et al. (2020, Chapter 8).

In practice, this challenge is commonly addressed by approximating the original problem by a convex one, for example by replacing δ_{ij} by a convex approximation $\hat{\delta}_{ij}$. However, the error introduced by this approximation then has to be accounted for in the analysis, for example by treating the mismatch as an uncertainty and applying robust control techniques (Cannon et al. 2011).

One common convex approximation of δ_{ij} used in the MPF literature (cf. Huang et al. 2019; Zhan and Li 2011a,b, 2013; Zhang et al. 2015; Zhou and Li 2017) is obtained by first rewriting the normed deviations from the desired separation as

$$\|\delta_{ij}\| = \|\|q_{ij}\| - d\| = \|\|q_{ij}\| - d\| \left\| \frac{q_{ij}}{\|q_{ij}\|} \right\| = \left\| q_{ij} - d\frac{q_{ij}}{\|q_{ij}\|} \right\|. \tag{3.9}$$

Then, the second term is approximated based on an estimation $\hat{q}_{ij}$,

$$\|\hat{\delta}_{ij}\| = \left\| q_{ij} - d\frac{\hat{q}_{ij}}{\|\hat{q}_{ij}\|} \right\|. \tag{3.10}$$

For a given $\hat{q}_{ij}$, (3.10) is then convex in q_{ij}.

In addition to the cost function not being convex, it is also not continuously differentiable at $q_{ij} = 0$, which corresponds to a collision between agents i and j. The application of optimality conditions and many of the commonly used optimization algorithms require that the cost and constraint functions are twice continuously differentiable (Luenberger and Ye 2016). Also note that the approximation in (3.10) is not even defined at $q_{ij} = 0$.

One way of resolving this problem is by replacing the norm-function in the definition of δ_{ij} by a continuously differentiable function, for example the σ-map in (2.24), yielding the approximation $\hat{\delta}_{ij} = \sigma_s(q_{ij}) - d$. An alternative approach to resolving this issue is to simply assume that no inter-agent collisions occur, removing the critical point $q_{ij} = 0$ from the considered state space.

3.2.2 Unknown Setpoint

In the line of reasoning for stability analysis presented in Section 2.4.2, the terminal ingredients are used to steer the states towards the desired equilibrium. However, in MPF, the setpoint the swarm converges to is not known a priori. Rather the agents will converge to the set of equilibria $\mathcal{A}$ in (3.1), the set of all states satisfying the flocking constraints. This complicates the design of the terminal ingredients as they then have to be designed with respect to all equilibria contained in $\mathcal{A}$ and not only a single point. Additionally, when considering Assumption 3.1, the set of equilibria then depends on the state at time k since the communication topology is considered constant over the prediction horizon. Mathematically, based on Assumption 3.1, the set $\mathcal{A}$ in (3.1) can then be rewritten as

$$\mathcal{A}(x_k) = \{x \in \mathbb{X} : \ \|q_{ij}\| = d \ \forall (i,j) \in \mathcal{E}(q_k), \ \|q_{ij}\| \geq r \ \forall (i,j) \notin \mathcal{E}(q_k), \ p = p_{\mathrm{r}}\}. \quad (3.11)$$

In order to avoid having to consider the set of all equilibria in the analysis, one common approach is to assume that the agents converge towards the α-lattice configuration closest to their initial state x_0 (Zhan and Li 2013; Zhang et al. 2015; Zhou and Li 2017). In the centralized case, this state $x_\alpha^*(x_0)$ can be computed by solving the optimization problem

$$\begin{aligned} x_\alpha^*(x_0) = \arg\min_{x \in \mathbb{X}} & \|x - y\| \\ \text{s.t. } & y \in \mathcal{A}(x_0). \end{aligned} \quad (3.12)$$

However, this point cannot be computed straightforwardly in a distributed fashion as the definition of $\mathcal{A}(x)$ in (3.11) requires knowledge of the global position vector q_0 and the constraints in (3.12) are nonlinear and coupled with respect to the agents' states (Yang et al. 2019). In general, this information is not available in distributed schemes.

In summary, a priori unknown setpoint in MPF can complicate the design of stabilizing terminal ingredients.

3.2.3 Time-Varying Communication Topology

In MPF, agent interactions are modeled via a state-dependent proximity graph $\mathcal{G}(\mathcal{V}, \mathcal{E}(q(k)))$. By definition, this graph is not necessarily static and consequently, an agent's neighbor set can change between sampling instances. Together with Assumption 3.1, these changes in the communication topology can break the line of reasoning in the stability analysis if not addressed explicitly. The nature of this issue can be demonstrated when considering the general CMPF-problem in the form of Problem 2.1 with stage cost (3.2), constraints

(3.3) and (3.4), and terminal ingredients $V_f = 0$, $\kappa_M = 0$, and $\mathbb{X}_f = \mathcal{A}$. Without loss of generality, consider $p_\mathrm{r} = 0$ for this example. With $U^*(k)$ being the optimal input at time k, and following the line of reasoning presented in Section 2.4.2, the warm start in (2.59) should result in a cost decrease at time $k+1$, yielding

$$\begin{aligned} 0 &\geq V(x_{k+1}, \tilde{U}_{k+1|k}) - V(x_k, U_k^*) \\ &= -\ell(x_k, u_{k|k}^*) + \sum_{m=0}^{M-2} \big(\ell(x_{k+m+1|k+1}, u_{k+m+1|k}^*) - \ell(x_{k+m+1|k}^*, u_{k+m+1|k}^*)\big) \\ &\qquad\qquad + \ell(x_{k+M|k+1}, 0) \\ &= -\ell(x_k, u_{k|k}^*) + \sum_{m=0}^{M-2} \Big(\sum_{(i,j)\in\mathcal{E}(q_{k+1})} \|\delta_{ij,k+m+1|k+1}\|_{R_\delta}^2 - \sum_{(i,j)\in\mathcal{E}(q_k)} \|\delta_{ij,k+m+1|k}^*\|_{R_\delta}^2 \Big) \\ &\qquad\qquad + \sum_{(i,j)\in\mathcal{E}(q_{k+1})} \|\delta_{ij,k+M|k+1}\|_{R_\delta}^2. \end{aligned} \tag{3.13}$$

By definition of $\tilde{U}(k+1|k)$ according to (2.59), $x(k+m+1|k+1) = x^*(k+m+1|k)$ holds. For static graphs ($\mathcal{E}(q_{k+1}) = \mathcal{E}(q_k)$), it holds that $\mathcal{A}(x_k) = \mathcal{A}(x_{k+1})$ and consequently $\ell(x_{k+M|k+1}, 0) = 0$. Equation (3.13) then simplifies to

$$0 \geq V(x_{k+1}, \tilde{U}_{k+1|k}) - V(x_k, U_k^*) = -\ell(x_k, u_{k|k}^*),$$

which is satisfied by the definition of $\ell(x_k, u_k)$.

Next, consider the scenario of a dynamic communication topology ($\mathcal{E}(q_{k+1}) \neq \mathcal{E}(q_k)$). In this case, (3.13) can be rewritten as

$$\begin{aligned} 0 \geq V(x_{k+1}, \tilde{U}_{k+1|k}) - V(x_k, U_k^*) = &- \ell(x_k, u_k) \\ &- \sum_{m=0}^{M-1} \sum_{\substack{(i,j)\in\mathcal{E}(q_k)\setminus \\ \mathcal{E}(q_{k+1})}} \|\delta_{ij,k+m+1|k}^*\|_{R_\delta}^2 \\ &+ \sum_{m=0}^{M-1} \sum_{\substack{(i,j)\in\mathcal{E}(q_{k+1})\setminus \\ \mathcal{E}(q_k)}} \|\delta_{ij,k+m+1|k+1}^*\|_{R_\delta}^2 \\ &+ \ell(x_{k+M|k+1}, 0), \end{aligned}$$

where the negative terms in the second line correspond to the edges that are removed during the transition from $\mathcal{E}(q_k)$ to $\mathcal{E}(q_{k+1})$, and the positive terms in the third line correspond to the edges added. This demonstrates that additional edges in the graph can result in the cost no longer decreasing due to the additional positive terms. Furthermore, the warm start $\tilde{U}(k+1|k)$ may in this scenario no longer be a feasible input sequence at time $k+1$ because the terminal equality constraint is not necessarily satisfied for those edges that were not included in the optimization problem at time k, i.e.

$$\delta_{ij,k+M|k}^* = 0, \ \forall (i,j) \in \mathcal{E}(q_{k+1}) \backslash \mathcal{E}(q_k)$$

does not hold necessarily, resulting in $\ell(x_{k+M|k+1}, 0) > 0$.

In conclusion, if not considered carefully, the time-varying communication topology has the potential to break the line of reasoning in the stability analysis. Methods for resolving these issues would involve a careful design of the stage cost and terminal ingredients to ensure the cost decrease and recursive feasibility in scenarios with changing communication topologies.

3.2.4 Prediction Mismatch in Distributed Optimization

Another challenge in analyzing distributed model predictive control is the prediction mismatch between agents. The way the DMPC and DMPF problems were introduced in Section 2.4.1 and 3.1.2, they allow for a coupling between agents in cost and constraints. However, when agents are solving their optimization problems in parallel, the control decisions of neighboring agents are not known exactly, requiring an estimation or approximation of this information. Approaches for computing $\hat{x}_{\bar{i}}$ include considering $\hat{x}_{\bar{i}}$ as additional optimization variables in the MPF problem (Huang et al. 2019) or estimating it based on assumptions on the future behavior of the neighboring agents (Hu et al. (2018, 2017), Zhan and Li (2011b), and Zhang et al. (2015)). The latter is the most common approach in the MPF literature. It is assumed that, at time k, agents communicate their current state $x_i(k)$ to their neighbors. Each agent then estimates the future states of its neighbors under the assumption that agents are moving with a constant velocity (Zhan and Li 2011b; Zhang et al. 2015),

$$\hat{x}_j(k+m|k) = \phi_j(m; x_j(k), 0), \quad \forall j \in \mathcal{N}_i(k),\, m \in \mathbb{I}_{0:M}, \tag{3.14}$$

assuming an equilibrium input $u_i^{\mathrm{e}} = 0$. Alternatively, agents can additionally exchange the last calculated optimal input sequence $U_i^*(k-1)$ (Hu et al. 2018, 2017). By constructing an estimated input sequence similar to the warm start in (2.59),

$$\hat{U}_j(k|k-1) = \operatorname{col}\big(u_j^*(k|k-1), \dots, u_j^*(k+M-2|k-1), 0\big), \quad \forall j \in \mathcal{N}_i(k), \tag{3.15}$$

agents can then compute the estimated neighbor states according to

$$\hat{x}_j(k+m|k) = \phi_j(m; x_j(k), \hat{U}_j(k|k-1)), \quad \forall j \in \mathcal{N}_i(k),\, m \in \mathbb{I}_{0:M}. \tag{3.16}$$

However, when using one of the above estimations, there is a mismatch between the estimated and the true neighbor states. This mismatch, also referred to as prediction mismatch, is a well-known challenge in the field of distributed MPC to cause problems in the stability analysis (Kloock and Werner 2020; Shamma 2008). The following example will demonstrate how the prediction mismatch can break the line of reasoning in the stability analysis when not addressed properly.

Consider the general DMPF-problem in the form of Problem 2.3 with stage cost (3.5), $p_{\mathrm{r}} = 0$, constraint sets (3.6) and (3.7), and terminal equality constraints, $V_{i,f} = 0$, $\kappa_{i,M} = 0$, $\mathbb{X}_{i,f} = \mathcal{A}_i(x_{\bar{i}})$. Furthermore, assume that the communication graph is static and that the

estimation $\hat{x}_{\bar{i}}$ is computed based on communicated data according to (3.15) and (3.16). With $U_i^*(k)$ being the optimal input at time k and following the line of reasoning presented in Section 2.4.2, the local warm start in (2.59) should result in a cost decrease at time $k+1$, yielding

$$
\begin{aligned}
0 &\geq V_i(x_{i,k+1}, \hat{X}_{\bar{i},k+m|k+1}, \tilde{U}_{i,k+1|k}) - V_i(x_{i,k}, \hat{X}_{\bar{i},k+m|k}, U_{i,k}^*) \qquad (3.17)\\
&= \sum_{m=1}^{M-1} \big(\ell_i(x_{i,k+m|k+1}, \hat{x}_{\bar{i},k+m|k+1}, u_{i,k+m|k}^*) - \ell_i(x_{i,k+m|k}^*, \hat{x}_{\bar{i},k+m|k}, u_{i,k+m|k}^*)\big)\\
&\qquad -\ell_i(x_{i,k}, \hat{x}_{\bar{i},k|k}, u_{i,k|k}^*) + \ell_i(x_{i,k+M|k+1}, \hat{x}_{\bar{i},k+M|k+1}, 0)\\
&= \sum_{m=1}^{M-1} \Big(\sum_{j\in\mathcal{N}_{i,k+1}} \|\delta(q_{i,k+m|k+1}, \hat{q}_{j,k+m|k+1})\|_{R_\delta}^2 - \sum_{j\in\mathcal{N}_{i,k}} \|\delta(q_{i,k+m|k}^*, \hat{q}_{j,k+m|k})\|_{R_\delta}^2 \Big)\\
&\qquad -\ell_i(x_{i,k}, \hat{x}_{\bar{i},k|k}, u_{i,k|k}^*) + \sum_{j\in\mathcal{N}_{i,k+1}} \|\delta(q_{i,k+M|k+1}, \hat{q}_{j,k+M|k+1})\|_{R_\delta}^2.
\end{aligned}
$$

As the information from which $\hat{X}_{\bar{i}}$ is calculated can change between sampling instances, $\hat{q}_{j,k+m|k}$ and $\hat{q}_{j,k+m|k+1}$) are not necessarily the same. Consequently, the δ-terms in (3.17) no longer sum up to zero. If the resulting mismatch is larger than the combined negative terms, (3.17) can become positive, breaking the line of reasoning in the stability analysis. For the same reason, recursive feasibility can no longer be guaranteed. Since $U_i^*(k)$ ensures $\delta(q_{i,k+M|k}, \hat{q}_{j,k+M|k}) = 0$ but not $\delta(q_{i,k+M|k+1}, \hat{q}_{j,k+M|k+1}) = 0$, the terminal equality constraint is not necessarily satisfied by the warm start.

This discussion demonstrates that, if not considered explicitly, the prediction mismatch has the potential to break the stability analysis. The same arguments can also be made for terminal ingredients other than the terminal equality constraints.

Apart from considering a centralized scheme, an alternative approach to addressing the prediction mismatch is to consider optimization algorithms where agents do not solve their optimization problems in parallel but instead sequentially. Such a scheme is presented in Müller et al. (2012), where agents negotiate with their neighbors to ensure that (3.17) is satisfied. If this is not the case, the terminal control law is applied. However, one limitation of this approach is that it requires a centralized entity to compute the terminal control law in scenarios where the terminal ingredients are coupled, which is the case in MPF.

An alternative approach for addressing the prediction mismatch is to impose constraints on the extent to which an agent can deviate from its trajectory computed at the previous time step to obtain a bound on the prediction mismatch (Keviczky et al. 2008). This is the approach employed in Hu et al. (2017) and Hu et al. (2018). However, as discussed in detail in Section 3.4, the application to the MPF analysis again presents challenges due to the couplings in the terminal set, cost, and controller.

3.3 On the Analysis of Model Predictive Flocking Using N-Paths

The following section is dedicated to one of the approaches used in the analysis of MPF based on geometric properties of the optimal state sequence, so-called N-paths. The discussion presented in this section follows the results presented in Hastedt and Werner (2023c). Originally, N-paths were introduced in Ferrari-Trecate et al. (2009) to analyze the stability of a DMPC consensus problem for agents with single-integrator dynamics without requiring terminal ingredients. In Zhan and Li (2013), the concept of N-paths was then applied to CMPF and sequential DMPF for agents with single-integrator dynamics. Later, these results were extended to DMPF with parallel optimization for agents with double-integrator dynamics (Zhang et al. 2015; Zhou and Li 2017). As the above-mentioned analyses of MPF based on N-paths do not require the design of terminal sets and terminal controllers, this line of reasoning appears to be an elegant solution to the complicated analysis of nonlinear and distributed MPC. However, a detailed analysis of the arguments presented in Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017) reveals several errors in the lines of reasoning. While some of these issues can be addressed by imposing additional constraints, other inherent problems cannot be resolved, breaking the line of reasoning in the stability analysis.

The remainder of this section is structured as follows: First, the concept of N-paths is introduced and the general reasoning behind the N-path-based analysis is presented. Afterwards, the approaches in the literature are discussed with respect to the general challenges in the MPF analysis, as introduced in Section 3.2, and problems specific to the N-path arguments.

3.3.1 N-Paths

The concept of N-paths was introduced in Ferrari-Trecate et al. (2009) to show stability for a distributed MPC consensus problem with single-integrator agents. In the following, the main definitions and theorems regarding N-paths are presented.

An N-path $T_A = \{A_1, A_2, \dots, A_N\} \subset \mathbb{R}^n$ is an ordered sequence of N points. Given two points $A_i, A_j \in T_A$, let $\overline{A_iA_j}$ be the segment joining them. Furthermore, let $|\overline{A_iA_j}|$ denote the length of that segment.

Definition 3.1 (Ferrari-Trecate et al. (2009, Definition 7))**.** An N-path T_A is said to be non-increasing with respect to a point $O \in \mathbb{R}^n$ if $|\overline{A_{j+1}O}| \leq |\overline{A_jO}|$, $\forall j \in \{1, \dots, N-1\}$.

Definition 3.2 (Ferrari-Trecate et al. (2009, Definition 8))**.** An N-path is said to be pointing towards $O \in \mathbb{R}^n$ if it is non-increasing with respect to O and $A_j \in \overline{A_1O}$, $\forall j \in \{1, \dots, N\}$.

With these definitions, the following theorem can be stated:

Theorem 3.1 (Ferrari-Trecate et al. (2009, Theorem 5)). *Let $T_A = \{A_1, \dots, A_N\} \in \mathbb{R}^n$ be an N-path. Given $O \in \mathbb{R}^n$, there always exists an N-path $T_B = \{B_1, \dots, B_N\} \in \mathbb{R}^n$ with $B_1 = A_1$, T_B pointing towards O, and T_B satisfying the following inequalities:*

$$|\overline{B_jO}| \leq |\overline{A_jO}|, \quad j = 1, \dots, N, \tag{3.18}$$

$$|\overline{B_jB_{j+1}}| \leq |\overline{A_jA_{j+1}}|, \quad j = 1, \dots, (N-1). \tag{3.19}$$

The proof is provided in Ferrari-Trecate et al. (2009). A visual representation of this result is provided in Fig. 3.2, where it is straightforward to see that $T_A \subset \mathbb{R}^2$ and T_B satisfy Theorem 3.1.

As pointed out in Ferrari-Trecate et al. (2009), the N-path analysis is only applicable to agents governed by single-integrator dynamics. To apply this concept to agents governed by double-integrator dynamics, in Zhang et al. (2015) the authors propose an extension of Theorem 3.1.

Lemma 3.2 (Zhang et al. (2015, Theorem 5)). *Let $T_A = \{A_1, \dots, A_N\} \in \mathbb{R}^n$ be an N-path. Given $O \in \mathbb{R}^n$, there always exists an N-path $T_B = \{B_1, \dots, B_N\} \in \mathbb{R}^n$ with $B_1 = A_1$, T_B pointing towards O, and T_B satisfying the following inequalities:*

$$|\overline{B_jO}| \leq |\overline{A_jO}|, \quad j = 1, \dots, N, \tag{3.20}$$

$$|\overline{B_jB_{j+1}}| \leq |\overline{A_jA_{j+1}}|, \quad j = 1, \dots, (N-1), \tag{3.21}$$

$$|\overline{B_{j+1}B_{j+2}} - \overline{B_jB_{j+1}}| \leq |\overline{A_{j+1}A_{j+2}} - \overline{A_jA_{j+1}}|, \quad j = 1, \dots, (N-2), \tag{3.22}$$

with

$$\overline{B_{N-1}B_N} = \overline{B_{N-2}B_{N-1}}. \tag{3.23}$$

Note that Lemma 3.2 only differs from Theorem 3.1 by the additional conditions in (3.22) and (3.23). This extension is one of the main reasons why the analysis provided in Zhang et al. (2015) and Zhou and Li (2017) does not hold as stated, since it can be shown that Lemma 3.2 does not hold. An in-depth discussion of this issue is provided in Section 3.3.3.

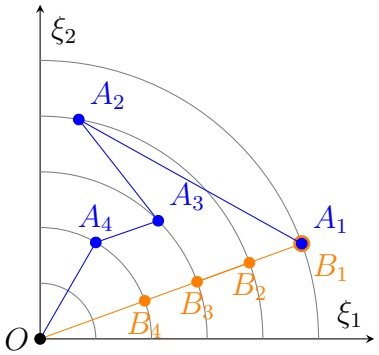


Figure 3.2: Visualization of Theorem 3.1.

3.3.2 Model Predictive Flocking Problem Statement and N-Path-Based Analysis

In the literature on MPF using N-paths, the normed and summed deviations from the desired configuration are commonly written as

$$\|g(q)\|^{\mu} = \sum_{(i,j)\in\mathcal{E}(q)} \|\delta(q_i, q_j)\|^{\mu} \tag{3.24}$$

for the centralized case and

$$\|g_i(\bar{q}_i)\|^{\mu} = \|g_i(q_i, q_{\bar{i}})\|^{\mu} = \sum_{j\in\mathcal{N}_i(q)} \|\delta(q_i, q_j)\|^{\mu}, \tag{3.25}$$

for the decentralized case, with $\mu \in \mathbb{I}_{1:2}$.

Problem Statements

In Zhan and Li (2013), with the authors considering agents governed by the discrete-time single-integrator dynamics in (2.9), the CMPF problem can be stated as follows:

Problem 3.1 (CMPF problem for N-path analysis)**.**

$$\begin{aligned} V^*(x(k)) &= \min_{U(k)} V(x(k), U(k)) \\ &= \min_{U(k)} \sum_{m=0}^{M-1} \left(\|g(q(k+m+1|k))\|^2 + r_u \|u(k+m|k)\|^2 \right) \\ &\text{s.t. } x(k+m) = \phi(m; x(k), U(k)), \quad m \in \mathbb{I}_{0:M}. \end{aligned}$$

Note that this problem does not consider any constraints on the states or inputs and the only tuning parameter is the scalar input weight r_u.

Analogously, in the distributed schemes considering the discrete-time double-integrator agents in (2.10), the DMPF problems can be generalized as follows:

Problem 3.2 (DMPF problem for N-path analysis)**.**

$$\begin{aligned} V_i^*(x_i(k), X_{\bar{i}}^M(k)) &= \min_{U_i(k)} V(x_i(k), X_{\bar{i}}^M(k), U_i(k)) \\ &= \min_{U_i(k)} \sum_{m=0}^{M-1} \left(\|g_i(q_i(k+m+1|k), q_{\bar{i}}(k+m+1|k))\|^2 + r_u \|u_i(k+m|k)\|^2 \right) \\ &\text{s.t. } x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ &\quad U_i(k) \in \mathcal{U}_i. \end{aligned}$$

In the MPF scheme presented by Zhang et al. (2015), input constraints in the form of (3.7) are considered. On the other hand, the approach presented in Zhou and Li (2017) does

not impose input constraints ($\mathcal{U}_i = \mathbb{R}^{Mn_{u,i}}$). While Zhang et al. (2015) explicitly state that $X_{\bar{i}}(k)$ is estimated according to (3.14), assuming the neighbors to move with constant velocities, in Zhou and Li (2017), it is only mentioned that the states of the neighboring agents are computed based on information up to time $k-1$ without explaining how this estimation is obtained. Moreover, the cost function of the distributed optimization problem lacks a term that penalizes deviations from a desired velocity. Hence, for Problem 3.2, in contrast to (3.11), the set of states resulting in a zero cost is given by

$$\mathcal{A}(x_k) = \{x \in \mathbb{R}^{n_x} : \ \|q_{ij}\| = d \ \forall (i,j) \in \mathcal{E}(q), \ \|q_{ij}\| \geq r \ \forall (i,j) \notin \mathcal{E}(q_k)\}. \tag{3.26}$$

Remark. In Zhou and Li (2017), the definition of the neighbor set is modified slightly by not considering those agents whose movement is already driving them towards the desired distance. This is referred to as neighbor screening. The remaining problem formulation and line of reasoning is the same as in Zhang et al. (2015). As this redefinition of the neighbor set does not affect the following arguments, no distinction is made notation-wise between the standard neighbor set and the one with neighbor screening in this section.

Stability Analysis

For the stability analysis, all the aforementioned publications are based on the following two assumptions.

Assumption 3.2. There exists a $\kappa \geq 0$ such that the MAS is connected across the interval $[k, k+\kappa]$ for all $k > 0$.

Assumption 3.3. Given a collective initial position $q_0 \in \mathbb{R}^{Nn_\mathrm{d}}$, there exists a nearest desired α-lattice state $q_\alpha^* \in \mathbb{R}^{Nn_\mathrm{d}}$. Furthermore, it then holds for all $q', q'' \in \mathbb{R}^{Nn_\mathrm{d}}$ that

$$\|q' - q_\alpha^*\| \leq \|q'' - q_\alpha^*\| \Rightarrow \|g(q')\| \leq \|g(q'')\|. \tag{3.27}$$

Note that in Assumption 3.3, the closest desired α-lattice can be computed as

$$q_\alpha^*(x) = \arg \min_{y \in \mathbb{R}^{Nn_\mathrm{d}}} \ |y|_{\mathcal{A}(x)}. \tag{3.28}$$

Assuming that Assumption 3.2 and 3.3 hold, Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017) then state theorems about their respective schemes converging towards the desired α-lattice configuration. For the exact statements, the reader is referred to the cited references. In the following, the general line of reasoning of the analysis in these papers is outlined.

The key argument in the analysis of model predictive flocking schemes using N-paths is the claim that the trajectory resulting from the solution of the optimization problem can be considered an N-path pointing towards the closest α-lattice q_α^* in the sense of Definition 3.2. In the distributed schemes, q_α^* is replaced with $\bar{q}_{i,\alpha}^*$, where $\bar{q}_{i,\alpha}^*$ is the closest desired α-lattice for the i^{th} subsystem extracted from q_α^* (Zhang et al. 2015; Zhou and Li 2017). The analysis then proceeds as follows:

Consider any sequence of positions $T_{Q_k} = \{q_{k+1} \dots, q_{k+M}\}$ with $O = q^*_\alpha$ and the nearest desired α-lattice state q^*_α. Then, according to Theorem 3.1 (single-integrator dynamics) or Lemma 3.2 (double-integrator dynamics), for $m \in \mathbb{I}_{1:M}$, there always exists an N-path of length M, denoted as $T^*_{Q_k}$, that is pointing towards q^*_α, with $\|q^*_{k+m} - q^*_\alpha\| \leq \|q_{k+m} - q^*_\alpha\|$. Assuming that Assumption 3.3 holds, this then yields

$$\|g(q^*_{k+m})\| \leq \|g(q_{k+m})\|, \quad m \in \mathbb{I}_{1:M}. \tag{3.29}$$

The same argument can also be made in the distributed case, resulting in

$$\|g_i(\bar{q}^*_{i,k+m})\| \leq \|g_i(\bar{q}_{i,k+m})\|, \quad m \in \mathbb{I}_{1:M}. \tag{3.30}$$

Rewriting the discrete-time dynamics in (2.9) and (2.10), the inputs at time k can be written as

$$\begin{cases} u_k = (q_{k+1} - q_k)/\tau & \text{(collective dynamics)} \\ u_{i,k} = (q_{ik+1} - q_{i,k})/\tau & \text{(single-agent dynamics)} \end{cases} \tag{3.31}$$

for single-integrator agents and as

$$\begin{cases} u_k = \big((q_{k+2} - q_{k+1}) - (q_{k+1} - q_k)\big)/\tau^2 & \text{(collective dynamics)} \\ u_{i,k} = \big((q_{i,k+2} - q_{i,k+1}) - (q_{i,k+1} - q_{i,k})\big)/\tau^2 & \text{(single-agent dynamics)} \end{cases} \tag{3.32}$$

for double-integrator agents. Then, by (3.19) in Theorem 3.1 or (3.22) in Lemma 3.2, it follows that

$$\|u^*_{k+m}\| \leq \|u_{k+m}\|, \qquad m \in \mathbb{I}_{1:M-1}, \tag{3.33}$$

and

$$\|u^*_{i,k+m}\| \leq \|u_{i,k+m}\|, \qquad m \in \mathbb{I}_{1:M-1}, \tag{3.34}$$

respectively. For the centralized case, considering the construction of the cost function of Problem 3.1, with (3.29) and (3.33) it follows that the optimal solution must be an N-path pointing towards q^*_α, resulting in

$$V(x_k, U^*_k) \leq V(x_k, U_k). \tag{3.35}$$

Otherwise, there would exist another solution to the respective problems yielding a lower cost. Similarly, for the decentralized case, with Problem 3.2, (3.30) and (3.34) lead to the conclusion that the optimal solution must be an N-path pointing towards $\bar{q}^*_{i,\alpha}$, and therefore

$$V_i(x_{i,k}, X^M_{\bar{i},k}, U^*_{i,k}) \leq V_i(x_{i,k}, X^M_{\bar{i},k}, U_{i,k}) \tag{3.36}$$

Since the optimal position sequences are pointing towards q^*_α, or $\bar{q}^*_{i,\alpha}$ in the decentralized case, it follows from Assumption 3.3 that

$$\|q^*_{k+m+1} - q^*_\alpha\| \leq \|q^*_{k+m} - q^*_\alpha\| \Rightarrow \|g(q^*_{k+m+1})\| \leq \|g(q^*_{k+m})\|, \quad m \in \mathbb{I}_{0:M-1}, \tag{3.37}$$

and

$$\|\bar{q}^*_{i,k+m+1} - \bar{q}^*_{i,\alpha}\| \leq \|\bar{q}^*_{i,k+m} - \bar{q}^*_{i,\alpha}\| \Rightarrow \|g_i(\bar{q}^*_{i,k+m+1})\| \leq \|g_i(\bar{q}^*_{i,k+m})\|, \; m \in \mathbb{I}_{0:M-1}, \tag{3.38}$$

respectively.

Based on the warm start as defined in (2.59) and (3.37), Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017) then argue that

$$V(x_{k+1}, \tilde{U}_{k+1|k}) - V(x_k, U_k^*) \leq 0 \tag{3.39}$$

and

$$V_i(x_{i,k+1}, X_{\bar{i},k+1}^M, \tilde{U}_{i,k+1|k}) - V_i(x_{i,k}, X_{\bar{i},k}^M, U_{i,k}^*) \leq 0 \tag{3.40}$$

hold, demonstrating that the value functions are Lyapunov functions. Since no state constraints are considered in the discussed schemes, recursive feasibility is also guaranteed if the optimization problems are feasible for $k = 0$.

3.3.3 Critical Discussion

While the authors in Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017) claim to prove that the proposed schemes converge towards the desired α-lattice-configuration by using the concept of N-paths, a careful analysis of the presented lines of reasoning in these references reveals several mistakes, resulting in the proofs of the stability theorems being incorrect. Some of these mistakes can be traced back to the general challenges in the MPF analysis as presented in Section 3.2. Others are specific to the N-paths arguments. In the following, the errors in the lines of reasoning in Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017) will be discussed in detail.

General MPF Challenges

In Zhan and Li (2013), Zhang et al. (2015), and Zhou and Li (2017), the authors only assume connectedness across an interval, allowing for time-varying communication topologies (Assumption 3.2). However, in none of the cited references, changes in the communication graph are addressed explicitly, resulting in the problem discussed in Section 3.2.3. Consequently, the presented analysis results do not hold in general, but only under the additional constraint that no edges are added to the interaction graph. Otherwise, (3.39) and (3.40) do not necessarily hold. For the remainder of this discussion it is therefore assumed that the interaction graph is static.

Additionally, in the distributed schemes proposed by Zhang et al. (2015) and Zhou and Li (2017), the prediction mismatch is not considered, which can result in (3.40) not holding, as explained in Section 3.2.4.

Analysis vs Implementation

In all MPF references referring to N-paths, for the implementation, δ_{ij} is replaced by the approximation $\hat{\delta}_{ij}$ defined in (3.10) This allows the authors to formulate the MPF problem as a quadratic optimization problem, avoiding the difficulties described in Section 3.2.1. Consequently, there is a significant difference between the analyzed problem and the one

used for generating the simulation examples. And while all of the aforementioned schemes show good results in simulation, the presented analyses do not hold for the schemes used in simulation scenarios.

Validity of Assumption 3.3

An important relation for the CMPF and DMPF analysis described in Section 3.3.2 is (3.27) in Assumption 3.3, as it establishes the relation between the distance of a configuration from the closest α-lattice and the flocking errors in (3.24) and (3.25). However, one can show that condition (3.27) does not hold for all $q', q'' \in \mathbb{R}^{Nn_\mathrm{d}}$, as stated in Assumption 3.3. This can be demonstrated by constructing a simple 1D-counterexample with $N = 2$ and $d = 2$.

Consider the initial configuration $q_0 = [-1.1\ 1.1]^\top$ with the closest desired α-lattice given by $q_\alpha^* = [-1\ 1]^\top$. With $q' = [-1.1\ 1]^\top$ and $q'' = [0\ 2]^\top$, this results in

$$0.1 = \|q' - q_\alpha^*\| < \|q'' - q_\alpha^*\| = \sqrt{2},$$

but

$$0.1 = \|g(q')\| > \|g(q'')\| = 0.$$

This contradicts the statement of Assumption 3.3. The presented counterexample therefore demonstrates that condition (3.27) in Assumption 3.3 does not hold for all $q', q'' \in \mathbb{R}^{Nn_\mathrm{d}}$ as stated. While the distance from the closest desired α-lattice is an absolute measure from q_α^*, the function $g(q)$ only measures the relative deviations from the desired distance.

With (3.27) not holding in general, (3.29) and (3.30) no longer follow from $T_{Q_k}^*$ pointing towards q_α^*, or $\bar{q}_{i,\alpha}^*$ in the distributed case. Consequently, (3.35) and (3.36) no longer hold as the optimal solution is not necessarily an N-path pointing towards the closest desired α-lattice. This breaks the line of reasoning of the N-path-based analysis.

To address this problem in CMPF, (3.27) could be enforced by constraining the admissible states in Problem 3.1 to a set that satisfies (3.27). For the example given above, one example for such a set would be the set of all configurations with a fixed desired α-lattice $q_\alpha^*(q_0)$, resulting in the constraint set

$$\mathbb{X}(q_\alpha^*, q_0) = \{q \in \mathbb{R}^2 : (q - q_\alpha^*) - \epsilon(q_0 - q_\alpha^*) = 0, \epsilon > -10\}. \tag{3.41}$$

However, since this constraint enforces the same ϵ for every agent, it imposes a highly restrictive constraint on the admissible states of the MAS. For large MASs with $n_\mathrm{d} > 1$, it is furthermore not trivial to compute $\mathbb{X}(q_\alpha^*, q_0)$ explicitly. On the other hand, for DMPF, agents are not aware of the global state q_k. The coupled constraint (3.41) can therefore not be enforced in distributed schemes. Consequently, there is no straightforward solution to address the issue with Assumption 3.3 for DMPF.

Distributed Computation of the Closest Desired Lattice

As outlined in Section 3.3.2, in the analysis of the distributed schemes, the authors of Zhang et al. (2015) and Zhou and Li (2017) claim that the optimal solution is an N-path

pointing towards q_α^*. However, as explained in the context of the general challenges in Section 3.2.2, in distributed schemes, the global setpoint is not known by the agents a priori. Moreover, with access to local information only, each agent converges to a locally optimal α-lattice

$$\mathcal{A}_i(x_{\bar{i}}) = \{x_i \in \mathbb{R}^{|\mathcal{N}_i| n_{x,i}} : \|q_{ij}\| = d \; \forall j \in \mathcal{N}_i\}. \tag{3.42}$$

instead of the global set of α-lattices in (3.26). Due to the different information available to individual agents, the locally optimal α-lattices do not necessarily align. Consider agents i and j to be neighbors. Then $q_{i,\alpha}^{i*}$, the optimal state of agent i calculated by agent i, may not be equal to $q_{i,\alpha}^{j*}$, the optimal state of agent i calculated by agent j, and both of them might be different from $q_{i,\alpha}^*$ in q_α^*. Consequently, for each agent, the obtained optimal position sequence will point towards

$$\bar{q}_{i,\alpha}^{i*} = \arg\min_y \; |y|_{\mathcal{A}_i(x_{\bar{i}})}$$

and not q_α^* in (3.28). With the optimal sequence no longer being an N-path pointing towards $\bar{q}_{i,\alpha}^*$ extracted from q_α^*, the line of reasoning breaks as all of the N-path-based arguments in Section 3.3.2 are no longer valid.

One possible solution for this problem would be to include the distributed computation of q_α^* to the algorithms in Zhang et al. (2015) and Zhou and Li (2017). However, as already outlined in Section 3.2.2, the distributed computation of q_α^* is not straightforward due to the nonlinear, coupled equality constraints in (3.12).

Remark. In Zhan and Li (2013), the authors furthermore propose a sequential DMPF scheme with additional inter-sample iterations. Only one agent is solving its optimization problem at the time and then communicates its control decision to its neighbors before the next agent solves its optimization problem. While this sequential approach can in theory resolve the issue of the prediction mismatch in DMPF, it still is subject to the discussed problems with Assumption 3.3 and the distributed computation of the desired lattice in the context of DMPF schemes. Furthermore, as the authors in Zhan and Li (2013) do not provide a proof for their sequential DMPF stability theorem, this approach is not discussed here in detail.

Double Integrator N-Path Extension

To apply the N-path line of reasoning to agents with double-integrator dynamics, Lemma 3.2 is required to establish (3.33) and (3.34) from (3.32) (Zhang et al. 2015; Zhou and Li 2017). In the original proof of this result, the authors claim that one can always find an N-path satisfying (3.20)-(3.22) (Zhang et al. 2015). However, this statement is not true and can be refuted by the following simple counterexample.

Consider the one-dimensional N-path $T_A = \{4, 1, -2\}$ with $O = 0$. Then, T_B can be constructed according to Lemma 3.2. The resulting paths T_A and T_B are depicted in Fig. 3.3. From the condition that B_1 has to be equal to A_1, it follows that

$$B_1 = A_1 = 4.$$

From (3.20) and (3.21), the only possible choice for B_2 is

$$B_2 = A_2 = 1,$$

as B_2 cannot be further distanced from O than A_2 and $|\overline{B_1B_2}| \leq |\overline{A_1A_2}|$ must hold. Since T_B has to point towards $O = 0$, one must select $B_3 \in [0, 1]$. Evaluating (3.22) this results in

$$|\overline{B_2B_3} - \overline{B_1B_2}| = |(B_3 - 1) - (-3)| \geq 2, \quad \forall B_3 \in [0, 1].$$

However, it also holds that

$$|\overline{A_2A_3} - \overline{A_1A_2}| = |(1 - 4) - (-2 - 1)| = 0,$$

yielding

$$|\overline{B_2B_3} - \overline{B_1B_2}| > |\overline{A_2A_3} - \overline{A_1A_2}|.$$

This contradicts the statement of (3.22). Moreover, with $B_3 \in [0, 1]$, B_3 cannot be chosen such that $\overline{B_2B_3} = \overline{B_1B_2}$ is satisfied. As a result, this simple counterexample disproves Lemma 3.2 by showing that one cannot always find an N-path fulfilling (3.22) and (3.23). It is straightforward to verify numerically that counterexamples can also be constructed for longer paths in higher dimensions. For more information, the reader is referred to Hastedt and Werner (2023c).

In the proof of Lemma 3.2 provided in Zhang et al. (2015), the authors claim to prove (3.22) and (3.23) based on adding additional points to the N-path T_A, effectively making it an N'-path with $N' > N$. However, this contradicts the statement of Lemma 3.2 which requires T_A and T_B to be paths of length N. Additionally, as the length of the N-paths is determined by the prediction horizon M of the optimization problems, adding additional points to the N-path $T^*_{Q_k}$ would result in a change of the prediction horizon after the optimization has been solved.

In summary, the above discussion shows that the stability analysis based on the concept of N-paths cannot be applied to schemes involving agents with double-integrator dynamics.

3.4 On the Analysis of Model Predictive Flocking Using Compatibility Constraints

As explained in Section 3.2 and Section 3.3, the prediction mismatch can cause the line of reasoning in the DMPF analysis to break if not considered explicitly. One approach

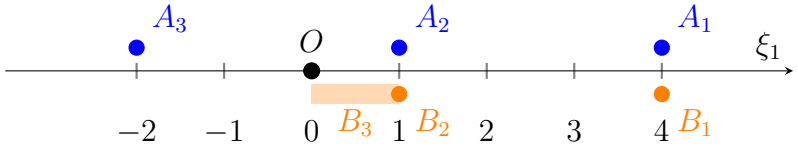


Figure 3.3: Visualization of counterexample for Lemma 3.2 with admissible region for B_3 indicated by orange rectangle.

to addressing this issue is via so-called compatibility or consistency constraints. By constraining the extent to which an agent may deviate from its decision at the previous time step, these compatibility constraints can effectively bound the prediction mismatch.

In Keviczky et al. (2004), feasibility is guaranteed for a DMPF scheme with coupled constraints by robustifying the constraints for all possible behaviors of neighboring agents. However, this constitutes a highly conservative approach. In Dunbar and Murray (2006), the authors propose a DMPC algorithm for formation control with hard compatibility constraints. By choosing the bound on the compatibility constraint depending on the sampling time and selecting the sampling time "small enough", the authors are able to show stability of their DMPC scheme using compatibility constraints. In the same year, in Keviczky et al. (2006), the authors consider a DMPC scheme with costs and constraints coupled via states and inputs, and show that, by adding a compatibility penalty to the cost function, a condition for checking stability can be formulated based on the magnitude of the compatibility penalty terms. It is important to emphasize that this work only shows that there exists such a condition, but not how to systematically design DMPC algorithms satisfying this condition. Such a condition was later formulated in Dunbar and Caveney (2012) for DMPF of vehicle platoons. By considering state couplings in the cost and adding a compatibility penalty term to the cost function, the authors show that stability can be guaranteed by a condition on the magnitude of the cost function weights and the prediction mismatch when considering only a single neighbor. In Zheng et al. (2017), this concept is then extended to vehicle platoons with unidirectional communication considering multiple neighbors.

In the context of MPF, compatibility constraints have been applied at least twice: once in Hu et al. (2017), where the authors propose a DMPF scheme with communication delays and once in Hu et al. (2018), where the authors consider a self-triggered scheme in continuous time with additional compatibility constraints. While in Hu et al. (2017), the deviation from the previous trajectories is only penalized in the cost function, in Hu et al. (2018), the authors consider hard constraints on the prediction error. In the following, the setup and line of reasoning in the analysis of both of these approaches will be presented and critically discussed. Similar to the discussion in the previous section, a careful analysis of the proposed lines of reasoning reveals that due to negligence of the general challenges in MPF (Section 3.2) and assumptions that do not hold in general, the analysis results in Hu et al. (2018, 2017) do not hold.

3.4.1 Compatibility Constraints

Consider a distributed MPC problem, where at time k, agents exchange their current states $x_i(k)$ and their optimal optimal input sequences from the previous step $U_i^*(k-1)$. At time k, with agent i having access to $U_j^*(k-1)$, each agent can compute

$$\hat{U}_j(k|k-1) = \mathrm{col}\big(u_j^*(k|k-1), \dots, u_j^*(k+M-2|k-1), 0\big), \quad \forall j \in \mathcal{N}_i(k) \cup i.$$

Agents can then compute the estimated states of the i^{th} subsystem according to

$$\hat{x}_j(k+m|k) = \phi_j(m; x_j(k), \hat{U}_j(k|k-1)), \quad \forall j \in \mathcal{N}_i(k) \cup i,\, m \in \mathbb{I}_{0:M}.$$

Compatibility constraints can then be included in the optimization problem as hard constraints $\|x_i - \hat{x}_i\| \leq \mathcal{K}$, where $\mathcal{K} > 0$ is the admissible mismatch, or by adding a term in the form of $\|x_i - \hat{x}_i\|_{R_c}$ to the stage cost, where $R_c \succeq 0$ is the compatibility weight. One important result used in the frameworks based on compatibility constraints follows directly from the triangle inequality and properties of the vector norm. Given $a, b, c \in \mathbb{R}^n$, it holds that

$$\|a\| - \|b\| \leq \|a - b\|,$$

and consequently, with $R \succ 0$,

$$\|a - c\|_R - \|b - c\|_R \leq \|a - b\|_R. \tag{3.43}$$

3.4.2 On the Analysis of Distributed Model Predictive Flocking in Hu et al. (2017)

In Hu et al. (2017), the authors consider a DMPF scheme with communication delays, where stability is guaranteed by penalizing deviations from the previous optimal trajectory in the cost function. As the discussions in this section are independent of the communication delay in Hu et al. (2017), the delay is considered to be zero here and removed from the problem statement for ease of notation.

Problem Statement

In their work, the authors consider a homogeneous MAS with agents governed by the linear discrete-time dynamics in (2.6),

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k).$$

For their optimization problem, the authors consider terminal equality constraints, no state constraints, and input constraints in the form of (3.7), resulting in the following DMPF problem:

Problem 3.3.

$$\begin{aligned} V_i^*(x_i(k), \hat{\tilde{X}}_i^M(k)) = &\min_{U_i(k)} V_i(x_i(k), \hat{\tilde{X}}_i^M(k), U_i(k)) \\ &\text{s.t. } x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ &\quad U_i(k) \in \mathcal{U}_i, \\ &\quad x_i(k+M) = x_{i,f}, \end{aligned}$$

with

$$V_i(x_i(k), \hat{\tilde{X}}_i^M(k), U_i(k)) = \sum_{m=0}^{M-1} \ell_i(x_{i,k+m+1|k}, \hat{\tilde{x}}_{i,k+m+1|k}, u_{i,k+m|k}) \tag{3.44}$$

and

$$\ell_i(x_i, \hat{\tilde{x}}_i, u_i) = \|u_i\|_{R_u} + \|p_i - p_{\mathrm{r}_i}\|_{R_p} + \sum_{j \in \mathcal{N}_i} \left\| q_{ij} - d\frac{\hat{q}_{ij}}{\|\hat{q}_{ij}\|} \right\|_{R_\delta} + \|q_i - \hat{q}_i\|_{R_c}. \tag{3.45}$$

Here, $R_u, R_p, R_\delta, R_c \succ 0$ are positive definite weighting matrices. The last term in (3.45) is the compatibility cost penalizing the deviations from the decision at the previous step. Note that, in this cost, the convex approximation of the flocking error is used (Section 3.2.1). Furthermore, contrary to the DMPF problems stated in Section 3.1 and Section 3.3, the individual normed terms in the stage cost are not squared, resulting in a cost function that is not continuously differentiable.

In Hu et al. (2017), the terminal equality constraint is given by

$$x_{i,f} = \begin{bmatrix} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \left(\hat{q}_{j,k+M|k} - d \frac{\hat{q}_{ij,k+M|k}}{\|\hat{q}_{ij,k+M|k}\|} \right) \\ p_\mathrm{r} \end{bmatrix}. \tag{3.46}$$

Note that the desired position can also be expressed as

$$\sum_{j \in \mathcal{N}_i} \left(\hat{q}_{j,k+M|k} - q_{i,k+M|k} - d \frac{\hat{q}_{ij,k+M|k}}{\|\hat{q}_{ij,k+M|k}\|} \right) = 0, \tag{3.47}$$

which corresponds to the average flocking error over all agents being zero. Moreover, contrary to Problem 2.3, Problem 3.3 now depends on $\hat{\tilde{X}}_i^M$, not only $\hat{X}_{\tilde{i}}^M$, as the consistency penalty depends on $\hat{X}_i^M$. With agents exchanging their previous optimal input trajectories $U_i^*(k-1)$, agents can compute $\hat{U}_j(k|k-1)$ and $\hat{X}_j(k|k)$ for all $j \in \mathcal{N}_i(k) \cup i$ according to (3.15) and (3.16).

Stability Analysis

For the analysis, the authors first state the following assumption and lemma.

Assumption 3.4 (Hu et al. (2017, Assumption 1))**.** There exists an integer $\kappa \geq 0$ such that the MAS is connected across the interval $[k, k+\kappa]$ for all $k \geq 0$.

Lemma 3.3 (Hu et al. (2017, Lemma 1))**.** *Let $\mathbb{X}_0 \subseteq \mathbb{R}^{n_x}$ be the set of states for which Problem 3.3 is feasible. Then, if $x(0) \in \mathbb{X}_0$, (3.3) is feasible for all $k > 0$.*

Here, Assumption 3.4 is equivalent to the connectivity assumption in the N-path analysis (Section 3.3). Lemma 3.3 ensures recursive feasibility. The proof is based on the argument that, if the problem is feasible at time $k = 0$, the warm start $\tilde{U}(k)$ in (2.59) is always a feasible control sequence at time $k+1$.

The main stability result in Hu et al. (2017) is then stated as follows:

Theorem 3.4 (Hu et al. (2017, Theorem 1))**.** *If Assumption 3.4 holds, $x(0) \in \mathbb{X}_0$, and*

$$\sum_{j \in \mathcal{N}_i} R_\delta \leq R_c, \tag{3.48}$$

the agents will form an α-lattice.

Note that (3.48) is equivalent to $|\mathcal{N}_i| R_\delta \leq R_c$ if all agents use the same weighting matrices. In the following, the line of reasoning used in the proof is outlined. For more details, the reader is referred to Hu et al. (2017).

Consider the value function $V_i^*(x_i(k), \hat{\tilde{X}}_i^M(k))$ at time k, with $U_i^*(k)$ being the corresponding optimal solution. Then, constructing the warm start $\tilde{U}_i(k+1|k)$ according to (2.59), the authors consider the sum of the local value functions as Lyapunov function, yielding

$$\sum_{i=1}^{N} V_i\big(x_i(k+1), \hat{\tilde{X}}_i^M(k+1), \tilde{U}_i(k+1|k)\big) - \sum_{i=1}^{N} V_i^*\big(x_i(k), \hat{\tilde{X}}_i^M(k)\big) \leq 0. \tag{3.49}$$

Since the warm start is not the optimal solution at time $k+1$, it follows that

$$\sum_{i=1}^{N} V_i^*\big(x_i(k+1), \hat{\tilde{X}}_i^M(k+1)\big) - \sum_{i=1}^{N} V_i^*\big(x_i(k), \hat{\tilde{X}}_i^M(k)\big) \leq 0. \tag{3.50}$$

Based on this result, the authors then establish asymptotic stability similar to the line of reasoning presented in Section 2.4.2.

The key step in this analysis is establishing that (3.49) holds, for which the compatibility constraints and (3.48) in Lemma 3.3 are required. To show that this holds, consider the difference in (3.49) for a single agent. Note that, with agents exchanging information between times k and $k+1$, due to the design of the warm start, it holds that

$$\hat{\tilde{x}}_i(k+m+1|k+1) = \bar{x}_i^*(k+m+1|k), m \in \mathbb{I}_{0:M-1}. \tag{3.51}$$

By canceling terms appearing in both, $V_i\big(x_{i,k+1}, \hat{\tilde{X}}_{i,k+1}^M, \tilde{U}_{i,k+1}\big)$ and $V_i^*\big(x_{i,k}, \hat{\tilde{X}}_{i,k}^M\big)$, it follows that

$$\begin{aligned}
\Delta V_{i,k} =& V_i\big(x_{i,k+1}, \hat{\tilde{X}}_{i,k+1}^M, \tilde{U}_{i,k+1}\big) - V_i^*\big(x_i(k), \hat{\tilde{X}}_i^M(k)\big) && (3.52)\\
=& -\ell_i(x_{i,k+1|k}, \hat{\tilde{x}}_{i,k+1|k}, u_{i,k}^*) + \ell_i(\tilde{x}_{i,k+1+M|k+1}, \hat{\tilde{x}}_{i,k+1+M|k+1}, 0) && (3.52a)\\
&+ \sum_{m=1}^{M-1} \sum_{j \in \mathcal{N}_i} \Big(\Big\| (q_{j,k+m+1|k}^* - q_{i,k+m+1|k}^*) - d \frac{q_{ij,k+m+1|k}^*}{\|q_{ij,k+m+1|k}^*\|} \Big\|_{R_\delta} && (3.52b)\\
&- \Big\| (\hat{q}_{j,k+m+1|k} - q_{i,k+m+1|k}^*) - d \frac{\hat{q}_{ij,k+m+1|k}}{\|\hat{q}_{ij,k+m+1|k}\|} \Big\|_{R_\delta} \Big) && (3.52c)\\
&- \sum_{m=1}^{M-1} \|q_{i,k+m+1|k}^* - \hat{q}_{i,k+m+1|k}\|_{R_c}. && (3.52d)
\end{aligned}$$

The authors then further simplify this equation. They claim that, according to the terminal equality constraint, the second term in (3.52a) is zero. Furthermore, by making the assumption

$$\frac{\hat{q}_{ij,k+m|k}}{\|\hat{q}_{ij,k+m|k}\|} = \frac{q_{ij,k+m|k}^*}{\|q_{ij,k+m|k}^*\|}, \quad \forall (i,j) \in \mathcal{E}(q), \quad m \in \mathbb{I}_{0:M}, \tag{3.53}$$

and applying the triangle inequality (3.43) to (3.52b) and (3.52c), (3.52) can be simplified to

$$\Delta V_{i,k} \leq \sum_{m=1}^{M-1} \left(\sum_{j \in \mathcal{N}_i} \|q^*_{j,k+m+1|k} - \hat{q}_{j,k+m+1|k}\|_{R_\delta} - \|q^*_{i,k+m+1|k} - \hat{q}_{i,k+m+1|k}\|_{R_c} \right). \tag{3.54}$$

Then, by summing $\Delta V_{i,k}$ over all agents, one obtains

$$\begin{aligned} \sum_{i=1}^{N} \Delta V_{i,k} &\leq \sum_{m=1}^{M-1} \left(\sum_{i=1}^{N} \Big(\sum_{j \in \mathcal{N}_i} \|q^*_{i,k+m+1|k} - \hat{q}_{i,k+m+1|k}\|_{R_\delta} - \|q^*_{i,k+m+1|k} - \hat{q}_{i,k+m+1|k}\|_{R_c} \Big) \right) \\ &\leq \sum_{m=1}^{M-1} \left(\sum_{i=1}^{N} \left(\|q^*_{i,k+m+1|k} - \hat{q}_{i,k+m+1|k}\|_{|\mathcal{N}_i|R_\delta - R_c} \right) \right). \end{aligned} \tag{3.55}$$

Consequently, if (3.48) holds, it follows that $\sum_{i=1}^{N} \Delta V_{i,k} \leq 0$, showing that (3.49) holds.

Critical Discussion

While the above analysis seems to elegantly address the prediction mismatch in DMPF using compatibility constraints, a careful examination of the line of reasoning presented in Hu et al. (2017) reveals several problems. In the following, these will be discussed in detail.

Time-Varying Communication Topology Analogous to the MPF schemes discussed in Section 3.3, in Hu et al. (2017), the authors only assume connectedness across an interval, allowing for time-varying communication topologies (Assumption 3.4). However, as in the previous discussion about the analysis using N-paths, changes in the communication graph are not addressed explicitly. This results in the same problem as described in Section 3.2.3. Changes in the communication topology can introduce additional positive terms in (3.52b), which can break the line of reasoning due to $\sum_{i=1}^{N} \Delta V_{i,k} \leq 0$ no longer holding. Consequently, the presented analysis does not hold in general, but only under the additional constraint that no edges are added, since otherwise (3.49), and therefore (3.50), do not necessarily hold. For the remainder of this discussion, it is therefore assumed that the communication topology is static.

Recursive Feasibility By Lemma 3.3, the authors claim to establish recursive feasibility. However, due to the distributed implementation of the algorithm, this is not true in general. Consider $p_{\mathrm{r}_i} = 0$. With the warm start (2.59), it follows that, $\forall i \in \mathcal{V}$,

$$\tilde{x}_{i,k+1+M|k+1} = x^*_{i,k+M|k}.$$

By design, the warm start sequences are based on the optimal solution at the previous time step, so $\hat{\tilde{X}}_{i,k}$ is constructed from $U^*_{j,k-1}$, for all $j \in \mathcal{N}_i(k) \cup i$, and $\hat{\tilde{X}}_{i,k+1}$ follows from $U^*_{i,k}$.

However, as the warm start is not necessarily the solution to the optimization problem at time k, it is likely that $U^*_{j,k-1}$ and $U^*_{i,k}$ are different. This results in

$$\hat{\tilde{x}}_{i,k+m|k+1} \neq \hat{\tilde{x}}_{i,k+m|k}, \quad m \in \mathbb{I}_{1:M}. \tag{3.56}$$

Consequently, changes in the neighbors' control decisions can cause (3.47) to no longer hold when agent i is applying the warm start sequence, invalidating the arguments in the proof of Lemma 3.3 and showing that the MPC scheme in Hu et al. (2017) is not inherently recursively feasible.

Mismatch Between Terminal Equality Constraint and α-Lattice In their analysis, the authors state that, in (3.52a), $\ell_i(\tilde{x}_{i,k+1+M|k+1}, \hat{\tilde{x}}_{i,k+1+M|k+1}, 0)$ is equal to zero due to the terminal equality constraint. Note that, with (3.51) and the terminal velocity constraint, it follows that

$$\ell_i(\tilde{x}_{i,k+1+M|k+1}, \hat{\tilde{x}}_{i,k+1+M|k+1}, 0) = \sum_{j\in\mathcal{N}_i} \left\| \tilde{q}_{ij,k+1+M|k+1} - d\frac{\tilde{q}_{ij,k+1+M|k+1}}{\|\tilde{q}_{ij,k+1+M|k+1}\|} \right\|_{R_\delta}. \tag{3.57}$$

However, the terminal equality constraint only drives the average formation error of each agent to zero instead of the absolute one. Consequently, there exist formations for which (3.47) is zero but which are not α-lattices,

$$\begin{aligned} 0 &= \sum_{j\in\mathcal{N}_i} \left(\tilde{q}_{ij,k+1+M|k+1} - d\frac{\tilde{q}_{ij,k+1+M|k+1}}{\|\tilde{q}_{ij,k+1+M|k+1}\|} \right) \\ &\leq \left\| \sum_{j\in\mathcal{N}_i} \left(\tilde{q}_{ij,k+1+M|k+1} - d\frac{\tilde{q}_{ij,k+1+M|k}}{\|\tilde{q}_{ij,k+1+M|k}\|} \right) \right\| \\ &\leq \sum_{j\in\mathcal{N}_i} \left\| \tilde{q}_{ij,k+1+M|k+1} - d\frac{\tilde{q}_{ij,k+1+M|k}}{\|\tilde{q}_{ij,k+1+M|k}\|} \right\|. \end{aligned}$$

Hence, the terminal equality constraint (3.47) not necessarily implies (3.57) being zero. As a result, the positive remainder in (3.52a) can lead to (3.49) not holding, breaking the line of reasoning.

Assumption on Estimated Agent Trajectories As already pointed out several times in this thesis, the warm start and corresponding estimated state sequence are not necessarily the optimal ones at time k. Handling the mismatch between the warm start and the optimal solution at time k is the main reason for introducing the compatibility constraints in the first place. In their proof, the authors of Hu et al. (2017) make the assumption in (3.53), which effectively removes the error introduced by the prediction mismatch in (3.52b) and (3.52c) by assuming equality of the estimated position sequence computed from the warm start and the optimal solution at time k. Based on the discussion above, however, assumption (3.53) cannot be expected to hold. With (3.53) not holding, (3.52b) and (3.52c) cannot be simplified to (3.54) as the terms in (3.53) do not cancel out. Consequently, (3.55) cannot be established, breaking the line of reasoning in the analysis.

3.4.3 On the Analysis of Distributed Model Predictive Flocking in Hu et al. (2018)

Unlike the previously discussed MPF algorithms, in Hu et al. (2018), the authors consider a continuous-time MPC scheme with a self-triggered mechanism to determine the next sampling instance. In contrast to event-triggered control, where the next sampling instance is determined by a monitored trigger condition, the self-triggered mechanism predetermines the next sampling instances based on the prediction of the future system behavior. Hence, event-triggered control can be considered reactive while self-triggered control is proactive (Heemels et al. 2012). To ensure convergence towards the desired formation and collision avoidance, in Hu et al. (2018), the authors include two compatibility constraints in their problem formulation. In the following, the posed problem and line of reasoning of the stability analysis are outlined first. Afterwards, the analysis will be discussed critically, pointing out several issues that can break the line of reasoning.

Problem Statement

In their work, the authors consider a homogeneous MAS with the agents being governed by linear continuous-time dynamics in (2.5),

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t). \tag{2.5}$$

Let t_k^i denote the k^{th} sampling instance of agent i. Furthermore, let $T > 0$ denote the continuous-time prediction horizon. With some slight abuse of notation, let $U_i(t_k^i)$ be the input trajectory for the time interval $[t_k^i, t_k^i + T]$. The corresponding state trajectory obtained from solving (2.5) with input $U_i(t_k^i)$ is given by $X_i(t_k^i)$. Optimal, estimated, and warm start variables are then defined as in the discrete-time case using the $*$, $\wedge$, and $\sim$ annotations, respectively.

In Hu et al. (2018), the following assumption is made on the order of triggering instances of neighboring agents.

Assumption 3.5. For each agent i, between sampling instances t_{k-1}^i and t_k^i, each neighbor of agent i triggers exactly once, meaning that, for all $j \in \mathcal{N}_i$,

$$t_{k-1}^i \leq t^j < t_k^i.$$

Analogous to (2.59), consider the following definitions of the continuous-time warm start trajectory and estimated neighbor trajectory based on the optimal solution at the previous triggering instance:

$$\hat{u}_i(t|t_k^i) = \tilde{u}_i(t|t_k^i) = \begin{cases} u_i^*(t|t_{k-1}^i), & t \in [t_k^i, t_{k-1}^i + T] \\ 0, & t \in (t_{k-1}^i + T, t_k^i + T], \end{cases} \tag{3.58}$$

$$\hat{u}_j(t|t_k^i) = \begin{cases} u_j^*(t|t_k^i), & t \in [t_k^i, t_k^j + T] \\ 0, & t \in (t_k^j + T, t_k^i + T]. \end{cases} \tag{3.59}$$

For the continuous-time DMPF problem, consider the cost function

$$V_i\big(x_i(t_k^i), \hat{\tilde{X}}_i(t_k^i), U_i(t_k^i)\big) = \int_{t_k^i}^{t_k^i+T} \ell_i\big(x_i(\tau|t_k^i), \hat{\tilde{x}}_i(\tau|t_k^i), u_i(\tau|t_k^i)\big)\,\mathrm{d}\tau \tag{3.60}$$

with stage cost

$$\ell_i\big(x_i, \hat{\tilde{x}}_i, u_i\big) = r_u\|u_i\| + r_p\|p_i - p_{\mathrm{r}_i}\| + r_\delta \sum_{j\in\mathcal{N}_i} \big|\|q_i - \hat{q}_j\| - d\big|. \tag{3.61}$$

Here, $r_u, r_p, r_\delta > 0$ are scalar weights and p_{r_i} is the reference velocity. As in the DMPF scheme with compatibility constraints discussed in Section 3.4.2, to apply the triangle inequality (3.43) in the analysis, the individual normed terms in the stage cost are not squared, resulting in a cost function that is not continuously differentiable. To simplify the notation in the following discussion, the abbreviated notation

$$\ell_i(\tau|t_k^i) = \ell_i\big(x_i(\tau|t_k^i), \hat{\tilde{x}}_i(\tau|t_k^i), u_i(\tau|t_k^i)\big)$$

is used. The continuous-time DMPF problem can then be stated as follows:

Problem 3.4.

$$\begin{aligned}
V_i^*(x_i(t_k^i), \hat{\tilde{X}}_i(t_k^i)) = \min_{U_i(t_k^i)}\ & V_i\big(x_i(t_k^i), \hat{\tilde{X}}_i(t_k^i), U_i(t_k^i)\big) \\
\text{s.t. }\ & \dot{x}_i(t|t_k^i) = A_i x_i(t|t_k^i) + B_i u_i(t|t_k^i), \quad t \in [t_k^i, t_k^i + T], \\
& u_i(t|t_k^i) \in \mathbb{U}, \quad t \in [t_k^i, t_k^i + T], \\
& \|q_i(t|t_k^i) - \hat{q}_i(t|t_k^i)\| \le \mathcal{K}, \quad t \in [t_k^i, t_k^i + T], \\
& \|q_i(t|t_k^i) - \hat{q}_j(t|t_k^i)\| \ge 2R + \mathcal{K}, \quad j \in \mathcal{N}_i, t \in [t_k^i, t_k^i + T], \\
& \|p_i(t|t_k^i) - p_{\mathrm{r}_i}\| = 0.
\end{aligned}$$

In this problem, the third constraint is the compatibility constraint with maximum admissible compatibility deviation $\mathcal{K}$. The fourth constraint ensures collision avoidance with a safety distance $2R$. Note that there is no terminal constraint on the position of agent i.

For the self-triggered mechanism determining the next triggering instance, the authors first define a continuous, positive, and decreasing function $b(t)$ with $b(0) \le 1$ and a continuous, positive, and increasing function

$$B(t) = \int_0^t b(\tau)\,\mathrm{d}\tau. \tag{3.62}$$

Note that, by definition, $B(t)$ can at most increase linearly with a slope of 1. The next sampling instance $t_{k+1}^i = t_k^i + \Delta_k^i$, $\Delta_k^i \in (0, T]$, is then determined by solving

$$\zeta \int_{t_k^i}^{ti_k+\Delta_k^i} \ell_i^*(\tau|t_k^i)\,\mathrm{d}\tau = r_\delta(T - \Delta_k^i)|\mathcal{N}_i|\mathcal{K} + B(\Delta_k^i)\ell_i^*(t_k^i + T|t_k^i) \tag{3.63}$$

for Δ_k^i, with $\zeta \in (\zeta_{\min}, 1)$ and

$$\zeta_{\min} = \max_{\Delta_k^i \in (0,T]} \frac{B(\Delta_k^i)b(T - \Delta_k^i)}{\Delta_k^i}.$$

Stability Analysis

In the following, the line of reasoning of the analysis in Hu et al. (2018) is outlined. For more details, the reader is referred to the original paper. Firstly, note that collision avoidance follows directly from the compatibility constraint (Hu et al. 2018, Lemma 1) and the triangle inequality with

$$\begin{aligned}\|q_i(t|t_k^i) - q_j(t|t_k^i)\| &= \|q_i(t|t_k^i) - q_j(t|t_k^i) + \hat{q}_j(t|t_k^i) - \hat{q}_j(t|t_k^i)\| \\ &\geq \|q_i(t|t_k^i) - \hat{q}_j(t|t_k^i)\| - \|\hat{q}_j(t|t_k^i) - q_j(t|t_k^i)\| \\ &\geq \|q_i(t|t_k^i) - \hat{q}_j(t|t_k^i)\| - \mathcal{K}.\end{aligned} \tag{3.64}$$

Consequently, from $\|q_i(t|t_k^i) - \hat{q}_j(t|t_k^i)\| \geq 2R + \mathcal{K}$, it follows that $\|q_i(t|t_k^i) - q_j(t|t_k^i)\| \geq 2R$.

To formulate their stability result, the authors first state the following assumptions and lemmas.

Assumption 3.6 (Hu et al. (2018, Assumption 1))**.** For all $t' \in [t_k^i, t_k^i + T]$ and $t \in [t', t_k^i + T]$, it holds that

$$\ell_i^*(t|t_k^i) \leq b(t - t')\ell_i^*(t'|t_k^i). \tag{3.65}$$

Assumption 3.7 (Hu et al. (2018, Assumption 2))**.** For all $\Delta_k^i \in [0, T]$ and $x_i, x_j \in \mathbb{R}^{n_{x,i}}$, $i \in \mathcal{N}_i$, it holds that

$$\int_{t_k^i+T}^{t_k^i+\Delta_k^i+T} \hat{\ell}_i(\tau|t_k^i + \Delta_k^i)\,\mathrm{d}\tau \leq B(\Delta_k^i)\ell_i^*(t_k^i + T|t_k^i). \tag{3.66}$$

Assumption 3.8 (Hu et al. (2018, Assumption 3))**.** There exists a $\tau \geq 0$ such that the MAS is connected across the interval $[t, t + \tau]$ for any $t \geq 0$.

While Assumption 3.6 imposes a tunable decay rate on the stage cost over the prediction horizon, Assumption 3.7 limits the increase of the cost function when the estimated input trajectory (3.58) is applied beyond the prediction horizon. Assumption 3.8 ensures connectivity across an interval analogous to Assumption 3.2 and Assumption 3.4. An important result is that the triggering-condition (3.63) has a solution at every triggering instance. In Hu et al. (2018) this is established with the following result.

Lemma 3.5 (Hu et al. (2018, Lemma 4))**.** *With Assumption 3.6, there always exists a* $\Delta_k^i \in (0, T]$ *satisfying* (3.63) *and* $\zeta \in (\zeta_{min}, 1)$.

The proof is provided in Hu et al. (2018). The following lemma then establishes recursive feasibility.

Lemma 3.6 (Hu et al. (2018, Lemma 2))**.** *If Problem 3.4 has a solution at time* t_k^i, (3.58) *is a feasible solution at* t_{k+1}^i.

The proof is based on showing that (3.58) satisfies the terminal velocity constraint. The satisfaction of the compatibility constraint in Problem 3.4 follows from the definition of (3.58). Collision avoidance is guaranteed by (3.64).

Lemma 3.7 (Hu et al. (2018, Lemma 3))**.** *If Assumption 3.7 holds, then*

$$V_i^*(x_i(t_{k+1}^i), \hat{\tilde{X}}_i(t_{k+1}^i)) - V_i^*(x_i(t_k^i), \hat{\tilde{X}}_i(t_k^i)) \leq -\int_{t_k^i}^{ti_k+\Delta_k^i} \ell_i^*(\tau|t_k^i)\, d\tau + r_\delta(T-\Delta_k^i)|\mathcal{N}_i|\mathcal{K} + B(\Delta_k^i)\ell_i^*(t_k^i+T|t_k^i). \tag{3.67}$$

The proof of this lemma again uses that $V_i(x_i(t_{k+1}^i), \hat{\tilde{X}}_i(t_{k+1}^i), \tilde{U}_i(t_{k+1}^i))$ is not necessarily the optimal solution at time t_{k+1}^i, so

$$V_i^*(x_i(t_{k+1}^i), \hat{\tilde{X}}_i(t_{k+1}^i)) \leq V_i(x_i(t_{k+1}^i), \hat{\tilde{X}}_i(t_{k+1}^i), \tilde{U}_i(t_{k+1}^i)). \tag{3.68}$$

By considering the difference

$$\Delta V_i = V_i(x_i(t_{k+1}^i), \hat{\tilde{X}}_i(t_{k+1}^i), \tilde{U}_i(t_{k+1}^i)) - V_i^*(x_i(t_k^i), \hat{\tilde{X}}_i(t_k^i)), \tag{3.69}$$

it follows that

$$\Delta V_i \leq -\int_{t_k^i}^{t_k^i+\Delta_k^i} \ell_i^*(\tau|t_k^i)\,\mathrm{d}\tau \tag{3.70}$$

$$+\int_{t_k^i+\Delta_k^i}^{t_k^i+T} \left(\hat{\ell}_i(\tau|t_{k+1}^i) - \ell_i^*(\tau|t_k^i)\right)\mathrm{d}\tau \tag{3.70a}$$

$$+\int_{t_k^i+T}^{t_k^i+\Delta_k^i+T} \hat{\ell}_i(\tau|t_{k+1}^i)\,\mathrm{d}\tau. \tag{3.70b}$$

Comparing this inequality to (3.67), the first term in (3.67) is equal to (3.70) and the third term follows from upper-bounding (3.70b) using Assumption 3.7. Upper-bounding (3.70a) by the second term in (3.67) follows analogously to (3.54). By inserting the definition of the cost function, canceling equal terms, and applying the triangle inequality, it follows that

$$\begin{aligned}\int_{t_k^i+\Delta_k^i}^{t_k^i+T} \left(\hat{\ell}_i(\tau|t_{k+1}^i) - \ell_i^*(\tau|t_k^i)\right)\mathrm{d}\tau \leq & r_\delta \int_{t_k^i+\Delta_k^i}^{t_k^i+T} \Big(\sum_{j\in\mathcal{N}_i} \big|\|q_i^*(\tau|t_k^i) - \hat{q}_j(\tau|t_{k+1}^i)\| - d\big| \\ & - \sum_{j\in\mathcal{N}_i} \big|\|q_i^*(\tau|t_k^i) - \hat{q}_j(\tau|t_k^i)\| - d\big|\Big)\,\mathrm{d}\tau \\ \leq & r_\delta \int_{t_k^i+\Delta_k^i}^{t_k^i+T} \Big(\sum_{j\in\mathcal{N}_i} \big\|\hat{q}_j(\tau|t_{k+1}^i) - \hat{q}_j(\tau|t_k^i)\big\|\Big)\,\mathrm{d}\tau \\ \leq & r_\delta(T-\Delta_k^i)|\mathcal{N}_i|\mathcal{K}.\end{aligned}$$

Based on these assumptions and lemmas, the stability theorem is stated as follows:

Theorem 3.8 (Hu et al. (2018, Theorem 1))**.** *Suppose Assumptions 3.6-3.8 hold and Problem 3.4 is feasible at $t = 0$. Then the MAS forms an α-lattice with a constant velocity and collisions are avoided.*

With the previously established Lemmas 3.6 and 3.7, the proof of this theorem is straightforward. From (3.7) and the trigger-condition in (3.63), it follows that

$$V_i^*(x_i(t_{k+1}^i), \hat{\tilde{X}}_i(t_{k+1}^i)) - V_i^*(x_i(t_k^i), \hat{\tilde{X}}_i(t_k^i)) \leq -(1-\zeta) \int_{t_k^i}^{ti_k+\Delta_k^i} \ell_i^*(\tau|t_k^i)\,\mathrm{d}\tau \leq 0, \quad (3.71)$$

establishing that the value function is a Lyapunov function. With the cost only being zero when an α-lattice with velocity p_{r_i} is formed, the authors conclude that the MAS converges towards the desired configuration.

Critical Discussion

In the following, the analysis presented above is critically discussed. As in the previously discussed analyses, in Hu et al. (2018), the authors do not appropriately handle time-varying communication topologies. Furthermore, there are issues with Assumptions 3.5-3.7 since they cannot be expected to hold in general. If possible, changes to the problem formulation to fix these problems are proposed.

Time-Varying Communication Topology As in the discussion in Section 3.4.2 and Section 3.3, in Hu et al. (2018), the authors only assume connectedness across an interval, allowing for time-varying communication topologies (Assumption 3.8). However, as in the other discussed analysis results in this chapter, changes in the communication graph are not addressed explicitly. Such changes may result in a positive remainder in (3.70a), which leads to (3.67) in Lemma 3.7 not holding, since not all terms in (3.70a) can be upper-bounded as described above. With Lemma 3.7 not holding, the line of reasoning breaks since (3.71) in Theorem 3.8 no longer holds. Consequently, the presented analysis does not hold under the assumption of time-varying communication topologies as stated in Assumption 3.8 but only for constant ones. For the remainder of this discussion, it is therefore again assumed that the communication topology is static.

Validity of Assumption 3.5 In Assumption 3.5, it is stated that between triggering instances of agent i, all of its neighbors trigger exactly once. However, for some configurations, this imposes an exact sequence of trigger events that have to happen. Consider for example the scenario with three agents, as depicted in Fig. 3.4 where all three agents are neighbors. Assume that the initial triggering order is

$$t_k^1 < t_k^2 < t_k^3.$$

Then, the only possible order for the agents' next triggering instances is given by

$$t_{k+1}^1 < t_{k+1}^2 < t_{k+1}^3,$$

since switching the order would violate Assumption 3.5 for at least one of the agents. For example, consider the order

$$t_k^1 < t_k^2 < t_k^3 < t_{k+1}^1 < t_{k+1}^3 < t_{k+1}^2$$

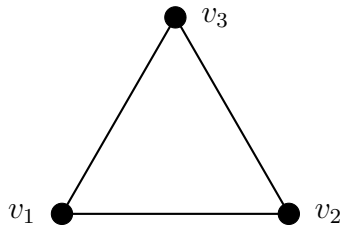


Figure 3.4: Scenario for demonstrating triggering order in Hu et al. (2018).

where t_{k+1}^2 and t_{k+1}^3 are switched. Then, Assumption 3.5 does not hold for agent 3 since agent 2 does not trigger between t_k^3 and t_{k+1}^3. Furthermore, from the perspective of agent 2, agent 3 triggers twice between t_k^2 and t_{k+1}^2. With each agent determining its next triggering instance based on (3.63), without considering the triggering instances of its neighbors, this means that neighboring agents can trigger multiple times or not at all between an agent's sampling instances.

However, since by the statement of Problem 3.4, the compatibility constraints limit the deviation from the previous trajectory by $\mathcal{K}$ at each sampling instance, having neighbor j trigger n_j times between sampling instances of agent i increases the possible deviation observed by agent i to

$$\|\hat{q}_j(t|t_{k+1}^i) - q_j(t|t_k^i)\| \leq n_j\mathcal{K}, \quad t \in [t_k^i, t_k^i + T], \tag{3.72}$$

which significantly impacts the line of reasoning in the stability analysis.

Firstly, collision avoidance can no longer be guaranteed by the collision avoidance constraint in Problem 3.4. By (3.64), to ensure collision avoidance, the required bound would have to be $2R+n_j\mathcal{K}$. Additionally, the statements of the trigger condition in (3.63) and Lemma 3.7 would require the adjustment of replacing $|\mathcal{N}_i|$ by $\sum_{j\in\mathcal{N}_i} n_j$ to account for Assumption 3.6 not holding. However, even if n_j would be known by agent i, this can cause the MPF problem to become infeasible since the bound in the collision avoidance constraint can change at each sampling instance while the compatibility constraint of agent i is still bounded by $\mathcal{K}$.

In summary, Assumption 3.5 cannot be expected to hold in general and violations of it can result in the MPF problem being infeasible.

Validity of Assumption 3.6 In Assumption 3.6, the authors assume that the decay rate of the stage cost over the prediction horizon can be considered a tuning parameter. Since this is not imposed as a constraint on the optimization problem, there is no justification for why the stage cost would satisfy (3.65). However, Assumption 3.6 is required in the proof of Lemma 3.5, which ensures that the trigger condition (3.63) is feasible at each sampling instance. Without it, the whole argument of the self-triggered scheme would break.

This issue may be resolved by imposing (3.65) in Assumption 3.6 as an additional constraint in the form of

$$\ell_i^*(t|t_k^i) - b(t-t')\ell_i^*(t'|t_k^i) \leq 0, \quad \forall t' \in [t_k^i, t_k^i + T],\, t \in [t', t_k^i + T],$$

to the optimization problem. However, depending on the choice of $b(t)$, this constraint could drastically reduce the size of the feasible set. Note that the constraint depends on the estimated neighbor trajectories. Similar to the discussion in Section 3.4.2, if these estimated trajectories change between sampling instances, the warm start in (3.58) is no longer necessarily a feasible solution at sampling instance t_{k+1}^i, resulting in

$$\hat{\ell}_i(t|t_{k+1}^i) - b(t-t')\hat{\ell}_i(t'|t_{k+1}^i) > 0, \quad \forall t' \in [t_{k+1}^i, t_{k+1}^i + T],\, t \in [t', t_{k+1}^i + T].$$

Consequently, the DMPF scheme would no longer be recursively feasible.

Validity of Assumption 3.7 In simple terms, Assumption 3.7 states that the increase of the cost function is limited when applying the estimated input trajectory (3.58) beyond the prediction horizon, regardless of how $\hat{\tilde{x}}_i$ changes from t_k^i to t_{k+1}^i. In the line of reasoning in Hu et al. (2018), Assumption 3.7 is essential to the proof of Lemma 3.7. However, as shown below, this assumption can easily be violated. Consider the reference velocity $p_{\mathrm{r}_i} = 0$. With (3.62), (3.66) can be rewritten as

$$\begin{aligned} 0 &\geq \int_{t_k^i+T}^{t_k^i+\Delta_k^i+T} \left(\hat{\ell}_i(\tau|t_{k+1}^i) - b(\tau - t_k^i - T)\ell_i^*(t_k^i+T|t_k^i)\right) \mathrm{d}\tau \\ &= \int_{t_k^i+T}^{t_k^i+\Delta_k^i+T} \Bigg(\sum_{j\in\mathcal{N}_i} \Big(\big| \|q_i^*(t_k^i+T|t_k^i) - \hat{q}_j(\tau|t_{k+1}^i)\| - d \big| \\ &\quad - b(\tau - t_k^i - T) \big| \|q_i^*(t_k^i+T|t_k^i) - \hat{q}_j(t_k^i+T|t_k^i)\| - d \big| \Big) \Bigg) \mathrm{d}\tau. \end{aligned}$$

Then, for $\hat{q}_j(t|t_{k+1}^i) = \hat{q}_j(t_k^i+T|t_k^i)$ for all $t \in [t_k^i+T, t_k^i+\Delta_k^i+T]$ and constant $b(t) = 1$, equality holds. However, as the shape of $b(t)$ is a tuning parameter, $b(t) < 1$ for $t > 0$ would lead to the right hand side being larger than zero, which contradicts the statement of Assumption 3.7. Similarly, if, while optimizing its own cost, neighbor j increases the deviation from the desired separation to agent i, this can also result in the right hand side being larger than zero, and Assumption 3.7 no longer holds. As the trajectory of neighboring agents is not constrained to fulfill (3.66), Assumption 3.7 can in general not be expected to hold true. This breaks the line of reasoning in Theorem 3.8, as (3.71) is no longer satisfied.

3.5 A Stability Analysis of Centralized Predictive Flocking Using Set Stability

The detailed discussions of the general challenges in the MPF analysis and the problems with the approaches in the related literature show that the analysis of MPF has not been addressed properly. To address it, this chapter presents a different approach to the analysis following the line of reasoning presented in Section 2.4.2. To account for the difficulties in the MPF analysis explained in Section 3.2, a centralized setting with static communication

topology is considered. Additionally, the stability analysis is based on an MPC scheme with terminal equality constraints.

3.5.1 Stability of Sets of Equilibria

Since, in MPF, the desired terminal state is not defined by a single equilibrium but rather a set of equilibria, the assumptions made in Section 2.4.2 must be modified to account for set stability. For the following definitions assume the equilibrium input to be zero. Let $\mathcal{A}$ denote the set of equilibria. Note that the general stability results in Section 2.2.4 are already stated in terms of sets.

Assumption 3.9. $\mathcal{A} \subseteq \mathbb{X}$ is closed.

Assumption 3.10. It holds for all $x \in \mathcal{A}$ that $f(x,0) \in \mathcal{A}$, $\ell(x,0) = 0$, $V_f(x) = 0$.

Assumption 3.11. The sets $\mathbb{X}$ and $\mathbb{X}_f$ are closed with $\mathcal{A} \subseteq \mathbb{X}_f \subseteq \mathbb{X}$. The set $\mathbb{U}$ is compact with $0 \in \mathbb{U}$.

Assumption 3.12. There exist functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that

$$\alpha_1(|x|_{\mathcal{A}}) \leq V^*(x) \leq \alpha_2(|x|_{\mathcal{A}}), \quad \forall x \in \mathbb{X}_M.$$

Assumption 3.13. For all $x \in \mathbb{X}_f$, there exists a $u \in \mathbb{U}$ satisfying $f(x,u) \in \mathbb{X}_f$ and

$$V_f(f(x,u)) - V_f(x) \leq -\ell(x,u). \tag{3.73}$$

Remark. In the analysis presented in Rawlings et al. (2020, Chapter 2), (2.61) and (2.62) are required to establish the lower bounds on the value function for Definition 2.4. In this section, the assumption regarding the existence of a lower bound is directly incorporated in Assumption 3.12, no longer requiring (2.61) and (2.62) in Assumption 3.13.

Note that Assumption 3.10 implies positive invariance of set $\mathcal{A}$ under the zero control law, and recall that $\mathbb{X}_M$ in Assumption 3.12 denotes the region of attraction. With these assumptions, the stability theorem for sets of equilibria can be stated similarly to Rawlings et al. (2020, Theorem 2.19).

Theorem 3.9. *Suppose Assumptions 2.1, 3.9, 3.10, 3.11, 3.12, and 3.13 are satisfied. Then, $\mathcal{A}$ is asymptotically stable in $\mathbb{X}_M$ under the control law* (2.51).

Proof. The proof follows the derivations in Rawlings et al. (2020, Chapter 2) and is only outlined here. In general, it is based on showing that $V^*(x)$ is a Lyapunov function according to Definition 2.4 and that Theorem 2.1 can be applied. Closedness of $\mathbb{X}$ and $\mathcal{A}$, positive invariance of $\mathcal{A}$, as well as local boundedness of f follow from Assumptions 2.1, 3.9, 3.10, and 3.11. Equations 2.19 and 2.20 follow directly from Assumption 3.12. Equation (2.21) can be established as follows: Let $U^*_{k|k}$ be the optimal solution at time k and consider the warm start sequence

$$\tilde{U}_{k+1|k} = \operatorname{col}\big(u^*_{k+1|k}, \ldots, u^*_{k+M-1|k}, \tilde{u}\big),$$

where $\tilde{u} \in \mathbb{U}$ is any admissible input. As $\tilde{U}_{k+1|k}$ is not the optimal solution for time $k+1$, it follows that

$$\begin{aligned} V(x_{k+1}, U^*_{k+1}) - V(x_k, U^*_k) &\leq V(x_{k+1}, \tilde{U}_{k+1|k}) - V(x_k, U^*_k) \\ &= -\ell(x_k, \kappa(x_k)) - V_f(x^*_{k+M|k}) + \ell(x^*_{k+M|k}, \tilde{u}) \\ &\quad + V_f(f(x^*_{k+M|k}, \tilde{u})), \end{aligned}$$

with all other terms canceling out by the choice of the warm start. With Assumption 3.13 it then follows that

$$V^*(x_{k+1}) - V^*(x_k) \leq -\ell(x_k, \kappa(x_k)),$$

establishing (2.21). Then, Theorem 2.1 can be applied. ■

In general, it is not trivial to show that Assumption 3.12 holds. However, if the value function happens to be continuous, the existence of upper and lower bounding $\mathcal{K}_\infty$ functions in Assumption 3.12 can also be established using the following lemma, which is an extension of Khalil (2002, Lemma 4.3).

Lemma 3.10. *Suppose $\mathcal{A} \subset \mathcal{X} \subset \mathbb{R}^n$ is closed. Let $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a continuous function with $V(x) = 0$, $\forall x \in \mathcal{A}$ and $V(x) > 0$ otherwise. Then, for some $r > 0$, there exist functions $\alpha_1, \alpha_2 \in \mathcal{K}$ defined on $[0, r]$ such that*

$$\alpha_1(|x|_\mathcal{A}) \leq V(x) \leq \alpha_2(|x|_\mathcal{A}), \ \forall x \in \mathcal{B}^c_r(\mathcal{A}).$$

If $\mathcal{X} = \mathbb{R}^n$, α_1 and α_2 will be defined on $[0, \infty)$ and the foregoing inequality will hold for all $x \in \mathbb{R}^n$. Moreover, if $V(x) \to \infty$ as $\|x\| \to \infty$, α_1 and α_2 can be chosen to belong to class $\mathcal{K}_\infty$.

Proof. The proof follows the same line of reasoning as the proof of Khalil (2002, Lemma 4.3). For $0 \leq s \leq r$, define $\psi_1(s)$ as

$$\psi_1(s) = \inf_{s \leq |x|_\mathcal{A} \leq r} V(x).$$

Then, $\psi_1(s)$ is continuous, positive definite, and increasing, but not necessarily strictly increasing. Let $\alpha_1(s)$ be a class $\mathcal{K}$ function such that $\alpha_1(s) \leq \delta\psi_1(s)$ with $0 < \delta < 1$. Then,

$$V(x) \geq \psi_1(|x|_\mathcal{A}) \geq \alpha_1(|x|_\mathcal{A})$$

for $|x|_\mathcal{A} \leq r$. Furthermore, for $0 \leq s \leq r$, define $\psi_2(s)$ as

$$\psi_2(s) = \sup_{|x|_\mathcal{A} \leq s} V(x).$$

Then, $\psi_2(s)$ is continuous, positive definite, and increasing, but not necessarily strictly increasing. Let $\alpha_2(s)$ be a class $\mathcal{K}$ function such that $\alpha_2(s) \geq \delta\psi_2(s)$ with $\delta > 1$. Then, for $|x|_\mathcal{A} \leq r$,

$$V(x) \leq \psi_2(|x|_\mathcal{A}) \leq \alpha_2(|x|_\mathcal{A}).$$

If $\mathcal{X} = \mathbb{R}^n$, the definitions of $\psi_1(s)$ and $\psi_2(s)$ are changed to

$$\psi_1(s) = \inf_{s \leq |x|_{\mathcal{A}}} V(x) \quad \text{and} \quad \psi_2(s) = \sup_{|x|_{\mathcal{A}} \leq s} V(x),$$

for $s \geq 0$. The functions $\psi_1(s)$ and $\psi_2(s)$ are again continuous, positive definite, and increasing, but not necessarily strictly increasing. By choosing $\alpha_1(s)$ and $\alpha_2(s)$ as before, one obtains

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}), \quad \forall x \in \mathbb{R}^n.$$

If $V(x) \to \infty$ as $\|x\| \to \infty$, $\psi_1(s)$ and $\psi_2(s)$ tend to infinity as $s \to \infty$. Hence, α_1 and α_2 can be chosen to belong to class $\mathcal{K}_\infty$. ■

Conditions for $V^*(x)$ being continuous are stated in Rawlings et al. (2020, Theorem 2.7). For example, this is the case when the agents' dynamics are linear with $\mathbb{X}$ closed and $\mathbb{U}$ compact (Rawlings et al. 2020, Theorem C.34).

3.5.2 A Stability Analysis for Centralized Model Predictive Flocking with Terminal Equality Constraints

To apply the above result to the CMPF analysis, consider the general CMPF as described in Section 3.1.1 with the stage cost (3.2),

$$\ell(x_{k+m|k}, u_{k+m|k}) = \|u_{k+m|k}\|^2_{R_u} + \|p_{k+m|k} - p_{\mathrm{r}}\|^2_{R_p} + \sum_{(i,j)\in\mathcal{E}(q_k)} \|\delta(q_{i,k+m|k}, q_{j,k+m|k})\|^2_{R_\delta},$$

$R_u, R_p, R_\delta \succ 0$, and terminal equality constraints ($V_f = 0$, $\kappa_M = 0$). Recall that $p_{\mathrm{r}} = 1_N \otimes p_{\mathrm{r}_i}$ is the collective reference velocity, with $p_{\mathrm{r}_i} \in \mathbb{R}^{n_{\mathrm{d}}}$ being the reference velocity for a single agent. The agents' dynamics are assumed to be linear,

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k),$$

with $x_i = [q_i \; p_i]^\top$. Input constraints are defined as in (3.4),

$$\mathcal{U} = \left\{ U \in \mathbb{R}^{Mn_u} : \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes I_{Mn_u} U - \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix} \otimes 1_{Mn_u} \leq 0 \right\},$$

and no state constraints are imposed ($\mathcal{X} = \mathbb{R}^{Mn_x}$).

To avoid the general problem of time-varying communication topologies, the following assumption is made.

Assumption 3.14 (Static communication topology)**.** Given the initial state x_0, it holds that

$$\mathcal{G}\big(\mathcal{V}, \mathcal{E}(q_k)\big) = \mathcal{G}\big(\mathcal{V}, \mathcal{E}(q_0)\big), \quad \forall k \geq 0.$$

Under this assumption, the set of desired α-lattices can be characterized as

$$\mathcal{A}(x_0) = \mathcal{A}_0 = \{x \in \mathbb{X} : \ \|q_{ij}\| = d \ \forall (i,j) \in \mathcal{E}(q_0), \ \|q_{ij}\| \geq r \ \forall (i,j) \notin \mathcal{E}(q_0), \ p = p_{\mathrm{r}}\}, \tag{3.74}$$

and the terminal set becomes $\mathbb{X}_f = \mathcal{A}_0$. With cost function

$$V(x_k, U_k) = \sum_{m=0}^{M-1} \ell(x_{k+m|k}, u_{k+m|k})$$

the resulting CMPF problem is formulated as follows:

Problem 3.5.

$$\begin{aligned} V^*(x(k)) = & \min_{U(k)} V(x(k), U(k)) \\ & \text{s.t. } x(k+m) = \phi(m; x(k), U(k)), \quad m \in \mathbb{I}_{0:M}, \\ & \quad X(k) \in \mathcal{X}, \\ & \quad U(k) \in \mathcal{U}, \\ & \quad x(k+M) \in \mathcal{A}_0. \end{aligned}$$

Furthermore, the following assumption is made to ensure that the problem is initially feasible and results in a connected α-lattice.

Assumption 3.15. $\mathcal{A}_0$ and $\mathbb{X}_M$ are nonempty, and $\mathcal{G}\big(\mathcal{V}, \mathcal{E}(q_0)\big)$ is connected.

The following lemma establishes that $\mathcal{A}_0$ is closed.

Lemma 3.11. *Suppose Assumptions 3.10, 3.11, and 3.15 hold. Then, the set $\mathcal{A}_0$ is closed and positive invariant for $x_{k+1} = f(x_k, \kappa(x_k))$ under control law (2.51).*

Proof. Positive invariance follows from Assumption 3.10. With $V(x,0) = 0, \forall x \in \mathcal{A}_0$, and $V(x,0) > 0, \forall x \notin \mathcal{A}_0$, the optimal input sequence for $x \in \mathcal{A}_0$ is given by $U_k^* = 0$, resulting in $V(x_k, U_k^*) = 0, \forall x_k \in \mathcal{A}_0$. Consequently if $x_k \in \mathcal{A}_0 \implies \phi(k'; x_k) \in \mathcal{A}_0$, $\forall k' \geq k$. To show that $\mathcal{A}_0$ is closed, consider the set of unconstrained α-lattices

$$\bar{\mathcal{A}}_0 = \{x \in \mathbb{R}^{n_x} : \ \|q_{ij}\| = d \ \forall (i,j) \in \mathcal{E}(q_0), \ \|q_{ij}\| \geq r \ \forall (i,j) \notin \mathcal{E}(q_0), \ p = p_{\mathrm{r}}\}.$$

Let $\bar{\mathcal{A}}_0^i \subset \bar{\mathcal{A}}_0$ denote the set of α-lattices relative to agent i given by the compact set

$$\bar{\mathcal{A}}_0^i = \{x \in \mathbb{R}^{n_x} : q_i = 0, \|q_{ij}\| = d \forall (i,j) \in \mathcal{E}(q_0), \|q_{ij}\| \geq r \, \forall (i,j) \notin \mathcal{E}(q_0), \, p = p_{\mathrm{r}}\}.$$

Furthermore, define the closed set

$$\mathcal{Q}_i = \{x \in \mathbb{R}^{n_x} : q_j = 0 \, \forall j \in \mathcal{V} \backslash \{i\}, p = 0\}. \tag{3.75}$$

The set of $\bar{\mathcal{A}}_0$ is then given by

$$\bar{\mathcal{A}}_0 = \bar{\mathcal{A}}_0^i \oplus \mathcal{Q}_i,$$

where $\oplus$ denotes the Minkowski sum. As $\bar{\mathcal{A}}_0^i$ is compact and $\mathcal{Q}_i$ is closed, this establishes that $\bar{\mathcal{A}}_0$ is closed (Molčanov 2017). With $\mathcal{A}_0 = \bar{\mathcal{A}}_0 \cap \mathbb{X}$, it follows that $\mathcal{A}_0$ is closed since both $\bar{\mathcal{A}}_0$ and $\mathbb{X}$ are closed (Molčanov 2017). ■

The stability theorem for the CMPF problem with terminal equality constraints can then be stated as follows:

Theorem 3.12 (CMPF stability)**.** *Suppose that Assumptions 3.14 and 3.15 hold. Then, for $x(0) \in \mathbb{X}_M$, Problem 3.5 is recursively feasible and the agents asymptotically converge towards an α-lattice with $p = p_\mathrm{r}$ under the control law* (2.51)*.*

Proof. The proof involves two parts. The first part addresses the recursive feasibility and the second part the convergence towards an α-lattice with $p = p_\mathrm{r}$.

Note that Assumptions 2.1, 3.10, and 3.11 are fulfilled by the definitions of the respective functions and sets. With $\mathbb{X}_M$ being nonempty, there exists a solution to Problem 3.5 at time $k = 0$. For the recursive feasibility, note that the warm start

$$\tilde{U}_{k+1|k} = \mathrm{col}\big(u^*_{k+1|k}, \dots, u^*_{k+M-1|k}, 0\big)$$

is always a feasible control sequence $\forall k > 0$, as it satisfies all the constraints. This establishes recursive feasibility.

For the convergence towards an α-lattice with $p = p_\mathrm{r}$, note that, by the definition of $\mathcal{A}_0$, this statement is equivalent to asymptotic stability of the set $\mathcal{A}_0$. Hence, proving this part requires showing that Assumptions 2.1, 3.9, 3.10, 3.11, 3.12, and 3.13 are satisfied so that Theorem 3.9 can be applied. Assumption 3.9 then follows from Lemma 3.11. Since f is linear and Assumptions 2.1 and 3.10 hold, by Rawlings et al. (2020, Theorem C.34), the value function $V^*(x)$ is continuous. Hence, by Lemma 3.10, Assumption 3.12 holds. Together with (3.73) being trivially satisfied for the terminal equality constraints, this shows that Assumption 3.13 also holds. ■

Remark. One way of ensuring that Assumption 3.14 is satisfied is to initialize the agents "close enough" to the set $\mathcal{A}_0$ such that no changes in the communication graph happen. Alternatively, the constraint $\|q_{ij}\| \geq r \ \forall (i,j) \notin \mathcal{E}(q_0)$ could be enforced via additional state constraints.

4 Model Predictive Flocking with Asymmetric Interaction Forces

Apart from the theoretical guarantees for model predictive flocking (MPF) discussed in Chapter 3, the performance, adaptability, and modularity of MPF frameworks play an important role in practical applications. When investigating the approaches taken in the related literature, there is still room for improvement with respect to these aspects.

In most of the existing literature, the predictive flocking behavior is modeled by penalizing the deviation from the desired separation, in the cost function of the optimization problem (Hu et al. 2018, 2017; Zhan and Li 2011a,b, 2013; Zhang et al. 2015; Zhou and Li 2017). One property resulting from this design are equally strong attractive and repulsive forces between agents when the deviation from the desired separation is different from zero. However, there are scenarios where this characteristic is undesirable. Larger repulsive forces can be desirable to reduce the risk of collisions between agents without affecting the attractive behavior. Similarly, there are scenarios where splitting of the swarm is advantageous, for example to smoothly avoid obstacles. For such scenarios, smaller attractive forces are desirable. This motivates the need for an MPF framework that allows for independent tuning of attractive and repulsive inter-agent behavior. Here, this approach is referred to as model predictive flocking with asymmetric interaction forces or asymmetric model predictive flocking (AMPF).

Furthermore, when investigating the scope of existing MPF algorithms, the majority of the existing frameworks only consider free flocking (Hu et al. 2018, 2017; Zhan and Li 2011a,b, 2013; Zhang et al. 2015; Zhou and Li 2017) and restrict the field of application to agents with double-integrator dynamics or even single-integrator dynamics (Zhan and Li 2013). One exception to this is the framework presented in Rochefort et al. (2012), where the authors present a distributed model predictive flocking (DMPF) scheme for nonholonomic ground vehicles. On the other hand, there exist two frameworks that also address obstacle avoidance and group objectives (Huang et al. 2019; Rochefort et al. 2012). However, in both of these references, the capabilities of the predictive control framework are not utilized to its full extend. In Rochefort et al. (2012), the authors consider a discrete decision space with a limited number of admissible control inputs that are also constrained to be constant over the prediction horizon, and in Huang et al. (2019), group objectives are handled in a suboptimal manner by adding a reference tracking term to the solution of the MPF problem similar to non-predictive algorithms in Olfati-Saber (2006).

This chapter addresses the aforementioned open topic by introducing a framework for asymmetric model predictive flocking (AMPF) with obstacle avoidance and group objectives

without restricting the agents to double-integrator dynamics. The desired AMPF behavior is modeled by considering state-dependent weighting factors in the cost function, allowing for a straightforward extension of the general MPF framework introduced in Section 3.1. After a general discussion of asymmetric interaction forces in Section 4.1, an asymmetric distributed model predictive flocking (ADMPF) framework is introduced in Section 4.2. Section 4.3 then demonstrates how the general analysis for centralized model predictive flocking that is presented in Section 3.5 can be extended to asymmetric centralized model predictive flocking (ACMPF). Moreover, a parameter study investigating how different assumptions on neighbor trajectories can affect the performance and prediction mismatch in ADMPF is presented in Section 4.4. Finally, in Section 4.5, the performance of the proposed ADMPF algorithm is compared to that of algorithms in the related literature. The concepts introduced in this chapter follow the ideas published in Hastedt and Werner (2023a,b).

4.1 Asymmetric Interaction Forces

In the general MPF problem introduced in Section 3.1, the flocking objectives are modeled by penalizing the deviations from the desired separation in the cost function by including the term

$$\ell_i^\delta = \sum_{j \in \mathcal{N}_i} r_\delta \|\delta(q_i, q_j)\|^2. \tag{4.1}$$

Depending on the sign of δ_{ij}, the forces between agents are either repulsive or attractive. Repulsive inter-agent forces correspond to deviations smaller than zero ($\delta_{ij} < 0$), and attractive inter-agent forces correspond to deviations larger than zero ($\delta_{ij} > 0$). Considering the stage cost component in (4.1), these forces are of equal strength, since positive and negative deviations from the desired separation are scaled with the same r_δ, yielding a framework with symmetric interaction forces. As motivated above, this behavior is not always desirable, highlighting the need for a framework with asymmetric interaction forces that allows for independent tuning of attractive and repulsive inter-agent forces. Note that in the non-predictive flocking described in Section 2.3, this behavior is implemented via the shape of the action function and the inter-agent potential. The above discussion is visualized in Fig. 4.1. In the non-predictive framework (blue), the attractive forces are limited by the cutoff function used in the design of the action function (see Section 2.3). For MPF with symmetric interaction forces (black), a trade-off has to be made between the desire for large repulsive forces and weak attractive forces. With the MPF framework with asymmetric interaction forces proposed in this chapter (orange), this trade-off is no longer necessary since attractive and repulsive forces can be tuned independently to achieve the desired behavior.

To penalize positive and negative deviations from the desired distance independently, the constant weight r_δ in (4.1) is replaced by a state-dependent weight $r_\delta(q_i, q_j)$, resulting in the stage cost term

$$\ell_i^\delta(x_i, x_{\bar{i}}) = \sum_{j \in \mathcal{N}_i} r_\delta(q_i, q_j) \|\delta(q_{ij})\|^2.$$

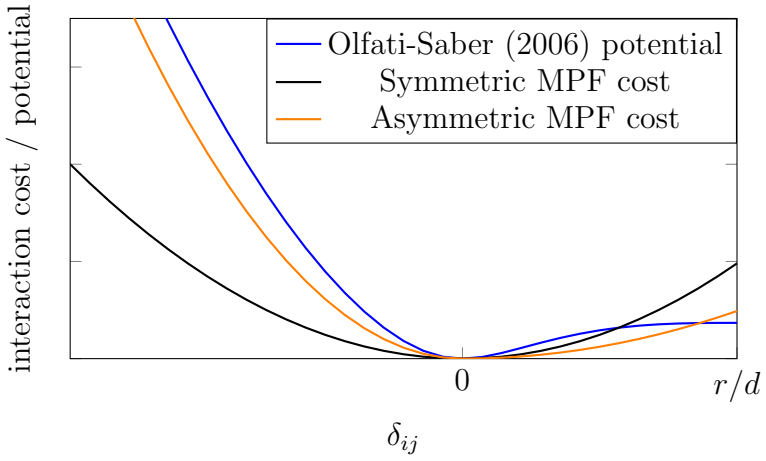


Figure 4.1: Comparison of interaction costs and potentials for different frameworks.

There are several ways to design $r_\delta(q_i, q_j)$. By choosing

$$r_\delta(q_i, q_j) = \begin{cases} r_\delta^+ & \text{if } \delta_{ij} > 0, \\ r_\delta^- & \text{if } \delta_{ij} \leq 0, \end{cases} \tag{4.2}$$

with $r_\delta^+, r_\delta^- \geq 0$, attractive and repulsive forces can be tuned independently. However, this formulation of $r_\delta(q_i, q_j)$ is not differentiable at $\delta_{ij} = 0$. As many optimization algorithms require the cost function to be continuously differentiable (Nocedal and Wright 2006), an alternative parameterization

$$r_\delta(q_i, q_j) = \frac{r_\delta^+ - r_\delta^-}{2} \tanh\left(\frac{\delta_{ij}}{\epsilon_\alpha}\right) + \frac{r_\delta^+ + r_\delta^-}{2} \tag{4.3}$$

is proposed, which smoothly varies between r_δ^+ and r_δ^- at the price of introducing additional nonlinearities in the cost. Here $\epsilon_\alpha > 0$ is a shaping parameter determining how quickly $r_\delta(\delta_{ij})$ transitions between r_δ^+ and r_δ^-. As $\lim_{\epsilon_\alpha \to 0} r_\delta(\delta_{ij})$, the definition in (4.3) is equal to the one in (4.2). Exemplary shapes of $r_\delta(q_i, q_j)$ for different values of ϵ_α are depicted in Fig. 4.2.

4.2 Problem Statement

In the following, the general problem for asymmetric distributed model predictive flocking (ADMPF) is derived. First, two approaches for estimating the trajectories of neighboring agents are discussed (Section 4.2.1). Subsequently, the ADMPF problem is introduced. Following the convention in Olfati-Saber (2006) the agents of the swarm, virtual agents generated by obstacles, and virtual leaders are referred to as α-,β-, and γ-agents, respectively. Similar to Section 2.3, free flocking is discussed first (α-ADMPF). This problem is then extended to ADMPF with group objectives (α, γ-ADMPF) and ADMPF with obstacle avoidance (α, β, γ-ADMPF). If not stated otherwise, consider the agent's state vectors to be given by $x_i = [q_i \; p_i]^\top$.

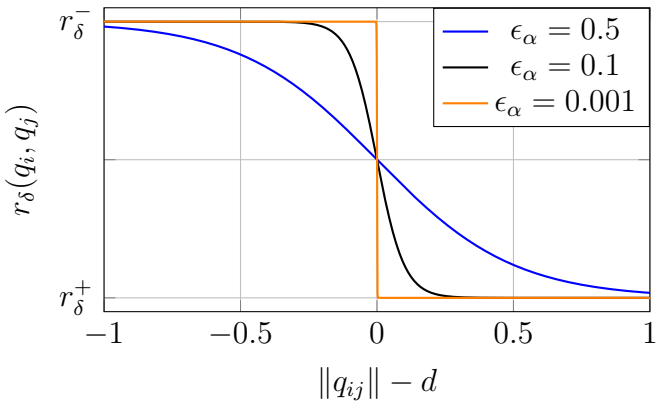


Figure 4.2: Shape of $r_\delta(\delta_{ij})$ in (4.3) for different values of ϵ_α.

4.2.1 Neighbor Trajectory Estimation

As explained in Section 3.2, one of the challenges in DMPF schemes is to estimate the behavior of neighboring agents. Following the discussion in Section 3.1.2, the behavior of neighboring agents is estimated based on an assumed input trajectory $\hat{U}_j$, resulting in the estimated state sequence

$$\hat{x}_j(k+m|k) = \phi_j(m; x_j(k), \hat{U}_j(k|k-1)), \quad \forall j \in \mathcal{N}_i(k),\ m \in \mathbb{I}_{0:M}. \tag{3.16}$$

In the literature on MPF, there are two common approaches for obtaining $\hat{U}_j$. The first approach is to assume a zero input sequence $\hat{U}_j = 0$ (Zhan and Li 2011b; Zhang et al. 2015; Zhou and Li 2017). The second approach is based on computing a warm start based on an exchange of the optimal control sequences of the previous step (Hu et al. 2018, 2017), resulting in

$$\hat{U}_j(k|k-1) = \operatorname{col}\big(u_j^*(k|k-1), \dots, u_j^*(k+M-2|k-1), 0\big), \quad \forall j \in \mathcal{N}_i(k). \tag{3.15}$$

This second approach is commonly used in combination with consistency constraints (see Section 3.4 and the references therein). In Section 4.4, the performance of these two approaches will be compared through simulation.

4.2.2 α-Asymmetric Distributed Model Predictive Flocking

To model the rules of cohesion, separation, and alignment (Reynolds 1987) in the ADMPF framework, consider the stage cost

$$\begin{aligned}\ell_i^\alpha(x_{i,k+m|k}, \hat{x}_{\bar{i},k+m|k}, u_{i,k+m|k}) = \|u_{i,k+m|k}\|_{R_u}^2 + \|p_{i,k+m|k} - \bar{p}_{i,k}^{\text{ave}}\|_{R_p}^2 \\ + \sum_{j \in \mathcal{N}_{i,k}} r_\delta(q_{i,k+m|k}, \hat{q}_{j,k+m|k}) \|\delta(q_{i,k+m|k}, \hat{q}_{j,k+m|k})\|^2,\end{aligned} \tag{4.4}$$

with $R_u, R_p \succ 0$ and r_δ according to (4.3) with $r_\delta^+, r_\delta^- > 0$. Here, $\bar{p}_{i,k}^{\text{ave}}$ is the average velocity of the i^{th} subsystem given by

$$\bar{p}_{i,k}^{\text{ave}} = \frac{1}{|\mathcal{N}_{i,k}| + 1}\big(p_{i,k} + \sum_{j \in \mathcal{N}_{i,k}} \hat{p}_{j,k}\big). \tag{4.5}$$

The same $\bar{p}_i^{\text{ave}}$ is considered for every step of the prediction horizon.

The free flocking ADMPF problem can then be stated as follows:

Problem 4.1 (α-ADMPF)**.**

$$\begin{aligned} V_i^*(x_{i,k}, \hat{X}_{\bar{i},k}^M) = \min_{U_{i,k}} \; & V_i^\alpha\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) \\ \text{s.t. } & x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ & (X_{i,k}^M, \hat{X}_{\bar{i},k}^M) \in \bar{\mathbb{X}}_i^{M+1}, \\ & U_{i,k} \in \mathcal{U}_i, \end{aligned}$$

with cost function

$$V_i^\alpha\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) = \sum_{m=0}^{M-1} \ell_i^\alpha(x_{i,k+m|k}, \hat{x}_{\bar{i},k+m|k}, u_{i,k+m|k}) + \ell_i^\alpha(x_{i,k+M|k}, \hat{x}_{\bar{i},k+M|k}, 0) \tag{4.6}$$

and constraint sets as defined in Section 2.4. Here, the last term in (4.6) can be considered the terminal cost, and the terminal constraint is included in the state constraints with $\mathbb{X}_{i,f} = \bar{\mathbb{X}}_i$.

4.2.3 α-γ-Asymmetric Distributed Model Predictive Flocking

In contrast to the reference tracking considered in Huang et al. (2019), in the presented ADMPF framework, reference tracking is directly included in the cost function. Consider a static reference (γ-agent) with $x_{\mathrm{r}_i} = [q_{\mathrm{r}_i}\ p_{\mathrm{r}_i}]^\top \in \mathbb{R}^{2n_\mathrm{d}}$. To avoid large navigational penalties when agents are far away from the γ-agent, each agent generates its own virtual reference $x_i^\mathrm{r}(k) = [q_i^\mathrm{r}(k)\ p_{\mathrm{r}_i}]^\top \in \mathbb{R}^{2n_\mathrm{d}}$ with

$$q_i^\mathrm{r}(k) = \begin{cases} q_{i,\mathrm{r}} & \text{if } \|q_i(k) - q_{\mathrm{r}_i}\| \le d_\mathrm{r}, \\ q_i(k) + d_\mathrm{r} \dfrac{q_i(k) - q_{\mathrm{r}_i}}{\|q_i(k) - q_{\mathrm{r}_i}\|}, & \text{otherwise.} \end{cases} \tag{4.7}$$

Here, $d_\mathrm{r} > 0$ denotes a look-ahead distance for the virtual reference. If q_{r_i} is further than d_r from q_i, agent i will generate its virtual reference on the line from q_i to q_{r_i} at a distance d_r from agent i. Consider the tracking stage cost

$$\ell_i^\gamma(x_i, x_i^\mathrm{r}) = \|x_i - x_i^\mathrm{r}\|_{R_\gamma}^2. \tag{4.8}$$

The cost associated with reference tracking then becomes

$$V_i^\gamma\big(x_{i,k}, x_{i,k}^\mathrm{r}, U_{i,k}\big) = \sum_{m=0}^{M} \ell_i^\gamma(x_{i,k+m|k}, x_{i,k}^\mathrm{r}), \tag{4.9}$$

leading to the following statement of the α-γ-ADMPF problem:

Problem 4.2 (α-γ-ADMPF)**.**

$$\begin{aligned} V_i^*(x_{i,k}, \hat{X}_{\bar{i},k}^M, x_{i,k}^{\mathrm{r}}) = &\min_{U_{i,k}} V_i^\alpha\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) + V_i^\gamma\big(x_{i,k}, x_{i,k}^{\mathrm{r}}, U_{i,k}\big) \\ &\text{s.t. } x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ &\quad (X_{i,k}^M, \hat{X}_{\bar{i},k}^M) \in \bar{\mathbb{X}}_i^{M+1}, \\ &\quad U_{i,k} \in \mathcal{U}_i. \end{aligned}$$

4.2.4 α-β-γ-Asymmetric Distributed Model Predictive Flocking

To incorporate obstacle avoidance in the ADMPF scheme, recall the framework for obstacle avoidance in non-predictive flocking described in Section 2.3. Each of the N^β static, spherical obstacles is characterized by its center state $x_o^\beta = \mathrm{col}(q_o^\beta, 0) \in \mathbb{R}^{2n_\mathrm{d}}$ and radius $r_o^\beta > 0$, $o \in \mathbb{I}_{1:N^\beta}$. Each agent i then considers a virtual β-agent on the boundary of the obstacle with its position $q_o^{\beta_i}(k)$ calculated according to (2.36). Given an obstacle detection range r_o, according to (2.37), the set of obstacles detected by agent i is denoted as $\mathcal{N}_{i,k}^\beta = \mathcal{N}_i^\beta(q_{i,k})$. Similar to the definition of the neighborhood state vector of agent i, let

$$q_{\bar{o}}^{\beta_i} = \mathrm{col}\big(q_1^{\beta_i}, \ldots, q_{|\mathcal{N}_i^\beta|}^{\beta_i}\big)$$

denote the position vector of the obstacles detected by agent i. Furthermore, define the deviation from the desired obstacle separation d_o as

$$\delta^\beta(q_i, q_o^{\beta_i}) = \delta_{io}^\beta = \|q_o^{\beta_i} - q_i\| - d_o. \tag{4.10}$$

One of the main disadvantages of MPF with symmetric interaction forces, particularly in the case of obstacle avoidance, is that the obstacle detection range r_o has to be set equal to the desired obstacle separation d_o in order to avoid attractive forces between obstacles and agents. However, this means that agents cannot preemptively avoid obstacles but just reactively. In ADMPF, this issue can be solved analogously to the implementation of the asymmetric interaction forces in (4.2) and (4.3). Consider the obstacle avoidance stage cost

$$\ell_i^\beta(x_{i,k+m|k}, q_{o,k|k}^{\beta_i}) = \sum_{o \in \mathcal{N}_{i,k}^\beta} r_\beta(q_{i,k+m|k}, q_{o,k|k}^{\beta_i}) \|\delta^\beta(q_{i,k+m|k}, q_{o,k|k}^{\beta_i})\|^2.$$

By selecting

$$r_\beta(q_{i,k+m|k}, q_{o,k|k}^{\beta_i}) = \begin{cases} 0 & \text{if } \delta_{io}^\beta > 0, \\ r_\beta^- & \text{if } \delta_{io}^\beta \leq 0, \end{cases} \tag{4.11}$$

with $r_\beta^- > 0$, the attractive force can be set to zero. Analogous to (4.3), the continuous differentiable weight is given by

$$r_\beta(q_{i,k+m|k}, q_{o,k|k}^{\beta_i}) = -\frac{r_\beta^-}{2} \tanh\left(\frac{\delta_{io}^\beta}{\epsilon_\beta}\right) + \frac{r_\beta^-}{2}, \tag{4.12}$$

with shaping parameter $\epsilon_\beta > 0$. Note that this parameterization of r_β results in small attractive forces due to r_β asymptotically approaching zero. However, for ϵ_β small enough, the attractive forces can be neglected.

Considering the cost associated with obstacle avoidance as

$$V_i^\beta\big(x_{i,k}, q_{\bar{o},k}^{\beta_i}, U_{i,k}\big) = \sum_{m=0}^{M} \ell_i^\beta(x_{i,k+m|k}, q_{\bar{o},k|k}^{\beta_i}), \tag{4.13}$$

the ADMPF problem with obstacle avoidance and group objectives can be stated.

Problem 4.3 (α-β-γ-ADMPF)**.**

$$\begin{aligned} V_i^*(x_{i,k}, \hat{X}_{\bar{i},k}^M, x_{i,k}^{\mathrm{r}}, q_{\bar{o},k}^{\beta_i}) = \min_{U_{i,k}} \; & V_i^\alpha\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) + V_i^\beta\big(x_{i,k}, q_{\bar{o},k}^{\beta_i}, U_{i,k}\big) + V_i^\gamma\big(x_{i,k}, x_{i,k}^{\mathrm{r}}, U_{i,k}\big) \\ \text{s.t. } & x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ & (X_{i,k}^M, \hat{X}_{\bar{i},k}^M) \in \bar{\mathbb{X}}_i^{M+1}, \\ & U_{i,k} \in \mathcal{U}_i. \end{aligned}$$

4.2.5 Numerical Optimization

To solve the resulting nonlinear optimization problem, sequential quadratic programming (SQP) is used (Nocedal and Wright 2006). Consider the nonlinear constrained optimization problem

$$\begin{aligned} \min_x \; & f(x) \\ \text{s.t. } & h(x) = 0, \\ & g(x) \leq 0, \end{aligned}$$

with optimization variable x. Here, the functions f, h, and g are assumed to be twice continuously differentiable. Given the current iterate x_κ, the minimizer of the quadratic subproblem

$$\begin{aligned} \min_x \; & \nabla f^\top(x_\kappa)x + x^\top \nabla^2 f(x_\kappa)x \\ \text{s.t.} & \nabla h^\top(x_\kappa)x + h(x_\kappa) = 0, \\ & \nabla g^\top(x_\kappa)x + g(x_\kappa) \leq 0, \end{aligned}$$

is used to define the next iterate $x_{\kappa+1}$. Here, $\nabla^2 f$ refers to the Hessian of f with respect to x. This procedure is then repeated until convergence. To reduce computational complexity, the SQP algorithm can be terminated after a specified number of N_{SQP} iterations (Cannon 2004). Setting N_{SQP} to 1 corresponds to solving a quadratic approximation of the nonlinear MPF problem similar to the approaches in Zhan and Li (2011a,b, 2013), Zhang et al. (2015), and Zhou and Li (2017). For larger N_{SQP}, the obtained solution gets closer to the optimal solution of the nonlinear optimization problem. The effect of N_{SQP} on the performance of the proposed algorithm will be investigated empirically in the parameter studies in Section 4.4.

Remark. As discussed earlier in Section 3.2.1, the cost functions derived in this chapter are not continuously differentiable for $q_i = q_j$ and $q_i = q_o^{\beta_i}$, which corresponds to inter-agent and agent-obstacle collisions. This issue can be resolved by replacing δ_{ij} and δ_{ij}^{β} with continuously differentiable approximations or by stating the assumption that no inter-agent and agent-obstacle collisions occur by appropriately choosing the weights in the cost functions. Here, the second approach is taken. The above derivations also hold when considering continuously differentiable δ_{ij} and δ_{ij}^{β}, for example by replacing the involved norm by the σ-map defined in (2.24).

4.3 Stability Analysis for Asymmetric Centralized Model Predictive Flocking

In this section, the centralized model predictive flocking (CMPF) analysis presented in Section 3.5 is extended to asymmetric centralized model predictive flocking (ACMPF). As in Section 3.5, the agents' dynamics are assumed to be linear, input constraints are defined as in (3.4),

$$\mathcal{U} = \left\{ U \in \mathbb{R}^{Mn_u} : \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes I_{Mn_u} U - \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix} \otimes 1_{Mn_u} \leq 0 \right\},$$

and no state constraints are imposed ($\mathcal{X} = \mathbb{R}^{Mn_x}$). Consider the cost function

$$V(x_k, U_k) = \sum_{m=0}^{M-1} \ell(x_{k+m|k}, u_{k+m|k}),$$

the stage cost

$$\begin{aligned} \ell(x_{k+m|k}, u_{k+m|k}) =& \|u_{k+m|k}\|_{R_u}^2 + \|p_{k+m|k} - p_{\mathrm{r}}\|_{R_p}^2 \\ &+ \sum_{(i,j)\in\mathcal{E}(q_0)} r_\delta(q_{i,k+m|k}, q_{j,k+m|k}) \|\delta(q_{i,k+m|k}, q_{j,k+m|k})\|^2, \end{aligned}$$

and $r_\delta(q_i, q_j)$ as defined in (4.3), with $R_u, R_p, r_\delta^+, r_\delta^- \succ 0$. The collective reference velocity is assumed to be constant with $p_{\mathrm{r}} = 1_N \otimes p_{\mathrm{r}_i}$, $p_{\mathrm{r}_i} \in \mathbb{R}^{n_{\mathrm{d}}}$. It is important to note that the introduction of the asymmetric interaction forces does not alter the set of equilibria $\mathcal{A}_0$ in (3.74). As in Section 3.5, consider the terminal set $V_f = 0$ and the terminal controller $\kappa_M = 0$. The ACMPF problem can then be stated as follows:

Problem 4.4 (ACMPF problem)**.**

$$\begin{aligned} V^*(x(k)) = &\min_{U(k)} V(x(k), U(k)) \\ &\text{s.t. } x(k+m) = \phi(m; x(k), U(k)), \quad m \in \mathbb{I}_{0:M}, \\ &\quad X(k) \in \mathcal{X}, \\ &\quad U(k) \in \mathcal{U}, \\ &\quad x(k+M) \in \mathcal{A}_0. \end{aligned}$$

The stability theorem corresponding to Problem 4.4 is then stated analogously to Theorem 3.12.

Theorem 4.1 (ACMPF stability)**.** *Suppose that Assumptions 3.14 and 3.15 hold. Then, Problem 4.4 is recursively feasible and the agents asymptotically converge towards an α-lattice with $p = p_\mathrm{r}$ under the control law* (2.51)*.*

Proof. Considering the asymmetric interaction weight $r_\delta(\delta_{ij})$ as defined in (4.3), the stage cost remains continuous, satisfying Assumption 2.1. The proof then proceeds analogously to that of Theorem 3.12 by showing that Theorem 3.9 can be applied. ■

4.4 Parameter Studies

In this section, several simulation scenarios are presented to investigate the performance of different setups of the ADMPF problem. Mainly, two aspects will be investigated. The first aspect is how the estimation of the trajectories of neighboring agents affects the performance and prediction mismatch. As introduced in Section 4.2.1, the behavior of neighboring agents can be estimated either by assuming a zero input sequence $\hat{U}_j = 0$, denoted as NE_0, or by computing a warm start based on the optimal solution at the previous time step $\hat{U}_j = \hat{U}_j(U^*_{j,k-1})$, denoted as NE_w. Moreover, the prediction mismatch and performance are highly affected by the choice of the prediction horizon M. In centralized MPC, longer prediction horizons usually improve the performance as agents can incorporate more of their future knowledge in their control decision. However, due to the partial information available in DMPF, this may not hold here as the prediction of the neighboring agents' behavior becomes less accurate with an increasing prediction horizon.

The second aspect to be investigated is how the number of SQP iterations affects the performance. As explained in Section 4.2.5, at each step, the nonlinear and non-convex ADMPF problems are solved as a sequence of quadratic optimization problems. By limiting the number of iterations, this results in a trade-off between the convergence towards the optimal solution and the required computational resources. This trade-off is investigated in the second simulation study by running simulations with varying numbers of SQP iterations and comparing the resulting performances.

All simulations in this section are performed using Matlab and the WiMAS library for wireless multi-agent simulations (Hespe et al. 2024). Tuning parameters are provided in Appendix A.1.

4.4.1 Trajectory Estimation and Prediction Horizon

In the first parameter study, the performance and resulting prediction mismatch are compared for the different methods of estimating the trajectories of neighboring agents - zero input (NE_0) and warm start (NE_w) - for varying prediction horizons.

Setup

Consider a swarm of $N = 20$ agents governed by the discrete-time double-integrator dynamics in (2.10), moving in a two-dimensional space $n_\mathrm{d} = 2$. Agents are initialized in an α-lattice formation at the origin (green) and are tasked with reaching a static reference point (orange) while avoiding two obstacles (black) on the way. The considered scenario is depicted in Fig. 4.3. In the following, the results are compared for the prediction horizons $M = \{2, 4, 6, 8, 10\}$, considering the average performance over the scenarios with obstacle radii $r_o^\beta = \{6, 6.5, 7, 7.5\}$. All of the scenarios were simulated with one SQP iteration ($N_\mathrm{SQP} = 1$) without state constraints and input constraints in the form of (3.7),

$$\mathcal{U}_i = \left\{ U_i \in \mathbb{R}^{Mn_{u,i}} : \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes I_{Mn_{u,i}} U_i - \begin{bmatrix} u_\mathrm{max} \\ u_\mathrm{min} \end{bmatrix} \otimes 1_{Mn_{u,i}} \leq 0 \right\}.$$

Performance Measures

To compare the performances, consider the average lattice irregularity

$$J_\alpha(k) = \frac{1}{|\mathcal{E}(q_k)|} \sum_{(i,j)\in\mathcal{E}(q_k)} \delta(q_{i,k}, q_{j,k})^2. \tag{4.14}$$

Furthermore, let

$$J_\alpha^\Sigma(t) = \frac{\tau}{t} \sum_{n=0}^{t/\tau} J_\alpha(n) \tag{4.15}$$

denote the integrated lattice irregularity up to time t. To determine the speed of convergence, the ± 0.02-band around $J_\alpha(k_f)$ is considered, where k_f is the final simulation step. The corresponding final simulation time is denoted as t_f. Analogously, the average

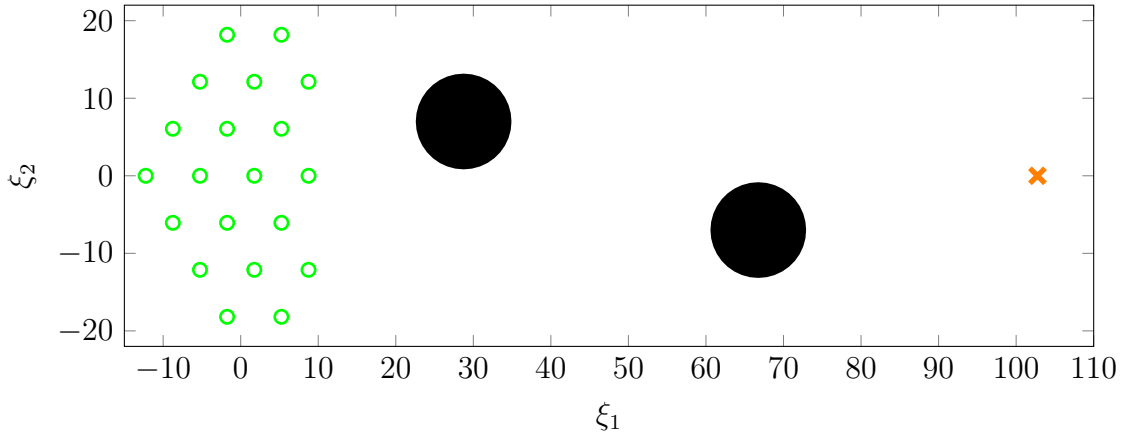


Figure 4.3: Scenario for prediction mismatch study with obstacle diameter $r_o^\beta = 6$. Initial states are indicated in green and the reference position in orange.

obstacle avoidance performance and integrated obstacle avoidance performance are defined as

$$J_\beta(k) = \sum_{i\in\mathcal{V}} \left(\frac{1}{|\mathcal{N}_{i,k}^\beta|} \sum_{o\in\mathcal{N}_{i,k}^\beta} \min\left(\delta^\beta(q_{i,k}, q_{o,k}^{\beta_i}), 0\right)^2 \right) \tag{4.16}$$

and

$$J_\beta^\Sigma(t) = \frac{\tau}{t} \sum_{n=0}^{t/\tau} J_\beta(n). \tag{4.17}$$

The required input energy is compared with respect to the average energy of the input signals of all agents given by

$$u_\mathrm{E} = \frac{\tau}{N} \sum_{n=0}^{t_f/\tau} \sum_{i\in\mathcal{V}} \|u_i(n)\|^2. \tag{4.18}$$

To compare the prediction mismatch, let

$$x_{\mathrm{m},ij}(k) = \|x_i^{i*}(k+M|k) - x_i^j(k+M|k)\|$$

denote the prediction mismatch at the final step of the prediction horizon. Here, x_i^{i*} is the optimal solution determined by agent i at time k and x_i^j denotes the assumption on agent i's state made by agent j. The average mismatch at step k is then given by

$$x_\mathrm{m}(k) = \frac{1}{|\mathcal{E}(q_k)|} \sum_{(i,j)\in\mathcal{E}(q_k)} x_{\mathrm{m},ij}(k)$$

and the average mismatch up to time t by

$$x_\mathrm{m}^\mathrm{ave}(t) = \frac{\tau}{t} \sum_{n=0}^{t/\tau} x_\mathrm{m}(k). \tag{4.19}$$

Results

The exemplary behavior of the agents with $r_o^\beta = 6$ and $M = 4$ is depicted in Fig. 4.4 for estimating the trajectories of neighboring agents based on the zero input sequence (NE_0). The prediction mismatch, performance in terms of integrated lattice irregularity, integrated obstacle avoidance performance, and input energy are summarized in Figures 4.5, 4.6, 4.7, and 4.8, respectively.

When estimating the trajectories of neighboring agents based on the optimal solution at the previous time step (NE_w), for $M > 4$, the agents are not able to complete the simulation successfully for any of the four obstacle diameters. In those scenarios, the agents are still able to clear the field of obstacles, however, they do not converge towards a static α-lattice but continue to move in the vicinity of the reference point. One explanation for the NE_w approach failing for larger horizons could be that, using this approach, the

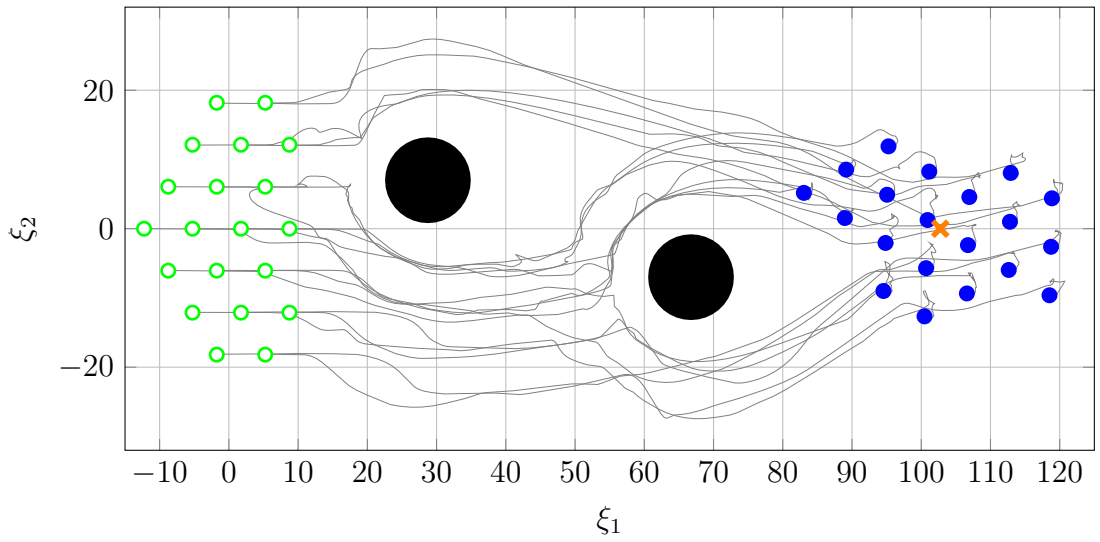


Figure 4.4: Exemplary agent trajectories for NE_0 implementation with $r_o^\beta = 6$ and $M = 4$. Positions at $t = t_f$ are indicated in blue.

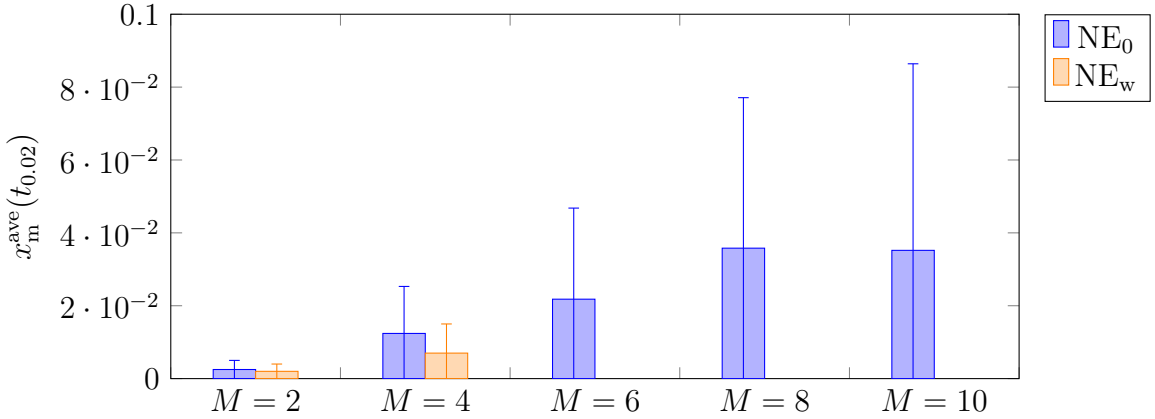


Figure 4.5: Prediction mismatch comparison for trajectory estimation and prediction horizon study averaged over varying obstacle diameters.

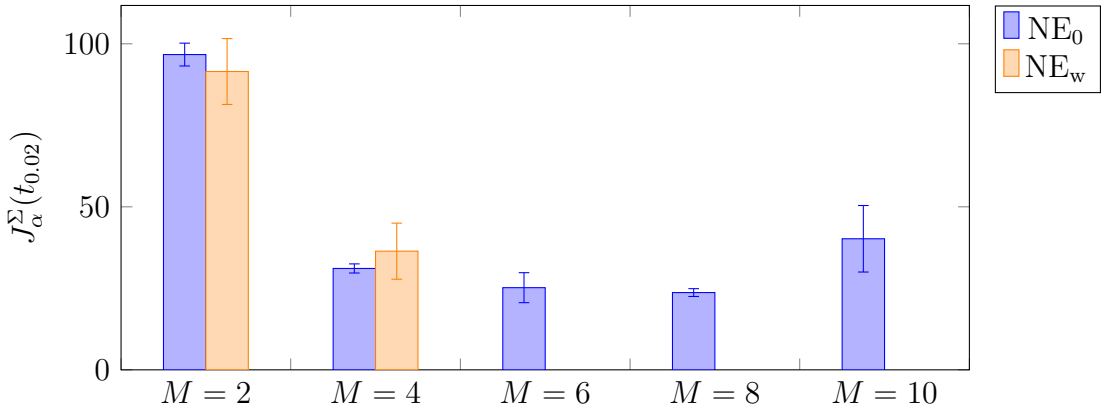


Figure 4.6: Integrated lattice irregularity comparison for trajectory estimation and prediction horizon study averaged over varying obstacle diameters.

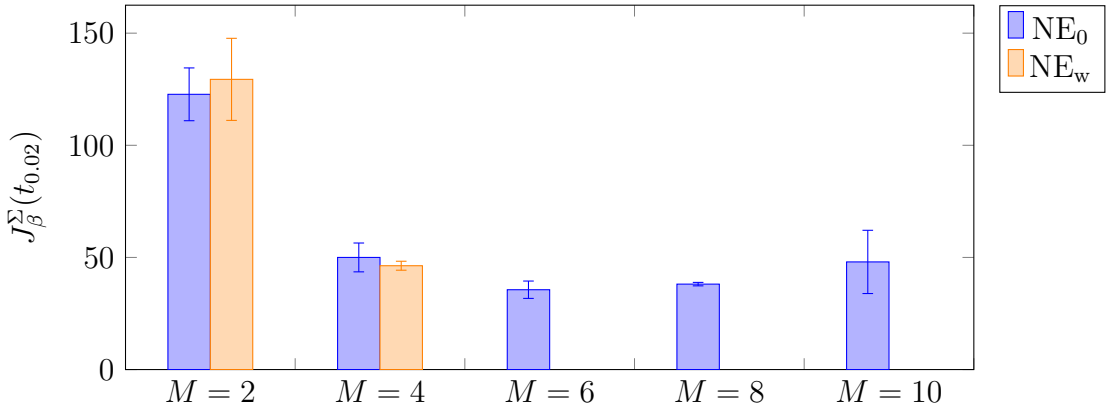


Figure 4.7: Integrated obstacle avoidance performances comparison for trajectory estimation and prediction horizon study averaged over varying obstacle diameters.

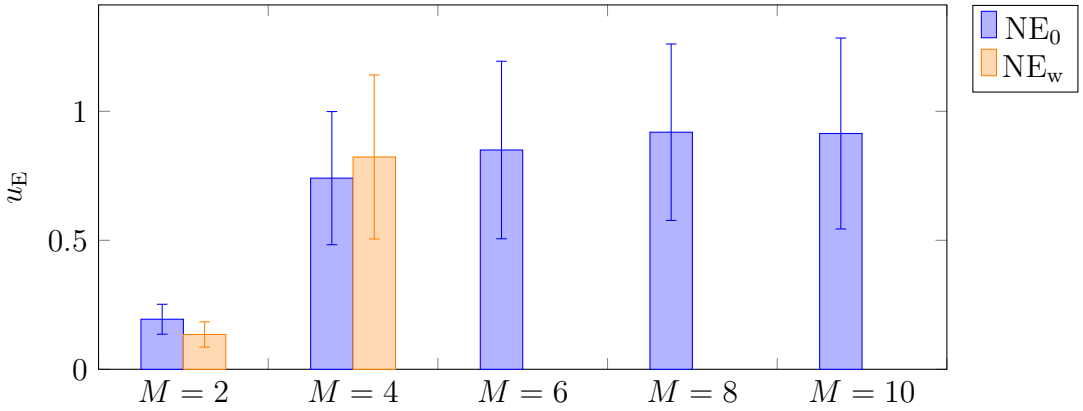
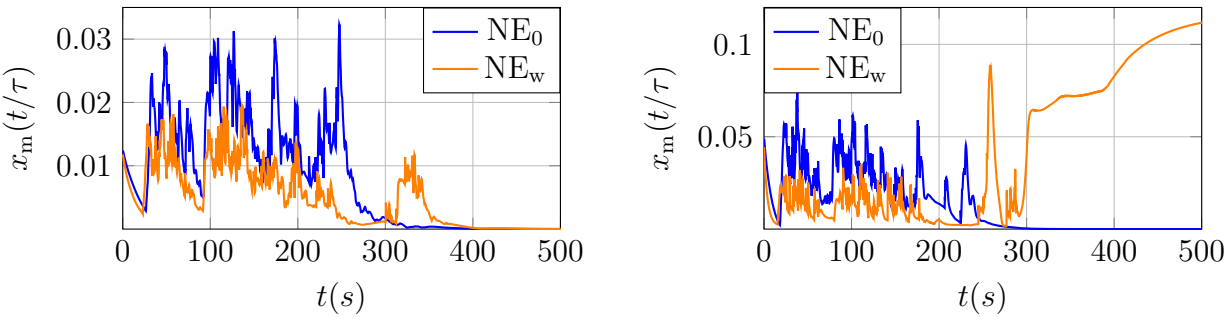


Figure 4.8: Input comparison for trajectory estimation and prediction horizon study averaged over varying obstacle diameters.

behavior of each agent is determined based on a specific neighbor trajectory that can change drastically between sampling instances when neighboring agents change their input sequences. On the other hand, the neighbor trajectory estimation with zero inputs (NE_0) can be considered assuming a "neutral" neighbor trajectory. For the first part of the scenario, the swarm is not significantly affected by this mismatch since the agents are mainly driven by the attractive force of the reference point. However, once they reach the target point, agents try to form a static α-lattice, which involves small movements in different directions to converge towards the desired formation. Depending on the behavior of an agent's neighbor, this can result in large deviations from the optimal solution at the previous step. This effect can be seen when plotting the prediction mismatch over the simulation time as depicted in Fig. 4.9. For $M = 4$, the mismatch converges to zero for both methods of estimating the neighbor trajectories. However, for NE_w with $M = 6$, the mismatch does not converge to zero. Instead, it increases for $t > 300\,\mathrm{s}$, which corresponds to the time where the first agents reach the reference point.

When comparing the prediction mismatch for the successfully completed simulations, average mismatch and its standard deviation increase with longer prediction horizons. Furthermore, comparing the integrated lattice irregularities and required input energy, for those scenarios completing successfully, the performance for NE_w and NE_0 are comparable. For $M = 2$, for both methods of trajectory estimation, the agents use very little input with large lattice and obstacle costs. This result can be explained by the contradicting objectives of the optimization problem. On the one hand, agents have to match the velocities of neighboring agents and the reference velocity. On the other hand, agents have to adjust their position to reach the reference point, avoid obstacles, and form an α-lattice. Due to the discretization of the double-integrator dynamics in (2.10), there is a delay of two steps before the input affects the position of the agents. Consequently, for short prediction horizons, the cost of the velocity and input outweigh the cost decrease by moving closer to the reference or changing the formation.

For $M = \{4, 6, 8\}$, the lattice irregularities and obstacle avoidance performance are similar before they increase again for large horizons. When increasing the horizon from $M = 8$



(a) Prediction mismatch for $M = 4$.

(b) Prediction mismatch for $M = 6$.

Figure 4.9: Prediction mismatch for different methods of neighbor trajectory estimation and prediction horizons.

to $M = 10$, the average lattice irregularity and obstacle avoidance performance increases again while the required input stays the same.

Even without a more in-depth analysis of the optimal trade-off between lattice irregularity and input energy in the range from $M = 2$ to $M = 8$, this study indicates that the best performance can be achieved with a prediction horizon in the range from $M = 4$ to $M = 8$. Furthermore, considering that the NE_0 implementation does not require the communication of previous input trajectories, this algorithm seems to be the preferable approach in ADMPF algorithms.

4.4.2 SQP Iterations

In this section, the effect of the number of SQP iterations on the performance of MPF is investigated. As the resulting optimization problem is nonlinear, the number of SQP iterations at each step will affect how close the obtained solution is to a local minimum of the nonlinear problem.

Setup

Consider the same setup as described in Section 4.4.1 with $N = 20$ agents governed by the discrete-time double-integrator dynamics in (2.10), moving in a two-dimensional space $n_\mathrm{d} = 2$. For the simulation scenarios, consider again the setup in Fig. 4.3, with obstacle radii $r_o^\beta = \{6, 6.5, 7, 7.5\}$. As the previous study showed that the best performance can be achieved with neighbor trajectory estimation with zero inputs (NE_0), this algorithm will be considered in this study with a prediction horizon of $M = 5$. For the number of SQP iterations, consider $N_\mathrm{SQP} = \{1, 2, 5, 10, 50\}$.

Performance Measures

To compare the performances, consider again the integrated irregularity J_α^Σ in (4.15), the integrated obstacle avoidance performance J_β^Σ in (4.17), and the average input energy u_E in Eq. (4.18). Furthermore, the convergence times $t_{0.02}$ are compared.

Results

To be able to compare the performances, algorithms were tuned to use similar amounts of input energy (Fig. 4.10). When comparing the lattice irregularities and obstacle avoidance performances in Fig. 4.11, and convergence times in Fig. 4.12, the number of SQP iterations does not affect the performance significantly for the considered scenario with linear agent dynamics. All of the compared scenarios show very similar convergence speeds, integrated lattice irregularities, and obstacle avoidance performances. Consequently, for the presented scenario with linear agent dynamics, N_SQP can be set to one without affecting the performance.

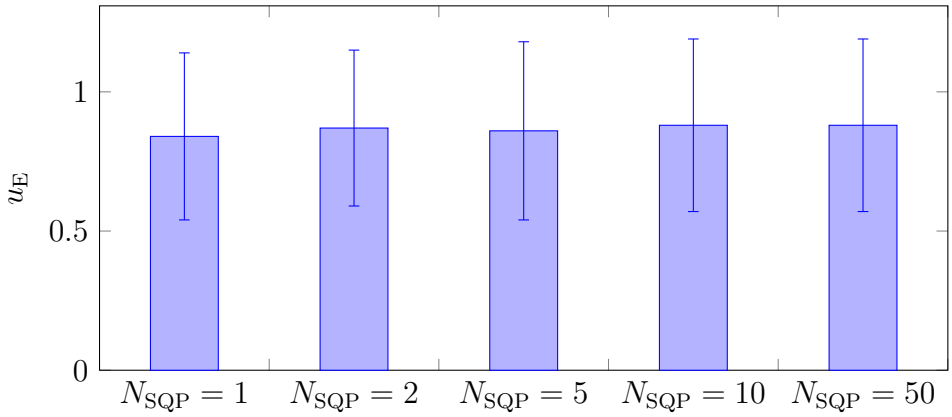


Figure 4.10: Input comparison for different numbers of SQP iterations.

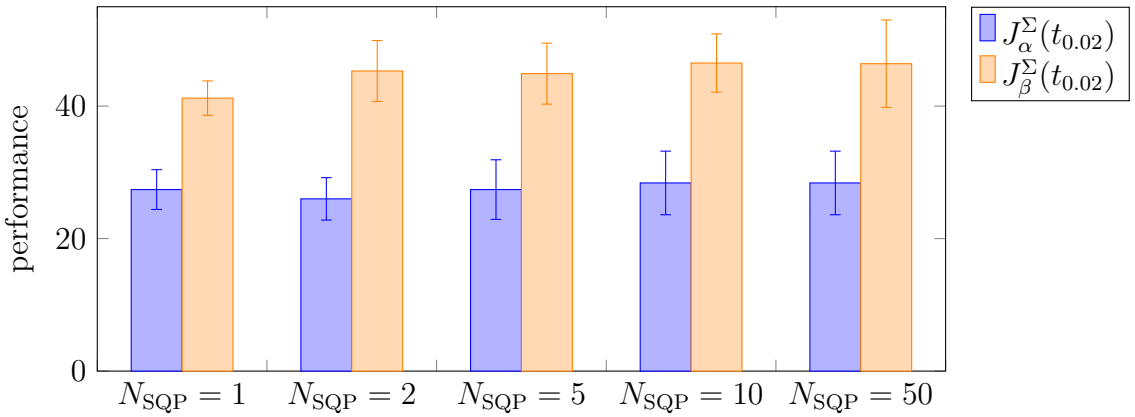


Figure 4.11: Integrated lattice irregularity and obstacle avoidance performance comparison for different numbers of SQP iterations.

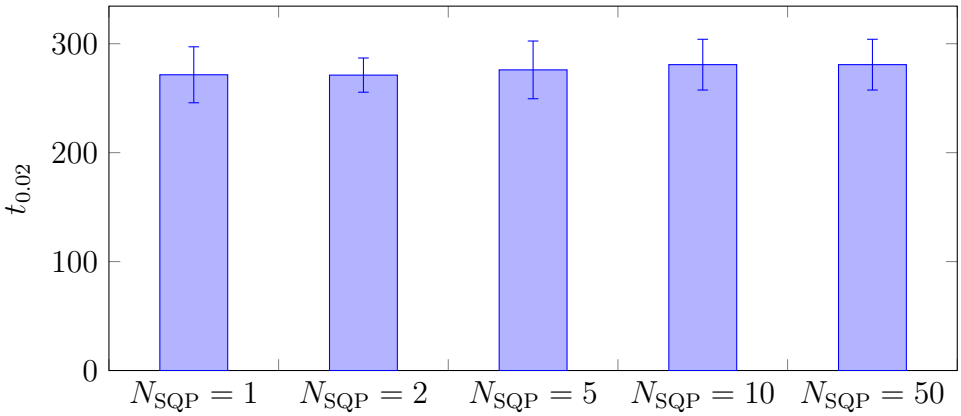


Figure 4.12: Convergence time comparison for different numbers of SQP iterations.

4.5 Performance Comparison

To compare the performance of the ADMPF algorithms presented in this section to the schemes in the related literature, the DMPF algorithm in Huang et al. (2019) and the non-predictive algorithm in Olfati-Saber (2006) are taken as reference. Two scenarios are considered: the first one considering one large obstacle and the second one a field of small obstacles.

4.5.1 Setup

For both considered scenarios, consider swarms of $N = 20$ agents moving in two-dimensional space ($n_\mathrm{d} = 2$). Agents are initialized close to an α-lattice formation around the origin and the reference is set to $p_\mathrm{r} = 0$. All simulation and tuning parameters are provided in Appendix A.1. For the γ-component in the Olfati-Saber (2006) flocking algorithm introduced in Section 2.3, the control law

$$u_i^\gamma(t) = c_1^\gamma \sigma_1(q_{\mathrm{r}_i}(t) - p_i(t)) + c_2^\gamma(p_{\mathrm{r}_i}(t) - p_i(t))$$

with tuning parameters $c_1^\gamma, c_2^\gamma > 0$ is used.

For the first scenario, consider a single obstacle at $q_1^\beta = [40\ 0]^\top$ with radius $r_1^\beta = 10$ is considered. The reference is set to $q_{\mathrm{r}_i} = [80\ 0]^\top$, $p_{\mathrm{r}_i} = 0_2$.

For the second scenario, a field of eight small obstacles with radii $r_o^\beta = 1$, $o \in \mathbb{I}_{1:8}$, and center positions

$$q_1^\beta = \begin{bmatrix} 30 \\ 0 \end{bmatrix},\ q_2^\beta = \begin{bmatrix} 30 \\ 26 \end{bmatrix},\ q_3^\beta = \begin{bmatrix} 30 \\ -26 \end{bmatrix},\ q_4^\beta = \begin{bmatrix} 45 \\ 13 \end{bmatrix},$$
$$q_5^\beta = \begin{bmatrix} 45 \\ -13 \end{bmatrix},\ q_6^\beta = \begin{bmatrix} 60 \\ 0 \end{bmatrix},\ q_7^\beta = \begin{bmatrix} 60 \\ 26 \end{bmatrix},\ q_8^\beta = \begin{bmatrix} 60 \\ -26 \end{bmatrix},$$

is considered. The reference is set to $q_{\mathrm{r}_i} = [90\ 0]^\top$, $p_{\mathrm{r}_i} = 0_2$.

4.5.2 Performance Measures

To compare the performances, consider again the integrated lattice irregularity J_α^Σ in (4.15), the integrated obstacle avoidance performance J_β^Σ in (4.17), and the convergence time $t_{0.02}$, as defined in Section 4.4.1. The final lattice irregularity is given by $J_\alpha(k_f)$. Furthermore, let t_obs denote the obstacle clearance time, i.e. the time difference between the first and last time any of the agents detects an obstacle. The algorithms were tuned to use similar amounts of input energy with respect to the average energy u_E in (4.18).

4.5.3 Results

In the following, the results for the two simulation scenarios will be presented. The performance of the different schemes and scenarios are summarized in Table 4.1.

Table 4.1: Summary of performance comparison between proposed ADMPF, predictive Huang-flocking (Huang et al. 2019), and non-predictive Olfati-Saber-flocking (Olfati-Saber 2006).

	Large Obstacle			Small Obstacles		
	ADMPF	Huang	Olfati-Saber	ADMPF	Huang	Olfati-Saber
$t_{0.02}$	109.6 s	162.8 s	196.4 s	133 s	-	273.9 s
t_{obs}	73.6 s	142 s	142.6 s	98 s	-	264.5 s
u_{E}	2.47	2.92	2.59	3.85	-	3.59
$J_\alpha(k_f)$	0.005	0.061	0.021	0.004	-	0.022
$J_\alpha^\Sigma(t_{0.02})$	11.55	15.35	14.17	22.42	-	24.19
$J_\beta^\Sigma(t_{0.02})$	13.12	38.05	67.94	21.4	-	112.68

Scenario 1 - Large Obstacle

For the three compared algorithms, the agents' trajectories are displayed in Fig. 4.13. With the ADMPF algorithm and the non-predictive Olfati-Saber flocking (Olfati-Saber 2006), the swarm splits in front of the obstacle and merges again after successfully passing it. On the other hand, in the case of the Huang MPF (Huang et al. 2019), the swarm does not split but passes the obstacle as a whole. This behavior can be explained by the large attractive forces acting between agents due to the symmetric interaction forces, preventing the swarm from splitting.

When comparing the performances summarized in Table 4.1, the proposed ADMPF algorithm results in a significantly faster convergence time and obstacle clearance time compared to the other approaches while also resulting in a lower integrated lattice irregularity and better obstacle avoidance performance. In comparison to the Olfati-Saber approach, this performance difference can be explained by the predictive nature of the ADMPF algorithm, validating the observation of an performance increase of predictive flocking compared to non-predictive flocking made in the related literature (Huang et al. 2019; Zhan and Li 2011b, 2013; Zhang et al. 2015; Zhou and Li 2017). When comparing the performance to that of the predictive approach by Huang, under the proposed algorithm the swarm clears the obstacle significantly faster and therefore also converges faster towards the reference. This indicates that a splitting of the swarm can increase the overall performance. The increased obstacle avoidance performance can also be explained by the fact that the ADMPF scheme considers a detection range larger than the desired obstacle separation ($r_o > d_o$), while, due to the symmetric interaction forces, the Huang-algorithm requires $r_o = d_o$ to avoid attraction towards the obstacle.

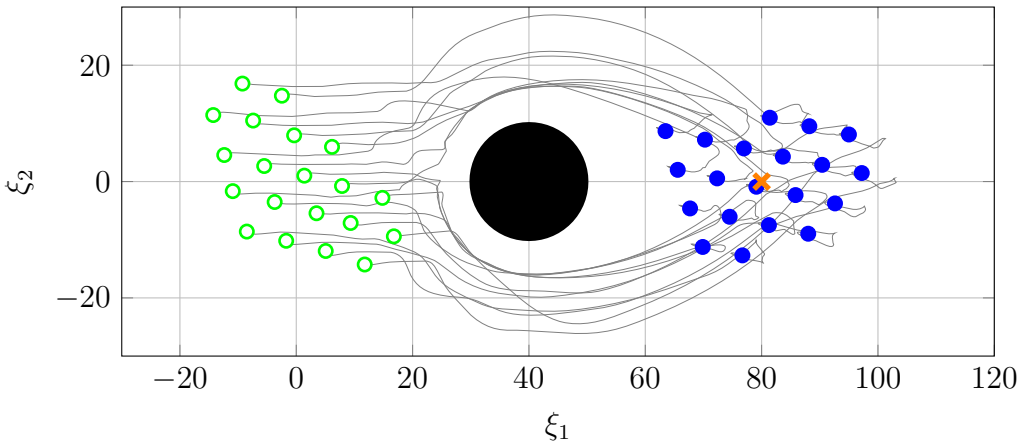
Another reason for the increased performance is the consideration of α-, β-, and γ-flocking components within the optimization problem. In the Huang-approach, only α- and β-flocking are considered part of the optimization problem and the γ-flocking component is added to the solution of the optimization problem afterwards, no longer resulting in an optimal input. Of the three compared approaches, the ADMPF algorithm also results

in the smallest final lattice irregularity. For flocking without γ-agent, this value should converge to zero, however, due to the trade-off between forming an α-lattice and converging to the reference point, it follows that $J_\alpha(k_f) > 0$ for flocking with group objectives. As the ADMPF algorithm is the only one of the three compared approaches that considers the reference tracking as part of the optimization problem, this indicates that considering this objective in the cost function reduces the final lattice irregularity.

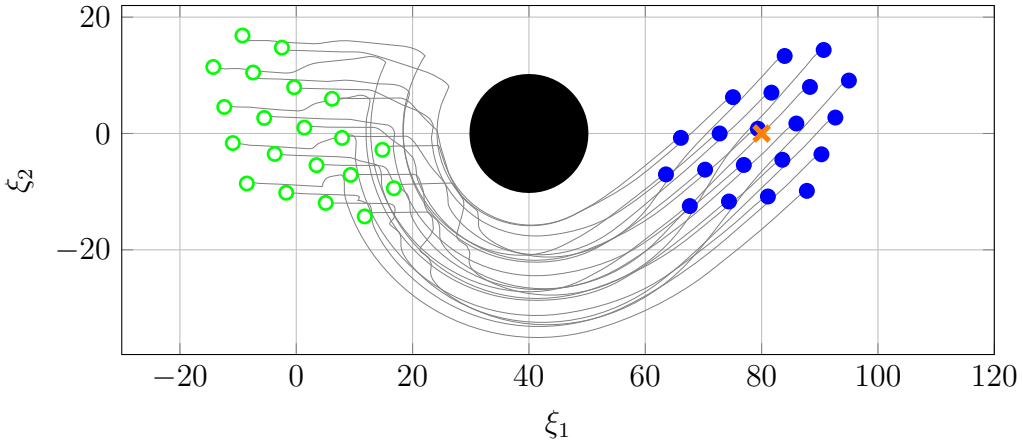
Scenario 2 - Small Obstacles

For the three compared algorithms, the agents' trajectories are displayed in Fig. 4.14. The simulation results are summarized in Table 4.1.

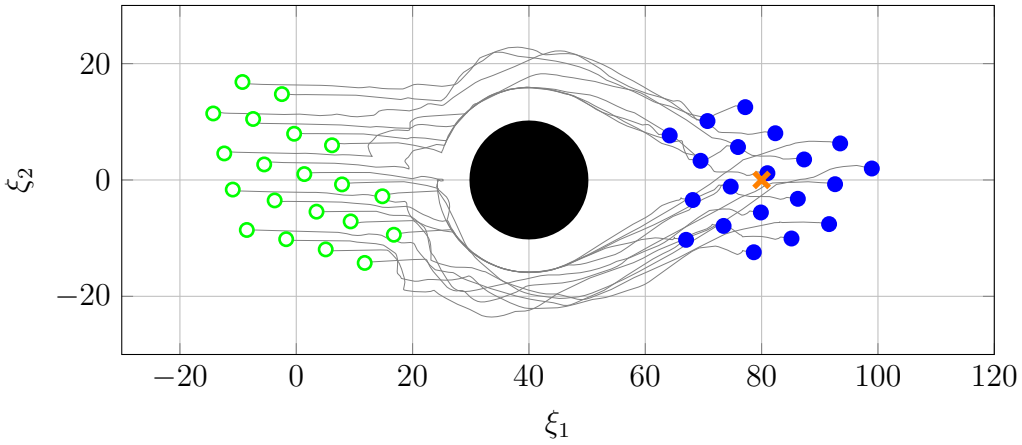
While the agents are able to complete the mission successfully under the ADMPF algorithm and the Olfati-Saber algorithm, the swarm gets stuck in the field of obstacles when applying the algorithm proposed in Huang et al. (2019). This behavior can again be explained by the symmetric design of the interaction forces. Due to the equal strength of attractive and repulsive inter-agent forces, the swarm does not split to clear the field of obstacles as the cost for doing so is too high. When comparing the performances of the ADMPF algorithm and the non-predictive Olfati-Saber flocking, the performance results are similar to the scenario with one large obstacle. The ADMPF algorithm results in a notably faster convergence, obstacle clearance time, and smaller final lattice irregularity. Wile the lattice irregularities $J_\alpha^\Sigma(t_f)$ are similar for both algorithms, the ADMPF algorithm yields a significantly better obstacle avoidance performance $J_\beta^\Sigma(t_f)$.



(a) Agent trajectories under ADMPF algorithm.

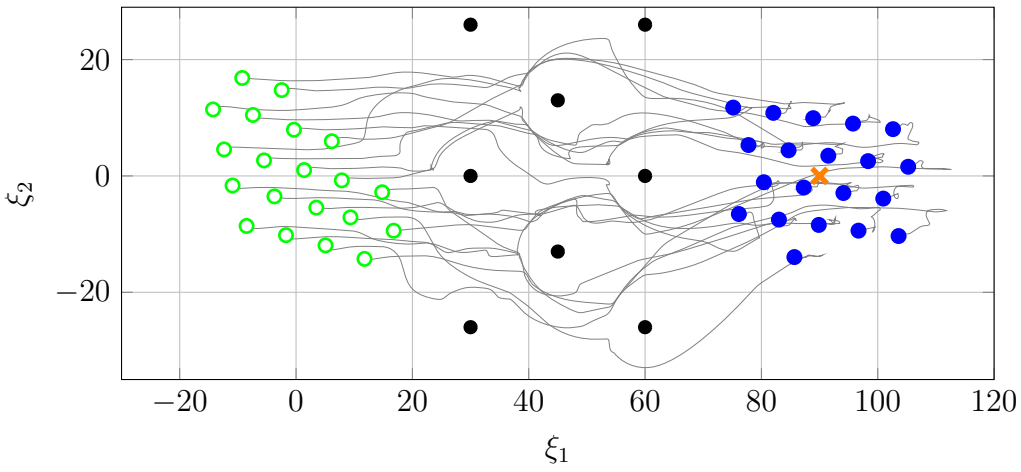


(b) Agent trajectories under Huang MPF algorithm (Huang et al. 2019).

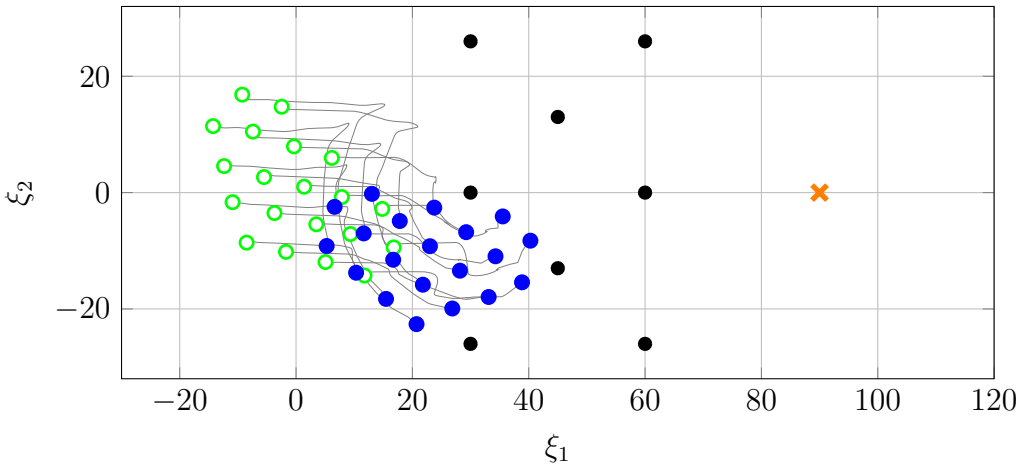


(c) Agent trajectories under Olfati-Saber flocking algorithm (Olfati-Saber 2006).

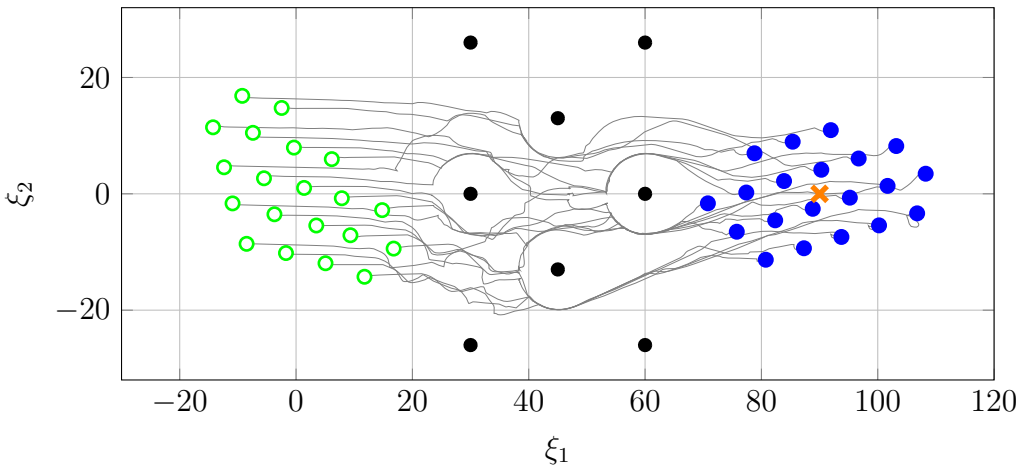
Figure 4.13: Agent trajectories for Scenario 1 under different algorithms. Initial positions are indicated in green, final positions in blue, and reference in orange.



(a) Agent trajectories under ADMPF algorithm.



(b) Agent trajectories under Huang MPF algorithm (Huang et al. 2019).



(c) Agent trajectories under Olfati-Saber flocking algorithm (Olfati-Saber 2006).

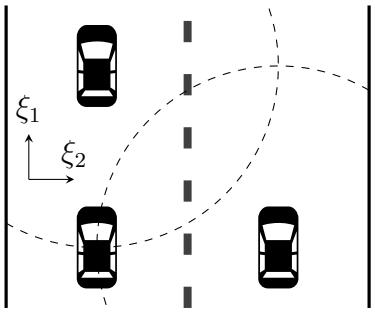
Figure 4.14: Agent trajectories for Scenario 2 under different algorithms. Initial positions are indicated in green, final positions in blue, and reference in orange.

5 Flocking with Ellipsoidal Level Sets

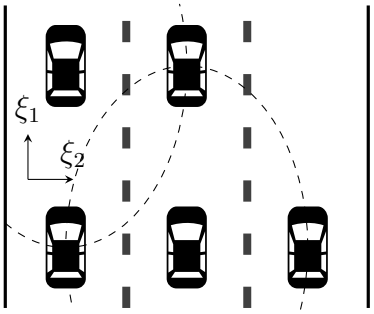
This chapter presents a framework for distributed flocking with ellipsoidal level sets, also referred to as elliptic flocking (EF). As introduced in Section 2.3, the objective of the flocking protocols discussed in the majority of the related literature is for the agents to form a desired configuration with identical relative distances between all members of the swarm, resulting in spherical desired separations (Barve and Nene 2013; Beaver and Malikopoulos 2021). Nevertheless, there are scenarios where it is advantageous to consider formations where the desired separations are not spherical, but elliptic, in order to enhance performance and optimize the use of available space.

Consider for example the group of autonomous ground vehicles driving on a highway, as depicted in Fig. 5.1. In this scenario, the agents are moving with a high velocity along the driving direction ξ_1, and the velocity in the ξ_2-direction is comparatively small. In order to address such a scenario with spherical flocking protocols, the desired distance has to be determined based on the safe separation for the high velocity ξ_1-direction. However, this also results in the same separation requirements for agents on neighboring lanes, resulting in unnecessarily large distances between lanes (Fig. 5.1a). On the other hand, when considering elliptic desired formations, as depicted in Fig. 5.1b, the available space can be utilized more efficiently by specifying the desired separations along the ξ_1- and ξ_2-directions individually. The concept of flocking protocols for elliptic formations in vehicular systems was initially introduced in Wang et al. (2022) and Wang et al. (2023).

A similar problem arises when considering swarms of multicopters in a three-dimensional space. Multicopters are in general very agile in the ξ_1-ξ_2-plane, requiring only small safety distances between them. However, due to the downwash produced by the rotors,

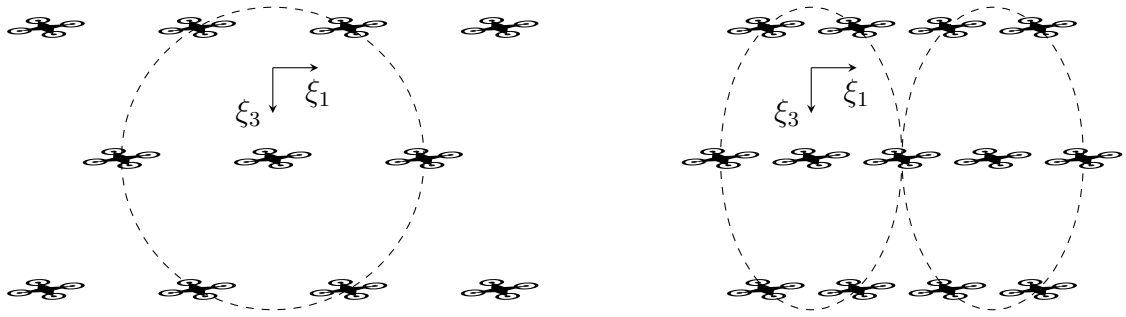


(a) Lane driving scenario with spherical desired separations.



(b) Lane driving scenario with elliptic desired separations.

Figure 5.1: Lane driving scenario indicating desired separations for selected cars.



(a) Multicopter downwash scenario with spherical desired separations.

(b) Multicopter downwash scenario with elliptic desired separations.

Figure 5.2: ξ_1-ξ_3 plane of multicopter scenario.

agents require a greater separation in the ξ_3-direction (Preiss et al. 2017). As depicted in Fig. 5.2, the available space can be used more efficiently when considering elliptic desired separations (Fig. 5.2b) instead of the spherical ones (Fig. 5.2a).

Unfortunately, the algorithms presented in Wang et al. (2022) and Wang et al. (2023) are not directly applicable to this scenario. Since their frameworks are tailored to vehicular systems, they cannot be applied to 3D flocking problems with arbitrary ellipsoidal formations. Furthermore, the approaches in Wang et al. (2022) and Wang et al. (2023) do not possess obstacle avoidance capabilities.

Following these motivating scenarios, the remainder of this chapter is dedicated to defining a general framework for elliptic flocking in n_d-dimensional spaces where n_d is typically 2 or 3. Firstly, mathematical descriptions of ellipses, ellipsoids, and elliptic lattices are provided in Section 5.1. The general framework for EF is then presented in Section 5.2. By formulating the proposed framework as a generalization of the framework in Olfati-Saber (2006) that is introduced in Section 2.3, the proposed EF framework allows for a straightforward integration of obstacle avoidance capabilities. Furthermore, Section 5.3 addresses the stability of the proposed framework and also discusses the analysis presented in Wang et al. (2022) based on the interconnection between action functions and collective flocking potentials. By formulating the EF framework as a generalization of the results in Olfati-Saber (2006), it is shown that this also allows the stability results in Olfati-Saber (2006) to be generalized to the elliptic setting addressed in this chapter. Finally, simulation examples are presented in Section 5.4, also providing a comparison between the proposed algorithms and that in Wang et al. (2022). The contents of this chapter are based on the results presented in Hastedt et al. (2024).

5.1 Mathematical Descriptions of Ellipses and Ellipsoids

Consider the two-dimensional ellipse depicted in Fig. 5.3. Let (ξ_1, ξ_2) denote the reference coordinate system, and let (ξ_1', ξ_2') be a coordinate system that is rotated by an angle θ with respect to (ξ_1, ξ_2). In Fig. 5.3, the origin of the coordinate system (ξ_1', ξ_2') coincides with the

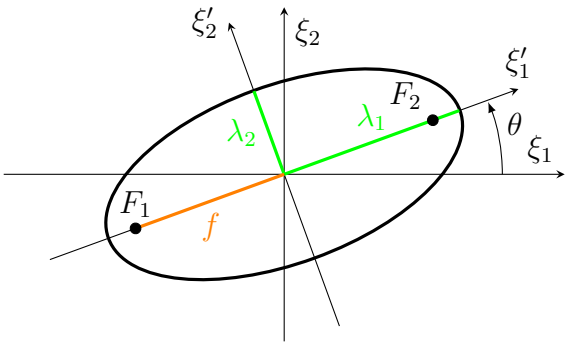


Figure 5.3: Ellipse rotated by angle θ with semi-axes λ_1 and λ_2, and focal points F_1 and F_2 with focal distance f.

center of the ellipse and its semi-axes coincide with the coordinate axes. Mathematically, there are multiple ways to describe this ellipse. In the rotated coordinate system, the ellipse E is described by the standard equation

$$E = \left\{ \begin{bmatrix} \xi_1' \\ \xi_2' \end{bmatrix} \in \mathbb{R}^2 : \frac{\xi_1'}{\lambda_1} + \frac{\xi_2'}{\lambda_2} = 1 \right\}. \tag{5.1}$$

Equivalently, the rotated ellipse can also be described as

$$E_T = \{\xi \in \mathbb{R}^2 : \|T\xi\| = 1\}, \tag{5.2}$$

where T is referred to as the elliptic transformation matrix. For arbitrary ellipses rotated by an angle θ, T can be expressed as

$$T = T(\lambda_1, \lambda_2, \theta) = \big(R(\theta)\Lambda(\lambda_1, \lambda_2)\big)^{-1},$$

where $\Lambda(\lambda_1, \lambda_2)$ is a positive definite, diagonal scaling matrix,

$$\Lambda(\lambda_1, \lambda_2) = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

and $R(\theta)$ is the 2D rotation matrix,

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

To ensure the ellipse is well-defined, the following assumption is made.

Assumption 5.1. The elliptic transformation matrix T is invertible.

Alternatively, an ellipse can also be defined via its focal points F_1 and F_2, with the geometric relationship $\lambda_2^2 = \lambda_1^2 - f^2$ for $\lambda_1 \geq \lambda_2$. Let f denote the distance of the foci from the center of the ellipse in the rotated coordinate system. With respect to the reference

system, the ellipse consists of all points with the sum of the distances to the foci being constant,

$$E_f = \left\{ \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \in \mathbb{R}^2 : \left\| \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} - R(\theta) \begin{bmatrix} f \\ 0 \end{bmatrix} \right\| + \left\| \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + R(\theta) \begin{bmatrix} f \\ 0 \end{bmatrix} \right\| = 2\lambda_1 \right\}. \tag{5.3}$$

This is the description of the ellipse used in elliptic flocking protocols in Wang et al. (2022) and Wang et al. (2023). In 2D, the parameterizations (5.3) and (5.2) are equivalent.

However, when considering three-dimensional ellipsoids with semi-axes λ_1, λ_2, and λ_3 and an orientation described by three Euler angles θ_1, θ_2, and θ_3, the definition via the focal points is no longer suitable as there exists no equivalent to the description in (5.3) for arbitrary ellipsoids. Equation (5.3) can only be extended to 3D ellipsoids with $\lambda_1 \geq \lambda_2 = \lambda_3$. On the other hand, the extension of the definition in (5.2) to three-dimensional ellipsoids is straightforward by defining a 3D transformation matrix

$$T = T(\lambda_1, \lambda_2, \lambda_3, \theta_1, \theta_2, \theta_3) = \big(R(\theta_1, \theta_2, \theta_3)\Lambda(\lambda_1, \lambda_2, \lambda_3)\big)^{-1},$$

with positive definite, diagonal scaling matrix

$$\Lambda(\lambda_1, \lambda_2, \lambda_3) = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

and $R(\theta_1, \theta_2, \theta_3)$ denoting a 3D rotation matrix. The 3D ellipsoid can be described by

$$E_{3\mathrm{D}} = \{\xi \in \mathbb{R}^3 : \|T\xi\| = 1\}.$$

Similar and Confocal Ellipses

In the context of flocking, one important measure is the deviation from the desired formation, which, in the context of the elliptic flocking protocols introduced in the next section, corresponds to the distance of a point from an ellipse. For the following discussion, without loss of generality, consider a 2D ellipse with $\theta = 0$. Then, for the definitions in (5.2) and (5.3), the level sets

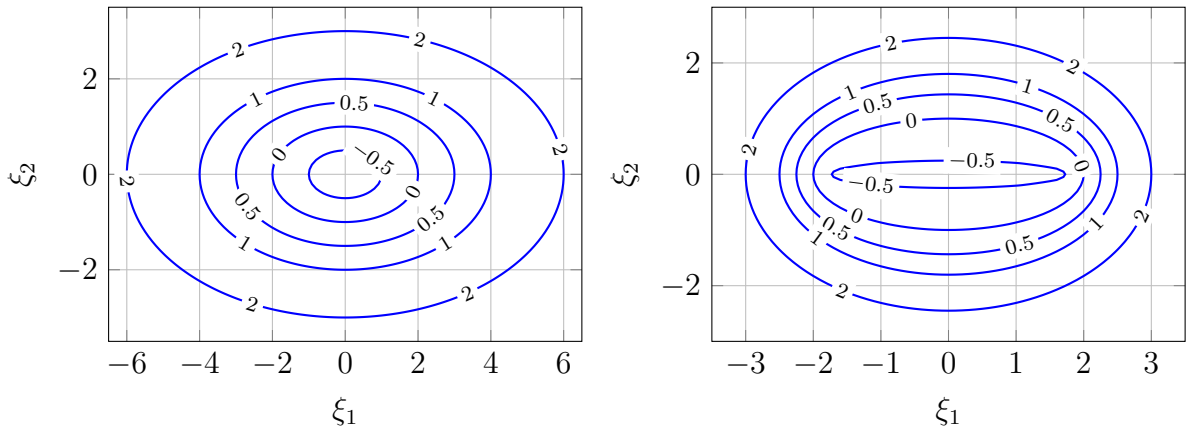
$$\ell_{\epsilon,T}(\xi) = \|T\xi\| - 1 = c \tag{5.4}$$

and

$$\ell_{\epsilon,f}(\xi) = \|\xi - f\| + \|\xi + f\| - 2\lambda_1 = c \tag{5.5}$$

can be utilized to visualize all points that are at distance c from the desired ellipse. In Fig. 5.4, the level curves for $\ell_{\epsilon,T} = c$ and $\ell_{\epsilon,f} = c$ are displayed for an ellipse with $\lambda_1 = 2$, $\lambda_2 = 1$, and $c = \{-0.5, 0, 0.5, 1, 2\}$. For $\ell_{\epsilon,T}(\xi) = c$ in Fig. 5.4a, the level sets form similar ellipses with semi-axes $\lambda_{1,T}^c = \sqrt{1+c}\lambda_1$ and $\lambda_{2,T}^c = \sqrt{1+c}\lambda_2$. As the ellipses are similar, the ratio between the semi-axes is constant, i.e.

$$\frac{\lambda_1}{\lambda_2} = \frac{\lambda_{1,T}^c}{\lambda_{2,T}^c}.$$



(a) Level sets for $\ell_{\epsilon,T}(\xi) = c$.

(b) Level sets for $\ell_{\epsilon,f}(\xi) = c$.

Figure 5.4: Level set comparison of $\ell_{\epsilon,T}(\xi) = c$ and $\ell_{\epsilon,f}(\xi) = c$ for $c = \{-0.5, 0, 0.5, 1, 2\}$ for an ellipse with $\lambda_1 = 2$, $\lambda_2 = 1$, and θ=0.

On the other hand, for $\ell_{\epsilon,f}(\xi) = c$ in Fig. 5.4b, the level sets form confocal ellipses with $\lambda^c_{1,f} = \sqrt{\lambda_1^2 + \tilde{c}}$ and $\lambda^c_{2,f} = \sqrt{\lambda_2^2 + \tilde{c}}$, with $\tilde{c} = c\lambda_1 + \frac{c^2}{4}$. Hence, using the focal point parameterization, the semi-axes ratio depends on the distance from the ellipse, i.e. for $c \neq 1$,

$$\frac{\lambda_1}{\lambda_2} \neq \frac{\lambda^c_{1,f}}{\lambda^c_{2,f}}.$$

The above discussion shows that, even though the ellipse definitions in (5.2) and (5.3) are equivalent, the level sets of the deviations from this ellipse in (5.4) and (5.5) are not.

5.2 Elliptic Flocking Protocols

As in Section 2.3, consider a homogeneous MAS, where the agents are governed by the double-integrator dynamics in (2.8), yielding the collective dynamics in (2.22). Before defining the protocols for free elliptic flocking (α-EF), elliptic flocking with group objectives (α-γ-EF), and elliptic flocking with obstacle avoidance (α-β-γ-EF), the desired configurations are defined analogously to the α-lattice and quasi α-lattice in Definitions 2.7 and 2.8. This also requires the definition of an elliptic communication topology. Due to the extensibility to 3D scenarios, the following deviations are based on the elliptic transformation matrix in (5.2).

5.2.1 Elliptic Proximity Graphs and ϵ-Lattices

To ensure that agents are only interacting with their direct neighbors, let $\mathcal{G}^\epsilon(q) = (\mathcal{V}, \mathcal{E}^\epsilon(q))$ define the elliptic proximity graph with the elliptic edge set

$$\mathcal{E}^\epsilon(q) = \{(i,j) \in \mathcal{V} : \|Tq_{ij}\| < r,\ j \neq i\}.$$

In the context of EF, the superscript ϵ is used to indicate quantities that are related to the elliptic flocking formulation rather than the closely related spherical framework in Olfati-Saber (2006). Similar to (2.27), the elements of the elliptic adjacency matrix are then defined as

$$a_{ij}^{\epsilon}\big(\sigma_s(Tq_{ij}(t))\big) = \rho_h\left(\frac{\sigma_s(Tq_{ij}(t))}{r_\alpha}\right), \tag{5.6}$$

and the corresponding graph Laplacian is denoted as $L^{\epsilon}(q)$ with $L^{\epsilon}_{(n_\mathrm{d})}(q) = L^{\epsilon}(q) \otimes I_{n_\mathrm{d}}$. Analogous to (2.15), the elliptic neighborhood of an agent i is defined as

$$\mathcal{N}_i^{\epsilon}(q) = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}^{\epsilon}\}.$$

Using these definitions, an elliptic lattice, also referred to as ϵ-lattice, can be defined analogously to Definition 2.7.

Definition 5.1 (ϵ-lattice)**.** Given an elliptic proximity graph $\mathcal{G}^{\epsilon}(q) = (\mathcal{V}, \mathcal{E}^{\epsilon}(q))$ with $r > d > 0$ and T satisfying Assumption 5.1, an ϵ-lattice is a configuration q satisfying the constraint $\|Tq_{ij}\| = d$, $\forall (i,j) \in \mathcal{E}^{\epsilon}(q)$.

A quasi ϵ-lattice is then defined similarly to Definition 2.8.

Definition 5.2 (Quasi ϵ-lattice)**.** Given an elliptic proximity graph $\mathcal{G}^{\epsilon}(q) = (\mathcal{V}, \mathcal{E}^{\epsilon}(q))$ with $r > d > 0$, T satisfying Assumption 5.1, and a small perturbation $0 < \eta \ll d$, a quasi ϵ-lattice is a configuration q satisfying $-\eta < \|Tq_{ij}\| - d < \eta$, $\forall (i,j) \in \mathcal{E}^{\epsilon}(q)$.

Note that the parameterization of the ϵ-lattice is not unique. For an ϵ-lattice parameterized by T and d, it also holds that, for $\kappa > 0$,

$$\kappa(\|Tq_{ij}\| - d) = 0,$$

i.e. (r, d, T) and $(r', d', T') = (\kappa r, \kappa d, \kappa T)$ define the same ϵ-lattice. One way of characterizing the different parameterizations is via the determinant of the transformation matrix T. To obtain a unique parameterization (r, d, T), this determinant can be normalized to one. The lattices with the normalized parameters $(\bar{r}, \bar{d}, \bar{T})$,

$$\bar{r} = \frac{r}{\sqrt[n_\mathrm{d}]{\det(T)}}, \quad \bar{d} = \frac{d}{\sqrt[n_\mathrm{d}]{\det(T)}}, \quad \bar{T} = \frac{T}{\sqrt[n_\mathrm{d}]{\det(T)}}, \tag{5.7}$$

are then referred to as the normalized ϵ-lattice and normalized quasi ϵ-lattice, respectively.

5.2.2 α-Elliptic Flocking

Following the derivations of Olfati-Saber (2006) as described in Section 2.3, the collective potential $V_\alpha^{\epsilon} : \mathbb{R}^{n_\mathrm{d}N} \to \mathbb{R}_{\geq 0}$ can be defined analogously to (2.30). Consider again the pairwise attractive/repulsive potential ψ_α and the action function ϕ_α in (2.29) and (2.28),

$$\psi_\alpha(z) = \int_{d_\alpha}^{z} \phi_\alpha(y)\, \mathrm{d}y, \quad \phi_\alpha(z) = \rho_h\left(\frac{z}{r_\alpha}\right)\frac{z - d_\alpha}{\sqrt{1 + \|z - d_\alpha\|^2}}.$$

The collective elliptic potential can then be defined as

$$V_\alpha^\epsilon(q(t)) = \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V},j\neq i}\psi_\alpha\big(\sigma_s(Tq_{ij}(t))\big). \tag{5.8}$$

The agent-wise control law for free elliptic flocking is then given analogously to (2.31),

$$u_{\epsilon,i}^\alpha(t) = -c_1^\alpha\sum_{j\in\mathcal{N}_i^\epsilon(q(t))}\nabla\psi_\alpha\big(\sigma_s(Tq_{ij}(t))\big) + c_2^\alpha\sum_{j\in\mathcal{N}_i^\epsilon(q(t))}a_{ij}^\epsilon\big(\sigma_s(Tq_{ij}(t))\big)p_{ij}(t), \tag{5.9}$$

with tuning parameters $c_1^\alpha, c_2^\alpha > 0$. Due to the transformation matrix, the derivative of the pairwise potential with respect to the position becomes

$$\nabla\psi_\alpha\big(\sigma_s(Tq_{ij}(t))\big) = -\phi_\alpha\big(\sigma_s(Tq_{ij}(t))\big)\frac{T^\top Tq_{ij}}{\sqrt{1+s\|Tq_{ij}\|^2}}. \tag{5.10}$$

The $T^\top T$-term in the numerator of (5.10) arises from applying the chain rule when taking the derivative with respect to q. Analogous to (2.33), the collective control law can be stated as

$$u_\epsilon^\alpha(t) = -c_1^\alpha\nabla V_\alpha^\epsilon(q(t)) - c_2^\alpha L_{(n_d)}^\epsilon\big(q(t)\big)p(t). \tag{5.11}$$

Remark. For $T = I$, it follows that $a_{ij}^\epsilon\big(\sigma_s(Tq_{ij})\big) = a_{ij}\big(\sigma_s(q_{ij})\big)$, $L_{(n_d)}^\epsilon(q) = L_{(n_d)}(q)$, $V_\alpha^\epsilon(q) = V_\alpha(q)$, and $\mathcal{G}^\epsilon(q) = \mathcal{G}(q)$. Consequently, the control laws in (5.9) and (5.11) are equal to the flocking laws in (2.31) and (2.33). Hence, by defining EF using the elliptic transformation matrix T, the presented framework can be seen as a generalization of the work in Olfati-Saber (2006).

On Direction-Dependency of Elliptic Flocking Control Laws

In the case of spherical flocking, the resulting flocking control law only depends on the relative distances between agents. For $T = I$, the control law (5.9) is therefore constant along a level set $\ell_\epsilon(q_{ij}) = \|Tq_{ij}\| - d = c$. On the other hand, in elliptic flocking ($T \neq I$), the control law also depends on the direction of the vector q_{ij} relative to a fixed coordinate frame. This direction-dependency is caused by direction-dependent scaling of the transformation matrix to achieve the desired elliptic formation. Since the parameters (r, d, T) do not uniquely determine the desired ϵ-lattice, the choice of the ϵ-lattice-parameterization (r, d, T) also affects the magnitude of the control input.

To illustrate these effects, consider two agents with $n_\mathrm{d} = 2$, $p_{ij} = 0$, and the flocking protocol (5.11) with $s = 0.1$, $h = 0.2$, and $c_1^\alpha = 8.385$. Let

$$\omega_{T,d}^c(\varphi) = (d+c)T^{-1}\begin{bmatrix}\cos(\varphi)\\ \sin(\varphi)\end{bmatrix}, \quad \varphi\in[0,2\pi),$$

define a parameterization of the level set $\Omega^c = \{q_{ij} \in \mathbb{R}^2 : \|Tq_{ij}\| - d = c\}$. Consider furthermore the following four sets of parameters (r_i, d_i, T_i)

$$(r_0, d_0, T_0) = \left(12, 10, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad (r_1, d_1, T_1) = \left(12, 10, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\right),$$
$$(r_2, d_2, T_2) = \left(6, 5, \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad (r_3, d_3, T_3) = \left(\frac{12}{\sqrt{2}}, \frac{10}{\sqrt{2}}, \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\right) = (\bar{r}, \bar{d}, \bar{T}).$$

Here, the first set of parameters, (r_0, d_0, T_0), corresponds to spherical flocking, while the other three all parameterize the same ϵ-lattice with different determinants of the transformation matrix T. The last set of parameters, $(i = 3)$, corresponds to the normalized parameterization $(\det(T) = 1)$.

The effects of the different parameterizations can be visualized when plotting the magnitudes of the control inputs along a level set parameterized by φ, $\|u_\epsilon^\alpha(\omega^c(\varphi))\|$, over $\varphi \in [0, 2\pi)$. For the four sets of parameters presented above, the corresponding normed control inputs are depicted in Fig. 5.5. As discussed above, for the spherical protocol (r_0, d_0, T_0), the control law is independent of φ. For the three elliptical protocols with parameterizations (r_i, d_i, T_i), $i \in \mathbb{I}_{1:3}$, the relative control input along the direction of the shorter semi-axis $\varphi_{\lambda_2} = \{\frac{\pi}{2}, \frac{3\pi}{2}\}$ is approximately two times larger compared to the longer semi-axis $\varphi_{\lambda_1} = \{0, 2\pi\}$, which corresponds to the ratio of the semi-axis lengths. When comparing the absolute values for the three elliptic parameterizations, this example furthermore shows that, for a given φ, the magnitude of the control input is approximately proportional to the determinant of the transformation matrix.

The above example shows that the parameterization of the desired configuration must be chosen carefully when implementing the elliptic flocking protocol as the magnitude of the control law is not only affected by the tuning parameter c_1^α, but also by the choice of r, d, and T, motivating the use of the normalized parameterization for practical applications.

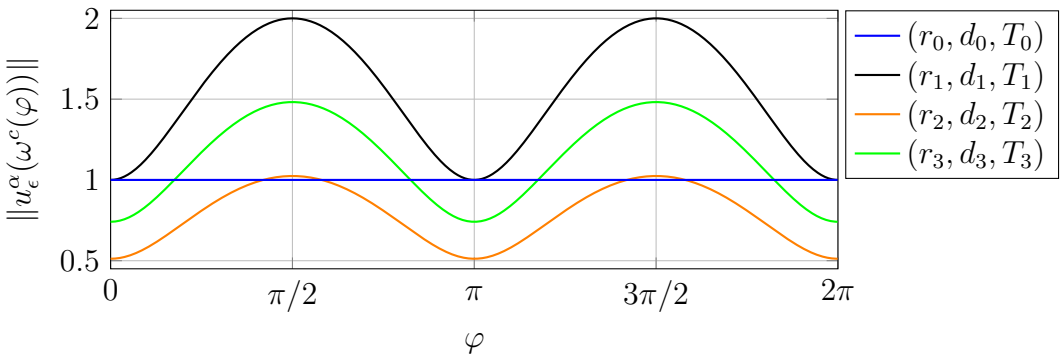


Figure 5.5: $\|u_\epsilon^\alpha(\omega_{T,d}^c(\varphi))\|$ for four parameter sets (r_i, d_i, T_i) for the level set $c = 0.1d_i$.

5.2.3 α-γ-Elliptic Flocking

As described in Section 2.3, group objectives can be added to the flocking protocols by considering virtual γ-agents with reference position $q_{i,\mathrm{r}} \in \mathbb{R}^{n_\mathrm{d}}$ and reference velocity $p_{\mathrm{r}_i} \in \mathbb{R}^{n_\mathrm{d}}$. The γ-flocking component can then be modeled as a function of the agent's state and γ-agent, $u_i^\gamma = f_i^\gamma(q_i, p_i, q_{\mathrm{r}_i}, p_{\mathrm{r}_i})$ and $u^\gamma = \mathrm{col}(u_1^\gamma, \dots, u_N^\gamma)$. The corresponding agent-wise and collective control laws in the elliptic framework are then given by

$$u_{\epsilon,i}(t) = u_{\epsilon,i}^\alpha(t) + u_i^\gamma(t) \tag{5.12}$$

and

$$u_\epsilon(t) = u_\epsilon^\alpha(t) + u^\gamma(t). \tag{5.13}$$

5.2.4 α-β-γ-Elliptic Flocking

Consider the definition of the β-agents as described in Section 2.3. Adding obstacle avoidance to the elliptic flocking protocol follows the same procedure as in the spherical case by adding an obstacle avoidance term to the agent-wise and collective control laws in (5.12) and (5.13),

$$u_{\epsilon,i}(t) = u_{\epsilon,i}^\alpha(t) + u_{\epsilon,i}^\beta(t) + u_i^\gamma(t), \tag{5.14}$$

and

$$u_\epsilon(t) = u_\epsilon^\alpha(t) + u_\epsilon^\beta(t) + u^\gamma(t), \tag{5.15}$$

where $u_{\epsilon,i}^\beta(t)$ and $u_\epsilon^\beta(t)$ are the corresponding agent-wise and collective obstacle avoidance components, respectively. The desired elliptical obstacle interactions can then be characterized by a transformation matrix T_β, following the same derivations as in Section 5.1. With a desired obstacle detection range $r_o > 0$, the set of elliptical β-neighbors is given by

$$\mathcal{N}_i^{\epsilon,\beta}(q) = \{o \in \mathcal{V}^\beta : \|T_\beta q_{io}\| < r_o\}. \tag{5.16}$$

With ψ_β and ϕ_β as defined in (2.39) and (2.38),

$$\psi_\beta(z) = \int_{d_\beta}^{z} \phi_\beta(y)\,\mathrm{d}y, \quad \phi_\beta(z) = \rho_h\left(\frac{z}{d_\beta}\right)\left(\nabla\sigma_1(z - d_\beta) - 1\right),$$

the elliptic obstacle avoidance potential is defined analogously to (2.40) as

$$V_\beta^\epsilon(q(t)) = \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{o\in\mathcal{V}^\beta} \psi_\beta\left(\sigma_s(T_\beta q_{io}(t))\right). \tag{5.17}$$

The agent-wise and collective obstacle avoidance inputs are then given by

$$u_i^\beta(t) = -c_1^\beta \sum_{o\in\mathcal{N}_i^{\epsilon,\beta}(t)} \nabla\psi_\beta\big(\sigma_s(T_\beta q_{io}(t))\big) + c_2^\beta \sum_{o\in\mathcal{N}_i^{\epsilon,\beta}(t)} \rho_h\left(\frac{\sigma_s(T_\beta q_{io}(t))}{d_\beta}\right) p_{io}(t), \tag{5.18}$$

and

$$u^\beta(t) = -c_1^\beta \nabla V_\beta^\epsilon(q(t)) - c_2^\beta \sum_{i\in\mathcal{V}}\sum_{o\in\mathcal{V}^\beta} \rho_h\left(\frac{\sigma_s(T_\beta q_{io}(t))}{d_\beta}\right) p_{io}(t). \tag{5.19}$$

Analogous to the free flocking case, the choice of (r_o, d_o, T_β) does not uniquely characterize the desired obstacle avoidance behavior, however, a unique parameterization can be obtained by defining the normalized parameters $(\bar{r}_o, \bar{d}_o, \bar{T}_\beta)$. As in the free flocking case, setting $T_\beta = I$ recovers the spherical obstacle avoidance described in Section 2.3.

5.3 Stability Analysis of Elliptic Flocking

As discussed in Section 5.2, the EF framework presented can be seen as a generalization of the spherical framework presented in Section 2.3. This section is dedicated to showing that the line of reasoning of the stability analysis for the spherical framework presented in Olfati-Saber (2006) can be extended to the general elliptic framework. The following subsections address the extension of the stability results in Olfati-Saber (2006) to free elliptic flocking (Section 5.3.1), elliptic flocking with group objectives (Section 5.3.2), and elliptic flocking with obstacle avoidance (Section 5.3.3).

5.3.1 Analysis of Free Elliptic Flocking

As demonstrated in Olfati-Saber (2006), it is more convenient to perform the analysis in a moving frame. Hence, to analyze the stability of the protocol in (5.9), the flocking dynamics are expressed in a moving coordinate system fixed to the center of the swarm. Let

$$q_{\text{ave}}(t) = \frac{1}{N}\sum_{i=1}^{N} q_i(t), \quad p_{\text{ave}}(t) = \frac{1}{N}\sum_{i=1}^{N} p_i(t),$$

denote the average (center) position and velocity of the swarm, respectively. The relative positions and velocities of each agent with respect to the center of the swarm are then given by

$$\begin{aligned} q_i^{\text{c}}(t) &= q_i(t) - q_{\text{ave}}(t), \\ p_i^{\text{c}}(t) &= p_i(t) - p_{\text{ave}}(t). \end{aligned}$$

Furthermore, let $q^{\text{c}} = \text{col}(q_i^{\text{c}}, \dots q_N^{\text{c}})$ and $p^{\text{c}} = \text{col}(p_i^{\text{c}}, \dots p_N^{\text{c}})$ denote the collective relative positions and velocities. From these definitions, it follows directly that $q_i^{\text{c}} - q_j^{\text{c}} = q_i - q_j$ and $p_i^{\text{c}} - p_j^{\text{c}} = p_i - p_j$, and consequently $V_\alpha^\epsilon(q^{\text{c}}) = V_\alpha^\epsilon(q)$ and $\nabla V_\alpha^\epsilon(q^{\text{c}}) = \nabla V_\alpha^\epsilon(q)$. For the free flocking protocol in (5.11), the collective dynamics in the moving frame are then given by

$$\begin{aligned} \dot{q}^{\text{c}}(t) &= \dot{p}^{\text{c}}(t), \\ \dot{p}^{\text{c}}(t) &= -c_1^\alpha \nabla V_\alpha^\epsilon(q^{\text{c}}(t)) - c_2^\alpha L_{(n_d)}^\epsilon\big(q^{\text{c}}(t)\big)p^{\text{c}}(t). \end{aligned} \tag{5.20}$$

To analyze the behavior of this system, consider the total energy of the system expressed via the Hamiltonian

$$H_\alpha(q^{\text{c}}(t), p^{\text{c}}(t)) = c_1^\alpha V_\alpha^\epsilon(q^{\text{c}}(t)) + K(p^{\text{c}}(t)), \tag{5.21}$$

with $K(p^c(t))$ denoting the potential energy of the system,

$$K(p^c(t)) = \frac{1}{2}\|p^c(t)\|^2. \tag{5.22}$$

Before stating the theorem for stability, the relation between the minima of the collective potential and all possible ϵ-lattice configurations is first established analogously to Olfati-Saber (2006, Lemma 3).

Lemma 5.1. *Given an ϵ-lattice with T satisfying Assumption 5.1. Then, every local minimum of $V_\alpha^\epsilon(q)$ is an ϵ-lattice and vice versa.*

Proof. Let q^* be an ϵ-lattice configuration. Using the facts that $V_\alpha^\epsilon(q^*) = 0$ and $\sigma_s(T(q_j^* - q_i^*)) = d_\alpha \implies \|T(q_j^* - q_i^*)\| = d$, the proof follows directly from the proof of Olfati-Saber (2006, Lemma 3). ■

With this lemma, the stability result for free flocking can be stated analogously to Olfati-Saber (2006, Theorem 1).

Theorem 5.2. *Consider a group of agents governed by the dynamics in* (5.20) *with T satisfying Assumption 5.1. Let $\Omega_{H_\alpha}^c = \{(q^c, p^c) : H_\alpha(q^c, p^c) < c\}$ denote a level set of the Hamiltonian such that for any solution starting in $\Omega_{H_\alpha}^c$, the agents form a cohesive flock $\forall t \geq 0$. Then, the following statements hold.*

1. *Almost every solution of* (5.20) *converges to an equilibrium $(q^{c*}, 0)$, where q^{c*} is an ϵ-lattice.*
2. *All agents asymptotically move with the same velocity, $\lim_{t\to\infty} p_{ij}(t) = 0$, $\forall i, j \in \mathcal{V}$.*
3. *Given $c < c' = \psi_\alpha(0)$, no inter-agent collisions occur, $q_{ij}(t) > 0$, $\forall i, j \in \mathcal{V}$, $\forall t \geq 0$.*

Proof. The proof follows directly from the proof of Olfati-Saber (2006, Theorem 1) by showing that the derivative of the Hamiltonian is monotonically decreasing, i.e.

$$\begin{aligned}
\dot{H}_\alpha(q^c(t), p^c(t)) &= c_1^\alpha(\nabla V_\alpha^\epsilon(q^c(t)))^\top p^c(t) + (p^c(t))^\top u_\epsilon^\alpha(t) \\
&= -c_2^\alpha(p^c(t))^\top L_{(n_d)}^\epsilon\big(q^c(t)\big)p^c(t) \\
&= -c_2^\alpha\frac{1}{2}\sum_{(i,j)\in\mathcal{E}^\epsilon(q^c(t))} a_{ij}^\epsilon(q^c(t))\|p_j^c(t) - p_i^c(t)\|^2 \\
&\leq 0.
\end{aligned}$$

Using LaSalle's invariance principle and Lemma 5.1, one can argue that the swarm converges towards an equilibrium $(q^{c*}, 0)$, where q^{c*} is an ϵ-lattice. With $p^c = 0$ corresponding to all agents moving with the same velocity, this completes the proof of statements 1 and 2. The proof of statement 3 is identical to the proof of Theorem 1 in Olfati-Saber (2006). ■

5.3.2 Analysis of Elliptic Flocking with Group Objectives

To analyze stability for the flocking protocol with group objectives in (5.13), consider a linear γ-flocking component

$$f_i^\gamma(q_i, p_i, q_{\mathrm{r}_i}, p_{\mathrm{r}_i}) = -c_1^\gamma(q_i - q_{\mathrm{r}_i}) - c_2^\gamma(p_i - p_{\mathrm{r}_i}), \tag{5.23}$$

with tuning parameters $c_1^\gamma, c_2^\gamma > 0$. Then, according to Olfati-Saber (2006, Lemma 2), the collective γ-component $f^\gamma(q, p, q_{\mathrm{r}_i}, p_{\mathrm{r}_i})$ can be decomposed into

$$f^\gamma(q, p, q_{\mathrm{r}_i}, p_{\mathrm{r}_i}) = g^\gamma(q^{\mathrm{c}}, p^{\mathrm{c}}) + 1_N \otimes h^\gamma(q_{\mathrm{ave}}, p_{\mathrm{ave}}, q_{\mathrm{r}_i}, p_{\mathrm{r}_i}) \tag{5.24}$$

with

$$g^\gamma(q^{\mathrm{c}}, p^{\mathrm{c}}) = -c_1^\gamma q^{\mathrm{c}} - c_2^\gamma p^{\mathrm{c}}, \tag{5.25}$$

$$h^\gamma(q_{\mathrm{ave}}, p_{\mathrm{ave}}, q_{\mathrm{r}_i}, p_{\mathrm{r}_i}) = -c_1^\gamma(q_{\mathrm{ave}} - q_{\mathrm{r}_i}) - c_2^\gamma(p_{\mathrm{ave}} - p_{\mathrm{r}_i}). \tag{5.26}$$

The collective dynamics in the moving frame are then given by

$$\begin{aligned} \dot{q}^{\mathrm{c}}(t) &= \dot{p}^{\mathrm{c}}(t), \\ \dot{p}^{\mathrm{c}}(t) &= -c_1^\alpha \nabla V_\alpha^\epsilon(q^{\mathrm{c}}(t)) - c_2^\alpha L_{(n_d)}^\epsilon\big(q^{\mathrm{c}}(t)\big)p^{\mathrm{c}}(t) + g^\gamma(q^{\mathrm{c}}(t), p^{\mathrm{c}}(t)), \end{aligned} \tag{5.27}$$

and the dynamics of the center of the swarm are

$$\begin{aligned} \dot{q}_{\mathrm{ave}}(t) &= p_{\mathrm{ave}}(t), \\ \dot{p}_{\mathrm{ave}}(t) &= h^\gamma(q_{\mathrm{ave}}(t), p_{\mathrm{ave}}(t), q_{\mathrm{r}_i}(t), p_{\mathrm{r}_i}(t)). \end{aligned} \tag{5.28}$$

Similar to (5.21), the Hamiltonian for flocking with the linear γ-agent in (5.23) can be stated as

$$H_\gamma(q^{\mathrm{c}}(t), p^{\mathrm{c}}(t)) = c_1^\alpha V_\alpha^\epsilon(q^{\mathrm{c}}(t)) + c_1^\gamma V_\gamma(q^{\mathrm{c}}(t)) + K(p^{\mathrm{c}}(t)), \tag{5.29}$$

with

$$V_\gamma(q^{\mathrm{c}}(t)) = \frac{1}{2}(q^{\mathrm{c}}(t))^\top q^{\mathrm{c}}(t). \tag{5.30}$$

The stability result for elliptic flocking with group objectives can then be stated analogously to Theorem 2 in Olfati-Saber (2006) as follows:

Theorem 5.3. *Consider a group of agents governed by the dynamics in* (5.27) *with T satisfying Assumption 5.1. Let $K(p^c(0))$ and $V_\gamma(q^c(0))$ be finite. Then, the following statements hold.*

1. *The group of agents is cohesive $\forall t \geq 0$.*
2. *Almost every solution of* (5.27) *asymptotically converges to an equilibrium $(q^{c*}, 0)$, where q^{c*} is a local minimum of $c_1^\alpha V_\alpha^\epsilon(q^c) + c_1^\gamma V_\gamma(q^c)$.*
3. *All agents asymptotically move with the same velocity, $\lim_{t\to\infty} p_{ij}(t) = 0$, $\forall i, j \in \mathcal{V}$.*
4. *If $H_\gamma(q^c(0), p^c(0)) < (k+1)c'$ with $c' = \psi_\alpha(0)$ and $k \in \mathcal{N}$, then at most k distinct pairs of agents can collide.*

Proof. The proof follows directly from the proof of Olfati-Saber (2006, Theorem 1) and is similar to the proof of Theorem 5.2. Statement 1 is based on $K(p^{\mathrm{c}}(0))$ and $V_\gamma(q^{\mathrm{c}}(0))$ being finite and the derivative of the Hamiltonian in (5.29) being strictly decreasing $\forall p^{\mathrm{c}}(t) \neq 0$,

$$\begin{aligned}\dot{H}_\gamma(q^{\mathrm{c}}(t), p^{\mathrm{c}}(t)) &= c_1^\alpha \big(\nabla V_\alpha^\epsilon(q^{\mathrm{c}}(t))\big)^\top p^{\mathrm{c}}(t) + c_1^\gamma \big(\nabla V_\gamma(q^{\mathrm{c}}(t))\big)^\top p^{\mathrm{c}}(t) + (p^{\mathrm{c}}(t))^\top \dot{p}^{\mathrm{c}}(t) \\ &= -c_2^\alpha (p^{\mathrm{c}}(t))^\top L_{(n_d)}^\epsilon \big(q^{\mathrm{c}}(t)\big) p^{\mathrm{c}}(t) - c_2^\gamma (p^{\mathrm{c}}(t))^\top p^{\mathrm{c}}(t) \\ &< 0.\end{aligned}$$

The proofs of statements 2-4 proceed along the lines of reasoning in the proof in Theorem 5.2. ■

Remark. Theorem 5.3 only shows convergence to the minima of $c_1^\alpha V_\alpha^\epsilon(q^{\mathrm{c}}) + c_1^\gamma V_\gamma(q^{\mathrm{c}})$, which is not necessarily a (quasi) ϵ-lattice. However, assuming that every minimum of $c_1^\alpha V_\alpha^\epsilon(q^{\mathrm{c}}) + c_1^\gamma V_\gamma(q^{\mathrm{c}})$ is a flock (Olfati-Saber 2006, Conjecture 1) and there exist gains c_1^γ, $c_1^\gamma > 0$ such that every minimum of $c_1^\alpha V_\alpha^\epsilon(q^{\mathrm{c}}) + c_1^\gamma V_\gamma(q^{\mathrm{c}})$ is a quasi ϵ-lattice with $\eta \ll d$ (Olfati-Saber 2006, Conjecture 2), one can conjecture that every configuration $q^{\mathrm{c}*}$ in statement 2 in Theorem 5.3 is a quasi ϵ-lattice (Olfati-Saber 2006, Theorem 4).

5.3.3 Analysis of Elliptic Flocking with Obstacle Avoidance

For the analysis of elliptic flocking with obstacle avoidance, consider again a linear γ component in the form of (5.23) with γ-agent dynamics

$$\begin{aligned}\dot{q}_{\mathrm{r}_i}(t) &= p_{\mathrm{r}_i}(t), \\ \dot{p}_{\mathrm{r}_i}(t) &= 0,\end{aligned} \tag{5.31}$$

with $(q_{\mathrm{r}_i}(0), p_{\mathrm{r}_i}(0)) = (q_{\gamma,0}, p_{\gamma,0})$. The Hamiltonian can then be stated as

$$H_\beta(q(t), p(t)) = c_1^\alpha V_\alpha^\epsilon(q(t)) + c_1^\beta V_\beta^\epsilon(q(t)) + c_1^\gamma V_\gamma(q(t)) + K(p(t)), \tag{5.32}$$

with

$$V_\gamma(q(t)) = \frac{1}{2} \sum_{i \in \mathcal{V}} \|q_i(t) - q_{\mathrm{r}_i}(t)\|^2, \tag{5.33}$$

and

$$K(p(t)) = \frac{1}{2} \|p(t)\|^2. \tag{5.34}$$

The theorem for flocking with obstacle avoidance can then be stated similar to Olfati-Saber (2006, Theorem 5).

Theorem 5.4. *Consider a group of agents governed by the dynamics in* (2.22) *applying the protocol in* (5.19)*. Let T satisfy Assumption 5.1 and $T_\beta = I$. Assume that the group objective is given by* (5.23) *with γ-agent dynamics* (5.31)*. If there exists a finite time $t_0 \geq 0$ such that the average velocity of all agents satisfies*

$$\frac{N}{2} (p_{ave}(t))^\top p_{r_i}(t) \leq K(p(t)), \quad \forall t \geq t_0, \tag{5.35}$$

then the energy of the system is monotonically decreasing along the trajectory of the collective dynamics for all $t \geq t_0$.

Proof. The proof follows the same line of reasoning as the proof of Olfati-Saber (2006, Theorem 5). Taking the derivative of the Hamiltonian in (5.32) and simplifying the resulting equation yields

$$\dot{H}_\beta(q(t),p(t)) = -c_2^\alpha \frac{1}{2} \sum_{(i,j)\in\mathcal{E}^\epsilon(q(t))} a_{ij}^\epsilon(q(t))\|p_j(t)-p_i(t)\|^2 \tag{5.36}$$

$$+ c_1^\beta \sum_{i\in\mathcal{V}} \sum_{o\in\mathcal{V}^\beta} \frac{\phi_\beta(\sigma_s(T_\beta q_{io}(t)))}{\sqrt{1+s\|T_\beta q_{io}(t)\|^2}} \left(T_\beta^\top T_\beta q_{io}(t)\right)^\top p_o^{\beta_i}(t) \tag{5.37}$$

$$+ c_2^\beta \sum_{i\in\mathcal{V}} \sum_{o\in\mathcal{V}^\beta} \rho_h\left(\frac{\sigma_s(T_\beta q_{io}(t))}{d_\beta}\right) (p_i(t))^\top p_{io}(t) \tag{5.38}$$

$$- c_2^\gamma \sum_{i\in\mathcal{V}} (p_i(t))^\top (p_i(t)-p_{\mathrm{r}_i}(t)). \tag{5.39}$$

To show that this derivative is non-positive, consider the four terms (5.36)-(5.39) individually. The term in (5.36) is non-positive by definition. For $T_\beta = I$, q_{io} and $p_o^{\beta_i}$ are orthogonal (Section 2.3.2) resulting in (5.37) being zero. Non-positivity of (5.38) is established by Olfati-Saber (2006, Lemma 5). Equation (5.39) can be rewritten as

$$\sum_{i\in\mathcal{V}} (p_i(t))^\top (p_i(t)-p_{\mathrm{r}_i}(t)) = 2\left(K(p(t)) - \frac{N}{2}(p_{\mathrm{r}_i}(t))^\top p_{\mathrm{ave}}(t)\right).$$

It therefore follows with (5.35) that (5.39) is also non-positive. As it holds for the terms (5.36)-(5.39) that they are less than or equal to zero, it follows that $\dot{H}_\beta(q(t),p(t)) \leq 0$ for all $t \geq t_0$. ■

Remark. Theorem 5.4 only holds for $T_\beta = I$, corresponding to spherical obstacle avoidance. For $T_\beta \neq I$, (5.37) is not necessarily non-positive as $q_{io}^\top T_\beta^\top T_\beta p_{io}(t) = 0$ not necessarily holds, which can result in $\dot{H}_\beta(q(t),p(t)) > 0$.

5.4 Simulation Results

In this section, the performance of the proposed elliptic flocking algorithms is demonstrated in three simulation scenarios. In the first scenario (Section 5.4.2), the performance of the proposed scheme is compared to that of the algorithm presented in Wang et al. (2022). Since the elliptic Wang flocking is only applicable to 2D scenarios, this scenario will focus on flocking with group objectives for desired formations with different levels of eccentricity in 2D. The second scenario (Section 5.4.3) then demonstrates the performance of the proposed scheme in 3D scenarios, which is one of the main advantages of the proposed scheme compared to the one presented in Wang et al. (2022). The third scenario (Section 5.4.4) demonstrates that the proposed framework not only works for static transformation

matrices, but also for dynamic ones with $T = T(t)$. Additionally, the elliptical obstacle avoidance capabilities are demonstrated. The algorithms are implemented in discrete time with the agents governed by the double-integrator dynamics in (2.10). All scenarios and algorithms consider the navigational feedback law

$$u_i^\gamma(t) = c_1^\gamma \sigma_1(q_{\mathrm{r}_i}(t) - q_i(t)) + c_2^\gamma(p_{\mathrm{r}_i}(t) - p_i(t)) \tag{5.40}$$

with tuning parameters $c_1^\gamma, c_2^\gamma > 0$.

The simulations are performed using an open-source MATLAB-based simulation framework for MASs (Hespe et al. 2024). Simulation and tuning parameters are provided in Appendix A.2.

5.4.1 Performance Measures

To evaluate and compare the performance of different algorithms and scenarios, performance measures for elliptic flocking are defined analogously to the performance measures introduced in Chapter 4. Convergence to the desired lattice is characterized by the elliptic lattice irregularity

$$J_\alpha^\epsilon(k) = \frac{1}{|\mathcal{E}^\epsilon(q_k)|} \sum_{(i,j)\in\mathcal{E}^\epsilon(q_k)} \left(\|\bar{T}q_{ij,k}\| - \bar{d}\right)^2. \tag{5.41}$$

Furthermore, let

$$J_\alpha^{\epsilon,\Sigma}(t) = \frac{\tau}{t} \sum_{n=0}^{t/\tau} J_\alpha^\epsilon(n) \tag{5.42}$$

denote the integrated elliptic lattice irregularity. The final irregularity is given by $J_\alpha^\epsilon(t_f)$, where t_f is the final simulation time. The speed of convergence is determined by the time it takes to converge towards the ± 0.02-band around $J_\alpha^{\epsilon,\Sigma}(t_f)$. This time is denoted as $t_{0.02}$. Furthermore, the maximum edge irregularity is given by

$$\eta_{\max}(t) = \max_{(i,j)\in\mathcal{E}^\epsilon(q(t))} |\|\bar{T}q_{ij}(t)\| - \bar{d}|, \tag{5.43}$$

with $\eta_{\max}(t_f)$ denoting the final maximum edge irregularity. As in Chapter 4, the required input energies are compared with respect to the average energy of the input signals of all agents u_{E}, as defined in (4.18).

5.4.2 Scenario 1: 2D Performance Comparison

In this scenario, the performance of the proposed elliptic flocking algorithm is compared to that of the algorithm in Wang et al. (2022) in 2D for desired formations with different eccentricities, applying the algorithm proposed in Section 5.2.3. Consider a group of $N = 20$ agents initialized randomly with initial positions $q_i(0) \in [-20, 20]^2$, $p_i(0) = 0_2$, and a static reference $q_{\mathrm{r}_i} = 0_2$, $p_{\mathrm{r}_i} = 0_2$.

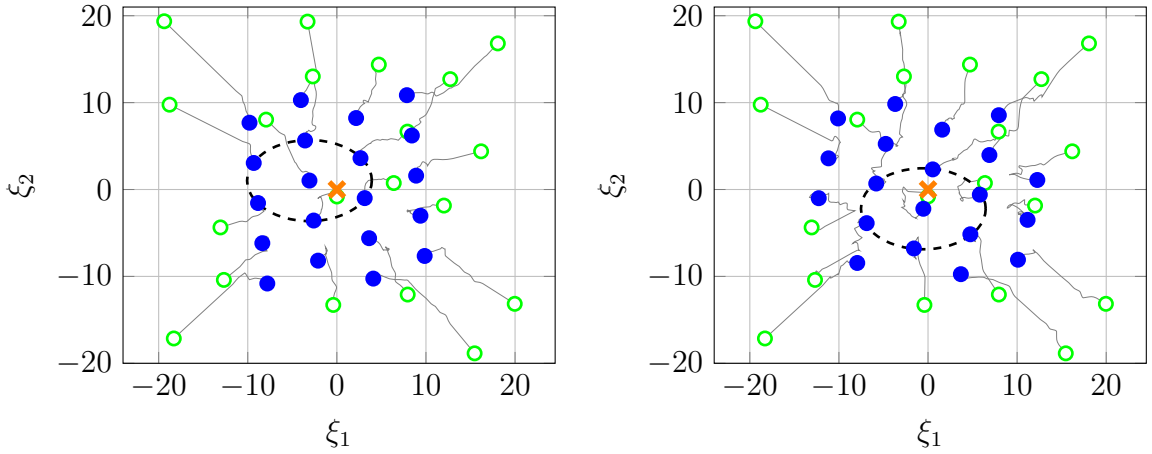
Small Eccentricity

First, consider a scenario with a small eccentricity with $\frac{\lambda_2}{\lambda_1} = \frac{2}{3}$ and $\theta = 0$. With $\lambda_1 = 7$, $d = 1$, and $r = 1.2$, the normalized ϵ-lattice parameters are given by

$$\bar{T} \approx \begin{bmatrix} 0.816 & 0 \\ 0 & 1.224 \end{bmatrix}, \quad \bar{d} \approx 5.715, \quad \bar{r} \approx 6.859.$$

For both algorithms, the trajectories of the agents are depicted in Fig. 5.6 with the initial and final positions indicated by green and blue dots, respectively. The reference position is indicated in orange. Additionally, to visualize the desired formation, the desired elliptic separation is indicated by a dashed black line for one of the agents. The lattice irregularities are depicted in Fig. 5.7. Performance results for this scenario are summarized in Table 5.1.

With both algorithms, the agents form a quasi ϵ-lattice due to the presence of the γ-agent. When comparing the performance, the proposed algorithm achieves its final lattice irregularity $J_\alpha^\epsilon(t_f) = 2.4 \times 10^{-3}$ within $t_{0.02} = 18.6\,\mathrm{s}$ and a maximum edge irregularity of $\eta_{\max}(t_f) = 0.08$. On the other hand, using the Wang algorithm, a final lattice irregularity of $J_\alpha^\epsilon(t_f) = 9.2 \times 10^{-3}$ (+283 %) is achieved within $t_{0.02} = 32.5\,\mathrm{s}$ (+75 %), yielding a maximum edge irregularity of $\eta_{\max}(t_f) = 0.17$ (+112 %). This comparison shows that for the small eccentricity, the proposed algorithm outperforms the one presented in Wang et al. (2022) in terms of lattice irregularity and convergence speed. Additionally, the proposed EF-algorithm achieves this superior performance using 37 % less input energy. This difference in performance can be explained by the different definitions of the flocking control laws and the desired configurations and deviations.



(a) Proposed algorithm.

(b) Algorithm in Wang et al. (2022).

Figure 5.6: Agent trajectories with markers for initial positions (green) and final positions (blue) for a 2D-formation with small eccentricity. The reference is marked in orange.

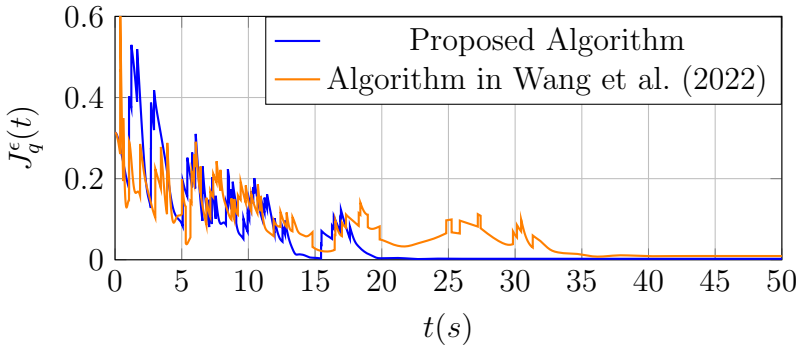


Figure 5.7: ϵ-lattice irregularities for a desired 2D-formation with small eccentricity.

Table 5.1: Scenario 1 evaluation summary.

	Small Eccentricity		Large Eccentricity	
	Proposed	Wang et al. (2022)	Proposed	Wang et al. (2022)
$J_\alpha^\epsilon(t_f)$	2.4×10^{-3}	9.2×10^{-3}	6.5×10^{-3}	1.24
$t_{0.02}$	18.6 s	32.5 s	19.3 s	68.79 s
$\eta_{\max}(t_f)$	0.08	0.17	0.18	3.13
u_{E}	13.01	20.71	15.87	65.55

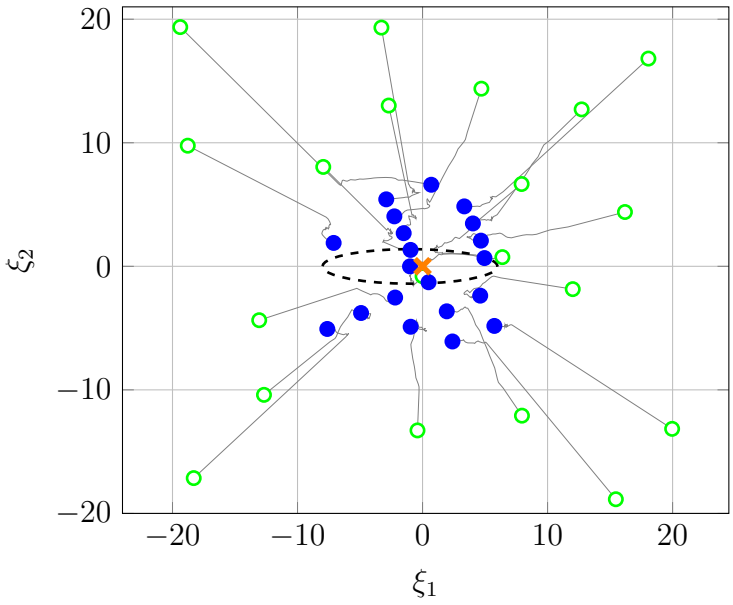
Large Eccentricity

Consider now a scenario with a large eccentricity with $\frac{\lambda_2}{\lambda_1} = 0.2$ and $\theta = 0$. As before, with $\lambda_1 = 7$, $d = 1$, and $r = 1.2$, the normalized ϵ-lattice parameters are given by

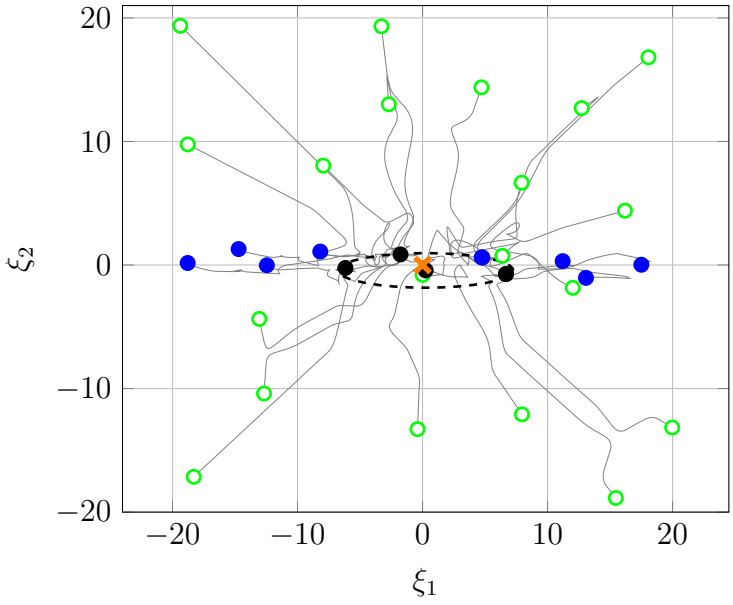
$$\bar{T} \approx \begin{bmatrix} 0.447 & 0 \\ 0 & 2.236 \end{bmatrix}, \quad \bar{d} \approx 3.13, \quad \bar{r} \approx 3.757.$$

The trajectories for the large eccentricity are displayed in Fig. 5.8 and the lattice irregularities in Fig. 5.9. Using the algorithm presented in this chapter, the swarm again converges towards the desired formation with a final lattice irregularity $J_\alpha^\epsilon(t_f) = 6.5 \times 10^{-3}$ within $t_{0.02} = 19.3\,\mathrm{s}$ and a maximum edge irregularity of $\eta_{\max}(t_f) = 0.18$. Examining the Wang algorithm's simulation results, the swarm no longer converges towards the desired quasi ϵ-lattice but several agents converge to the same locations. In Fig. 5.8b these agents are indicated with black markers. Due to this behavior, the resulting final irregularity of the Wang scheme is very large ($J_q^\epsilon(t_f) = 1.24$) with a maximum edge irregularity of $\eta_{\max}(t_f) = 3.13 = \bar{d}$. The results of this simulation scenario are also summarized in Table 5.1. This result can be explained by the definition of the elliptic interaction range. In the algorithm proposed in this chapter, the proximity graph is adjusted for the desired elliptic formation (Section 5.2.1) and an elliptic cut-off is applied to the control laws. However, in Wang et al. (2022), the interaction graph and cut-off are spherical. Hence, for large eccentricities, agents interact not only with their direct neighbors but also with their neighbors' neighbors along the shorter semi-axis, resulting in a non-planar graph.

Consequently, the Wang algorithm can only be applied to scenarios with desired formations with small eccentricities. More precisely, it has to hold that $2\lambda_2 > r$ in order for the agents to only interact with their direct neighbors.

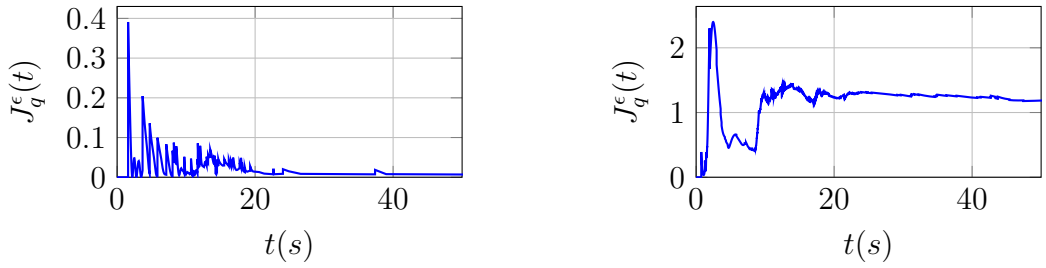


(a) Proposed algorithm.



(b) Algorithm in Wang et al. (2022).

Figure 5.8: Agent trajectories with markers for initial positions (green) and final positions (blue) for a 2D-formation with large eccentricity. Agents converging towards the same positions are indicated in black. The reference is marked in orange.



(a) Proposed algorithm.

(b) Algorithm in Wang et al. (2022).

Figure 5.9: ϵ-lattice irregularities for a desired 2D-formation with large eccentricity.

5.4.3 Scenario 2: 3D Simulation Example

One main advantage of the algorithms presented here is that they can also be applied to 3D scenarios. In the following, the performance of the proposed algorithm will be demonstrated in a 3D scenario with $N = 20$ agents and navigational feedback in the form of (5.40). The reference is given by $q_{\mathrm{r}_i} = 0_3$, $p_{\mathrm{r}_i} = 0_3$.

Consider a desired ϵ-lattice with $\frac{\lambda_2}{\lambda_1} = 0.8$ and $\frac{\lambda_3}{\lambda_1} = 0.4$ without rotation ($R(\theta_1, \theta_2, \theta_3) = I_3$). With $\lambda_1 = 7$, $d = 1$, and $r = 1.2$, the normalized ϵ-lattice parameters are given by

$$\bar{T} \approx \begin{bmatrix} 0.684 & 0 & 0 \\ 0 & 0.855 & 0 \\ 0 & 0 & 1.71 \end{bmatrix}, \quad \bar{d} \approx 4.788, \quad \bar{r} \approx 5.746.$$

The 3D trajectories of the agents are depicted in Fig. 5.10, and the lattice irregularity is provided in Fig. 5.11. Performance results for this scenario are summarized in Table 5.1.

Under the flocking protocol in Section 5.2.3, the agents form a 3D quasi ϵ-lattice with a final irregularity of $J_\alpha^\epsilon(t_f) = 1.7 \times 10^{-3}$ and a maximum edge irregularity of $\eta_{\max}(t_f) = 0.11$. The increase of the lattice irregularity around 70 s is caused by slight adjustments to the lattice structure, reducing the potential introduced by the nonlinear γ-agent.

5.4.4 Scenario 3: Time-Varying Desired Formations and Obstacle Avoidance

So far, the ellipsoidal transformation matrix T has been considered to be constant. This scenario demonstrates that the presented framework can also handle time-varying transformation matrices $T = T(t)$. Consider a γ-agent with a reference position $q_{\mathrm{r}_i}(t)$ describing a 300° circular arc with radius 40. The reference velocity $p_{\mathrm{r}_i}(t)$ is set such that the γ-agent reaches its target point within 100 s. The desired formation is then specified to have a constant ratio between the semi-axis $\frac{\lambda_2}{\lambda_1} = 0.5$, i.e. $\Lambda = \Lambda(\lambda_1, \lambda_2)$ and a time-varying rotation matrix such that the major axis is always pointing in the direction of p_γ, resulting in a time-varying rotation matrix. The transformation matrix $T(t)$ is then given by $T(t) = (R(t)\Lambda)^{-1}$. Additionally, an obstacle of radius 4 is placed on the circular

reference arc. As in the previous scenarios, the lattice parameters are set to $\lambda_1 = 7$, $d = 1$, and $r = 1.2$. For the obstacle avoidance, the corresponding parameters are set to $T_\beta = I_2$, $d_\beta = 6$, and $r_\beta = 8.4$.

In Fig. 5.12, the agents' positions are indicated for different times of the mission. White-Green and blue circles denote the initial and final configurations ($t_f = 150\,\mathrm{s}$), and the white-blue and blue triangles denote the positions at 46 s and 78 s, respectively. The reference trajectory is indicated in orange. The desired elliptic formations at the different stages of the mission show that the swarm is able to track the time-varying reference successfully while also avoiding the obstacle along the way.

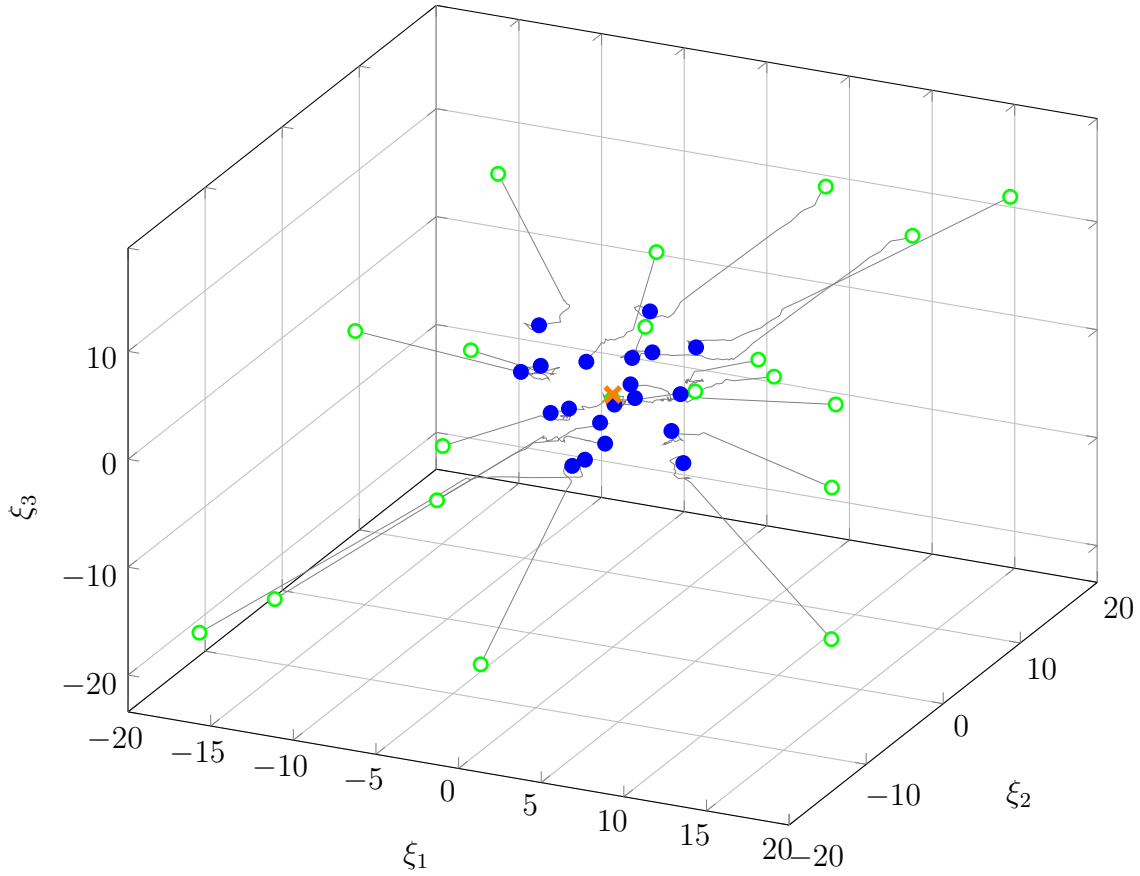


Figure 5.10: Agent trajectories with markers for initial positions (green) and final positions (blue) for a desired 3D-formation.

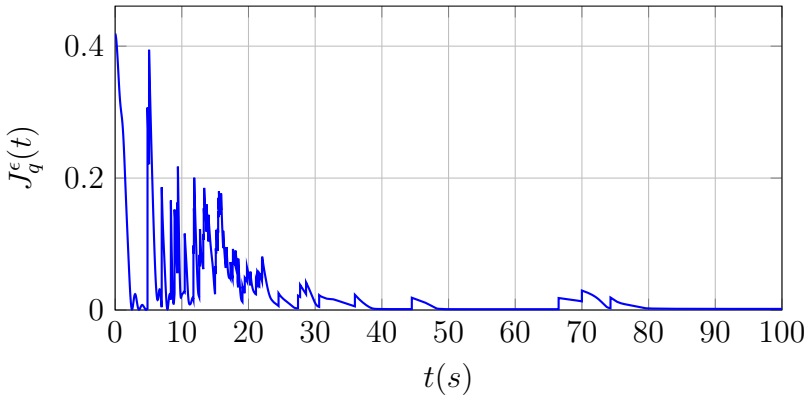


Figure 5.11: ϵ-lattice irregularity for a desired 3D-formation.

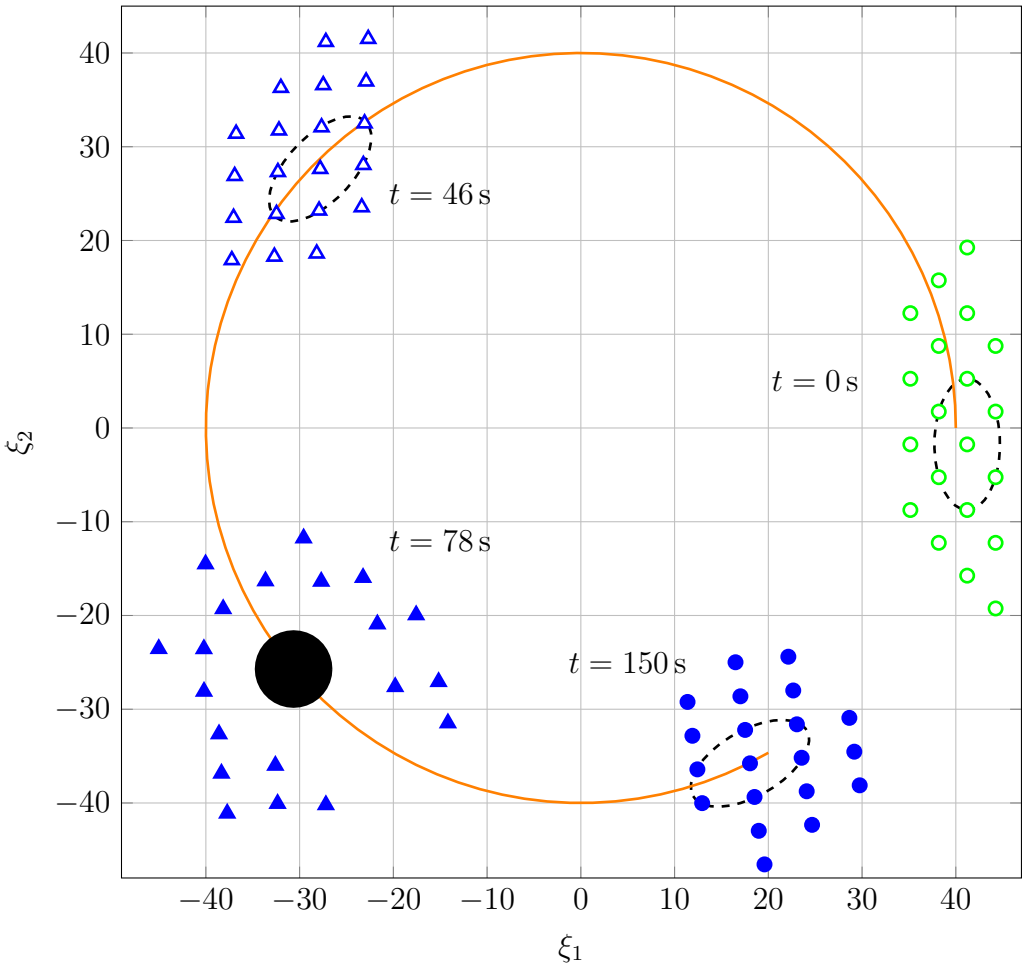


Figure 5.12: Agent trajectories with markers for initial positions (green) and final positions (blue). Positions at intermediate times are indicated by triangles. The orange line represents the reference trajectory.

6 Elliptic Model Predictive Flocking

In the previous chapters, novel frameworks for elliptic flocking (EF) and model predictive flocking (MPF) have been proposed. In this chapter, the advantages of both frameworks are combined to create a framework for elliptic model predictive flocking (EMPF). Especially for the scenarios described in Chapter 5, such as autonomous driving or swarm control of multicopters, EMPF allows for the combination of the features of elliptic flocking with the safety and optimality of predictive flocking. So far, this topic has not been addressed in the literature.

The remainder of this chapter is structured as follows: In Section 6.1, the elliptic distributed model predictive flocking (EDMPF) framework is introduced by combining the frameworks proposed in Sections 4.2 and 5.2. In Section 6.2, the analysis for centralized model predictive flocking presented in Section 3.5 is extended to elliptic centralized model predictive flocking (ECMPF). Finally, in Section 6.3, the performance of the proposed EMPF framework is validated through simulation and compared to the performance of the non-predictive EF framework proposed in Chapter 5.

6.1 Elliptic Model Predictive Flocking Framework

This section introduces the EDMPF framework. Similar to Chapters 4 and 5, the free flocking scenario is considered first (α-EDMPF). This is followed by an extension to flocking with group objectives (α, γ-EDMPF) and flocking with elliptic obstacle avoidance (α, β, γ-EDMPF).

Following the notation introduced in Chapter 5, the desired ϵ-lattice is given by Definition 5.1 and the elliptic transformation matrix is denoted by T. Recall that T is assumed to be invertible (Assumption 5.1). Furthermore, the elliptic communication topology is defined as $\mathcal{G}^\epsilon(q) = (\mathcal{V}, \mathcal{E}^\epsilon(q))$ with the elliptic edge set

$$\mathcal{E}^\epsilon(q) = \{(i,j) \in \mathcal{V} : \|Tq_{ij}\| < r,\ j \neq i\},$$

and the elliptic neighborhood

$$\mathcal{N}_i^\epsilon(q) = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}^\epsilon\}.$$

6.1.1 Optimization and Neighbor Trajectory Estimation

As in Chapter 4, the resulting nonlinear optimization problems derived in the remainder of this chapter are solved using sequential quadratic programming (SQP), which is terminated

after a specified number of N_{SQP} iterations. Following the discussion in Section 4.2.1 and the results of the parameter studies in Section 4.4, the trajectories of neighboring agents are computed according to (3.16), assuming zero input sequences ($\hat{U}_j = 0$).

6.1.2 α-Elliptic Distributed Model Predictive Flocking

Consider the stage cost of the form

$$\begin{aligned}\ell_i^{\epsilon,\alpha}(x_{i,k+m|k}, \hat{x}_{\bar{i},k+m|k}, u_{i,k+m|k}) = \|u_{i,k+m|k}\|_{R_u}^2 + \|p_{i,k+m|k} - \bar{p}_{i,k}^{\epsilon,\mathrm{ave}}\|_{R_p}^2 \\ + \sum_{j\in\mathcal{N}_{i,k}^{\epsilon}} r_\delta^\epsilon(q_{i,k+m|k}, \hat{q}_{j,k+m|k})\|\delta^\epsilon(q_{i,k+m|k}, \hat{q}_{j,k+m|k})\|^2 \quad ,\end{aligned} \tag{6.1}$$

where the elliptic deviation is defined as

$$\delta^\epsilon(q_i, q_j) = \delta_{ij}^\epsilon = \|Tq_{ij}\| - d.$$

Similar to the average velocity in Section 4.2.2, the average velocity of agent i and the agents in the elliptic neighborhood set $\mathcal{N}_i^\epsilon$ is given by

$$\bar{p}_{i,k}^{\epsilon,\mathrm{ave}} = \frac{1}{|\mathcal{N}_{i,k}^\epsilon| + 1}\Big(p_{i,k} + \sum_{j\in\mathcal{N}_{i,k}^\epsilon} \hat{p}_{j,k}\Big). \tag{6.2}$$

As derived in Section 4.1, the state-dependent flocking gain

$$r_\delta^\epsilon(q_i, q_j) = \frac{r_\delta^+ - r_\delta^-}{2}\tanh\left(\frac{\delta_{ij}^\epsilon}{\epsilon_\alpha}\right) + \frac{r_\delta^+ + r_\delta^-}{2} \tag{6.3}$$

enables the independent tuning of attractive and repulsive inter-agent forces by defining the continuously differentiable weight analogously to (4.3).

By defining the α-EDMPF cost function as

$$V_i^{\epsilon,\alpha}\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) = \sum_{m=0}^{M-1} \ell_i^{\epsilon,\alpha}(x_{i,k+m|k}, \hat{x}_{\bar{i},k+m|k}, u_{i,k+m|k}) + \ell_i^{\epsilon,\alpha}(x_{i,k+M|k}, \hat{x}_{\bar{i},k+M|k}, 0), \tag{6.4}$$

with constraint sets $\bar{\mathbb{X}}_i$ and $\mathcal{U}_i$ as defined in Section 2.4, the free flocking EDMPF problem can then be stated as follows:

Problem 6.1 (α-EDMPF)**.**

$$\begin{aligned}V_i^{\epsilon*}(x_{i,k}, \hat{X}_{\bar{i},k}^M) = &\min_{U_{i,k}} V_i^{\epsilon,\alpha}\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) \\ &\text{s.t. } x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ &\quad (X_{i,k}^M, \hat{X}_{\bar{i},k}^M) \in \bar{\mathbb{X}}_i^{M+1}, \\ &\quad U_{i,k} \in \mathcal{U}_i.\end{aligned}$$

The last term in (6.4) can be considered the terminal cost, and the terminal constraint is included in the state constraints with $\mathbb{X}_{i,f} = \bar{\mathbb{X}}_i$.

As described in Section 5.2.2, the desired ϵ-lattice is not uniquely defined by the parameters (r, d, T). Therefore, as in the non-predictive framework, the normalized parameters $(\bar{r}, \bar{d}, \bar{T})$ that are calculated according to (5.7) are used.

6.1.3 α, γ-Elliptic Distributed Model Predictive Flocking

Since the objective of tracking the γ-agent is not affected by the elliptic agent-agent interactions, the tracking of group objectives can be integrated into the EDMPF problem as described in the framework for distributed model predictive flocking with asymmetric interaction forces in Section 4.2.3. With the stage cost ℓ_i^γ in (4.8) and the corresponding cost function V_i^γ in (4.9), the α, γ-EDMPF problem is stated as

Problem 6.2 (α-γ-EDMPF)**.**

$$\begin{aligned} V_i^{\epsilon *}(x_{i,k}, \hat{X}_{\bar{i},k}^M) = \min_{U_{i,k}} \; & V_i^{\epsilon,\alpha}\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) + V_i^{\gamma}\big(x_{i,k}, U_{i,k}\big) \\ \text{s.t. } & x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ & (X_{i,k}^M, \hat{X}_{\bar{i},k}^M) \in \bar{\mathbb{X}}_i^{M+1}, \\ & U_{i,k} \in \mathcal{U}_i. \end{aligned}$$

6.1.4 α, β, γ-Elliptic Distributed Model Predictive Flocking

As in the previous descriptions of obstacle avoidance, each of the considered N^β spherical obstacles is characterized by its center state $x_o^\beta = \mathrm{col}(q_o^\beta, 0) \in \mathbb{R}^{2n_\mathrm{d}}$ and radius $r_o^\beta > 0$, $o \in \mathbb{I}_{1:N^\beta}$. Recall that $\mathcal{V}^\beta$ denotes the set of all obstacles. For each agent i, a virtual β-agent is considered at the boundary of the obstacle with position $q_o^{\beta_i}(k)$ (Sections 2.3.2, 4.2.4, 5.2.4). Following the results in Section 5.2.4, by defining an elliptic transformation matrix T_β for obstacle avoidance, let

$$\mathcal{N}_i^{\epsilon,\beta}(q) = \{o \in \mathcal{V}^\beta : \|T_\beta q_{io}\| < r_o\} \tag{5.16}$$

denote the set of obstacles within the obstacle detection range r_o.

The elliptic deviation from the desired obstacle separation d_o is defined as

$$\delta^{\epsilon,\beta}(q_i, q_o^{\beta_i}) = \delta_{io}^{\epsilon,\beta} = \|T_\beta q_{io}\| - d_o. \tag{6.5}$$

Analogous to the asymmetric obstacle avoidance weight proposed in Section 4.2.4, define

$$r_\beta^\epsilon(q_i, q_o) = -\frac{r_\delta^-}{2} \tanh\left(\frac{\delta_{io}^{\epsilon,\beta}}{\epsilon_\beta}\right) + \frac{r_\delta^-}{2} \tag{6.6}$$

as the asymmetric elliptical obstacle avoidance weight. Note that the parameters (r_o, d_o, T_β) do not uniquely characterize the desired obstacle avoidance behavior. As in the non-predictive elliptic flocking protocol described in Section 5.2.4, a unique parameterization

can be obtained by considering the normalized parameters $(\bar{r}_o, \bar{d}_o, \bar{T}_\beta)$ computed according to (5.7).

The EDMPF problem with group objectives and obstacle avoidance can then be stated as follows:

Problem 6.3 (α-β-γ-EDMPF)**.**

$$\begin{aligned} V_i^{\epsilon*}(x_{i,k}, \hat{X}_{\bar{i},k}^M, q_{\bar{o},k}^{\beta_i}) = &\min_{U_{i,k}} V_i^{\epsilon,\alpha}\big(x_{i,k}, \hat{X}_{\bar{i},k}^M, U_{i,k}\big) + V_i^{\epsilon,\beta}\big(x_{i,k}, q_{\bar{o},k}^{\beta_i}, U_{i,k}\big) + V_i^{\gamma}\big(x_{i,k}, U_{i,k}\big) \\ \text{s.t. } & x_i(k+m) = \phi_i(m; x_i(k), U_i(k)), \quad m \in \mathbb{I}_{0:M}, \\ & (X_{i,k}^M, \hat{X}_{\bar{i},k}^M) \in \bar{\mathbb{X}}_i^{M+1}, \\ & U_{i,k} \in \mathcal{U}_i. \end{aligned}$$

Here, the cost associated with obstacle avoidance is given by

$$V_i^{\epsilon,\beta}\big(x_{i,k}, q_{\bar{o},k}^{\beta_i}, U_{i,k}\big) = \sum_{m=0}^{M} \ell_i^{\epsilon,\beta}(x_{i,k+m|k}, q_{\bar{o},k|k}^{\beta_i}),$$

where

$$\ell_i^{\epsilon,\beta}(x_{i,k+m|k}, , q_{o,k|k}^{\beta_i}) = \sum_{o \in \mathcal{N}_{i,k}^{\epsilon,\beta}} r_\beta^\epsilon(q_{i,k+m|k}, q_{o,k|k}^{\beta_i}) \|\delta^\beta(q_{i,k+m|k}, q_{o,k|k}^{\beta_i})\|^2$$

is the corresponding stage cost.

6.2 Stability Analysis of Elliptic Centralized Model Predictive Flocking

Similar to the extension presented in Section 4.3, the stability analysis for centralized model predictive flocking (CMPF) in Section 3.5 can also be extended to elliptic centralized model predictive flocking (ECMPF). Consider a homogeneous MAS with agents governed by the linear discrete-time dynamics in (2.6),

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k),$$

with $x_i = [q_i \; p_i]^\top$. The considered elliptic centralized model predictive flocking (ECMPF) cost function is given by

$$V^\epsilon(x(k), U(k)) = \sum_{m=0}^{M-1} \ell^\epsilon(x_{k+m|k}, u_{k+m|k}),$$

with stage cost

$$\begin{aligned} \ell^\epsilon(x_{k+m|k}, u_{k+m|k}) = &\|u_{k+m|k}\|_{R_u}^2 + \|p_{k+m|k} - p_{\mathrm{r}}\|_{R_p}^2 \\ &+ \sum_{(i,j) \in \mathcal{E}^\epsilon(q_k)} r_\delta^\epsilon(q_{i,k+m|k}, q_{j,k+m|k}) \|\delta^\epsilon(q_{i,k+m|k}, q_{j,k+m|k})\|^2. \end{aligned}$$

Recall that $p_\mathrm{r} = 1_N \otimes p_{\mathrm{r}_i}$ denotes the collective reference velocity with $p_{\mathrm{r}_i} \in \mathbb{R}^{n_\mathrm{d}}$. The input constraints are defined as in (3.4),

$$\mathcal{U} = \left\{ U \in \mathbb{R}^{Mn_u} : \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes I_{Mn_u} U - \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix} \otimes 1_{Mn_u} \leq 0 \right\},$$

and no state constraints are imposed ($\mathcal{X} = \mathbb{R}^{Mn_x}$). As in Section 3.5, the communication topology is assumed to be static.

Assumption 6.1 (Static elliptic communication topology)**.** Given an initial state x_0, it holds that

$$\mathcal{G}\big(\mathcal{V}, \mathcal{E}^\epsilon(q_k)\big) = \mathcal{G}\big(\mathcal{V}, \mathcal{E}^\epsilon(q_0)\big), \quad \forall k \geq 0.$$

Then, as in Section 3.5, the terminal cost and the terminal control law are chosen as $V_f = 0$ and $\kappa_M = 0$. The terminal set is given by the set of all admissible ϵ-lattices at time k, given by

$$\mathcal{A}^\epsilon(x_0) = \mathcal{A}_0^\epsilon = \{x \in \mathbb{X} : \ \|Tq_{ij}\| = d \ \forall (i,j) \in \mathcal{E}^\epsilon(q_0), \ \|Tq_{ij}\| \geq r \ \forall (i,j) \notin \mathcal{E}^\epsilon(q_0), \ p = p_\mathrm{r}\}. \tag{6.7}$$

The ECMPF problem is then stated as follows:

Problem 6.4.

$$\begin{aligned} V^{\epsilon *}(x(k)) = & \min_{U(k)} V^\epsilon(x(k), U(k)) \\ \text{s.t. } & x(k+m) = \phi(m; x(k), U(k)), \quad m \in \mathbb{I}_{0:M}, \\ & X(k) \in \mathcal{X}, \\ & U(k) \in \mathcal{U}, \\ & x(k+M) \in \mathcal{A}_0^\epsilon. \end{aligned}$$

For the analysis, as in Section 3.5, the problem is assumed to be initially feasible, the region of attraction $\mathbb{X}_M$ is assumed to be nonempty, and the communication topology is assumed to be connected.

Assumption 6.2. $\mathcal{A}_0^\epsilon$ and $\mathbb{X}_M$ are nonempty and $\mathcal{G}\big(\mathcal{V}, \mathcal{E}^\epsilon(q_0)\big)$ is connected.

Closedness of $\mathcal{A}_0^\epsilon$ can be established analogously to Lemma 3.11.

Lemma 6.1. *Suppose Assumptions 5.1, 3.10, and 3.11 hold for $\mathcal{A}_0^\epsilon$. Then, the set $\mathcal{A}_0^\epsilon$ is closed and positive invariant for $x_{k+1} = f(x_k, \kappa(x_k))$ under control law* (2.51)*.*

The proof follows the same line of reasoning as the proof of Lemma 3.11.

Stability is guaranteed by the following theorem similar to Theorem 3.12.

Theorem 6.2 (ECMPF stability)**.** *Suppose that Assumptions 6.1 and 6.2 hold. Then, for $x(0) \in \mathbb{X}_M$, Problem 6.4 is recursively feasible and the agents asymptotically converge towards an ϵ-lattice with $p = p_\mathrm{r}$ under the control law* (2.51)*.*

Proof. The proof follows the same line of reasoning as the proof of Theorem 3.12. By showing that Assumptions 2.1, 3.9, 3.10, 3.11, 3.12, and 3.13 are satisfied for Problem 6.4 and $\mathcal{A}_0^\epsilon$, Theorem 3.9 can be applied. Recursive feasibility can also be established analogously. ■

6.3 Numerical Example

In this section, the performance of the proposed EDMPF algorithms is demonstrated in simulation and the performance is compared to the non-predictive EF algorithm presented in Chapter 5. Of the scenarios presented so far, one in which an elliptical desired formation is expected to significantly improve performance is a scenario with a field of small obstacles, similar to the one considered in Section 4.5. By reducing the desired inter-agent distance in the direction perpendicular to the main direction of movement, the agents are expected to flow smoother through the field of obstacles.

6.3.1 Setup

Consider a swarm of $N = 20$ agents, with $n_\mathrm{d} = 2$, governed by the discrete-time double-integrator dynamics in (2.10). The agents are tasked with clearing a field of eight small obstacles with radii $r_o^\beta = 1$, $o \in \mathbb{I}_{1:8}$, and center positions

$$q_1^\beta = \begin{bmatrix} 30 \\ 0 \end{bmatrix}, \; q_2^\beta = \begin{bmatrix} 30 \\ 14 \end{bmatrix}, \; q_3^\beta = \begin{bmatrix} 30 \\ -14 \end{bmatrix}, \; q_4^\beta = \begin{bmatrix} 45 \\ 7 \end{bmatrix},$$
$$q_5^\beta = \begin{bmatrix} 45 \\ -7 \end{bmatrix}, \; q_6^\beta = \begin{bmatrix} 60 \\ 0 \end{bmatrix}, \; q_7^\beta = \begin{bmatrix} 60 \\ 14 \end{bmatrix}, \; q_8^\beta = \begin{bmatrix} 60 \\ -14 \end{bmatrix}.$$

Agents are initialized around the origin with $p(0) = 0$. The reference is set to $q_{\mathrm{r}_i} = [90\ 0]^\top$ and $p_{\mathrm{r}_i} = 0$. Note that, compared to the setup described in Section 4.5, the obstacles are much closer together in the ξ_2-direction. To allow the agents to safely pass this field of obstacles, consider the ϵ-lattice and elliptic obstacle avoidance parameters

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad d = 7, \quad r = 8.4,$$
$$T_\beta = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad d_o = 6, \quad r_o = 8.4.$$

According to (5.7), the corresponding normalized parameters are then given by

$$\bar{T} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} \end{bmatrix}, \quad \bar{d} = \frac{7}{\sqrt{2}}, \quad \bar{r} = \frac{8.4}{\sqrt{2}},$$
$$\bar{T}_\beta = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} \end{bmatrix}, \quad \bar{d}_o = \frac{6}{\sqrt{2}}, \quad \bar{r}_o = \frac{8.4}{\sqrt{2}}.$$

For reference tracking of the non-predictive algorithm, consider again the navigational feedback law

$$u_i^\gamma(t) = c_1^\gamma \sigma_1(q_{\mathrm{r}_i}(t) - p_i(t)) + c_2^\gamma(p_{\mathrm{r}_i}(t) - p_i(t)).$$

To allow a fair comparison, the algorithms were tuned to use similar amounts of input energy. The tuning parameters for the compared algorithms are provided in Appendix A.3.

6.3.2 Performance Measures

To compare the performances, consider again the elliptic lattice irregularity $J_\alpha^\epsilon(k)$ and the integrated elliptic lattice irregularity $J_\alpha^{\epsilon,\Sigma}(t)$ as defined in Section 5.4. The speed of convergence is again indicated by $t_{0.02}$, the time it takes the agents to converge towards the ± 0.02-band around $J_\alpha^\epsilon(k_f)$. As in Chapters 4 and 5, the required input energies are compared with respect to the average energy of the input signals of all agents u_{E}, defined in (4.18). To quantify the obstacle avoidance performance in the elliptic frameworks, let

$$J_\beta^\epsilon(k) = \sum_{i\in\mathcal{V}} \left(\frac{1}{|\mathcal{N}_{i,k}^{\epsilon,\beta}|} \sum_{o\in\mathcal{N}_{i,k}^{\epsilon,\beta}} \min\left(\|\bar{T}_\beta q_{io,k}\| - \bar{d}_o, 0\right)^2 \right) \tag{6.8}$$

and

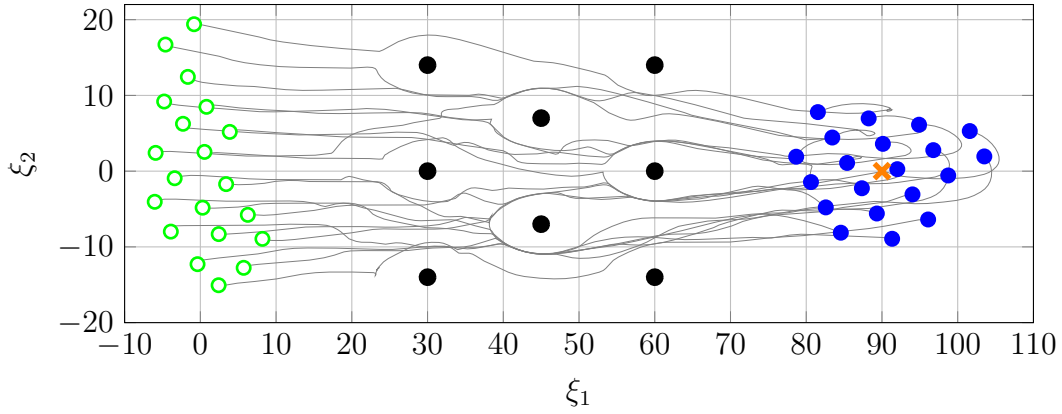
$$J_\beta^{\epsilon,\Sigma}(t) = \frac{\tau}{t} \sum_{n=0}^{t/\tau} J_\beta^\epsilon(n). \tag{6.9}$$

denote the elliptic obstacle avoidance performance and integrated elliptic obstacle avoidance performance, respectively. As in Chapter 4, the obstacle clearance time t_{obs} is given by the time difference between the first and last time any of the obstacles is detected by one of the agents.

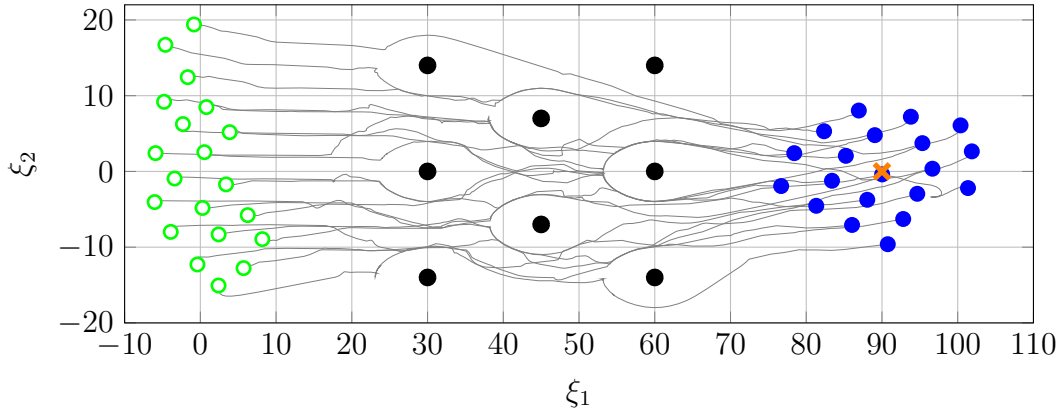
6.3.3 Results

For both algorithms, the resulting agents' trajectories are depicted in Fig. 6.1. With both algorithms, the agents are able to pass the field of obstacles successfully and form a quasi ϵ-lattice at the desired location. When comparing the performance with respect to the measures defined in Section 6.3.2, the results are similar to the comparison of the ADMPF and Olfati-Saber algorithms in Section 4.5. The results are summarized in Table 6.1.

Under the predictive algorithm, the agents clear the field of obstacles significantly faster and also converge to the desired location about three times faster compared to the non-predictive EF algorithm. Furthermore, the EDMPF scheme shows a better performance with respect to the final lattice irregularity $J_\alpha^\epsilon(k_f)$ and the integrated lattice irregularity $J_\alpha^{\epsilon,\Sigma}(t_{0.02})$. When comparing the obstacle avoidance performance, the EF scheme shows an integrated elliptic obstacle avoidance performance that is even better than that of the predictive scheme. Overall, this scenario demonstrates that the general advantages of MPF over non-predictive flocking also hold for scenarios with elliptic desired formations.



(a) Agent trajectories under EDMPF algorithm.



(b) Agent trajectories under EF algorithm presented in Chapter 5.

Figure 6.1: Agent trajectories for different algorithms with initial positions indicated in green, final positions in blue, and the reference in orange.

Table 6.1: Summary for EDMPF performance comparison.

	$t_{0.02}$	t_{obs}	u_{E}	$J_{\alpha}^{\epsilon}(k_f)$	$J_{\alpha}^{\epsilon,\Sigma}(t_{0.02})$	$J_{\beta}^{\epsilon,\Sigma}(t_{0.02})$
EDMPF	93.2 s	66.2 s	4.59	0.003	5.55	6.1
EF	280.8 s	194 s	4.31	0.01	10.04	4.52

7 Conclusion and Future Work

In this chapter, the results of this thesis are summarized and future research directions are presented.

7.1 Summary

This thesis addresses several topics related to the analysis and design in the field of model predictive flocking (MPF) and flocking control in general. One of the main contributions is related to the stability analysis of MPF by conducting a detailed investigation of the challenges involved in the MPF analysis and by critically discussing the analysis results in the related literature. The discussion in Chapter 3 demonstrates that the MPF analysis is still an open issue. This is shown by demonstrating that the identified challenges can break the line of reasoning of the stability analysis if not addressed properly, and by pointing out several errors in the existing MPF analysis results based on N-paths (Zhan and Li 2013; Zhang et al. 2015; Zhou and Li 2017) and compatibility constraints (Hu et al. 2018, 2017). Considering these observations, an analysis for centralized model predictive flocking (CMPF) based on the stability of sets of equilibria is proposed. In Chapter 4, a novel MPF framework with asymmetric interaction forces is introduced, enabling independent tuning of attractive and repulsive inter-agent forces, while also allowing for the consideration of obstacle avoidance and group objectives. It is furthermore shown that the analysis presented in Chapter 3 can also be applied to asymmetric centralized model predictive flocking. The performance of the proposed schemes is demonstrated in simulation, showing an improved performance compared to MPF algorithms in the related literature.

Chapter 5 and Chapter 6 focus on the field of elliptic flocking (EF), where the desired distances between agents can be tuned individually in different directions. First, non-predictive EF is addressed in Chapter 5 by formulating the desired behavior as a generalization of the framework proposed in Olfati-Saber (2006), also showing that the stability results in Olfati-Saber (2006) can be extended to the EF framework. The performance of the proposed EF algorithms is then validated in simulation, showing an increased performance compared to existing EF frameworks (Wang et al. 2022). Finally, in Chapter 6, the results presented in the previous chapters are combined into a framework for elliptic model predictive flocking (EMPF) with asymmetric interaction forces, enabling an independent design of desired distances in different directions as well as an independent tuning of attractive and repulsive inter-agent forces. Furthermore, it is shown that the analysis presented in Chapter 3 can also be extended to the analysis of EMPF. The performance of the proposed elliptic model predictive flocking (EMPF) framework is validated in simulation.

In conclusion, this thesis contributes to the analysis of model predictive flocking by addressing the challenges and flaws in the existing approaches, and by providing an analysis framework applicable to MPF, including the extensions to asymmetric and elliptic model predictive flocking. With respect to the design of MPF algorithms, the proposed frameworks for asymmetric and elliptic model predictive flocking allow tailoring of the MPF algorithms to different scenarios, resulting in an increased performance compared to existing frameworks.

7.2 Outlook

While this thesis addresses some of the open questions in the field of model predictive flocking, there are still numerous open problems that are not covered here. Some of these are listed below.

With respect to the analysis of MPF, one major open research question is to extend the analysis presented in Chapter 3 to a less restrictive setting, allowing for time-varying communication topologies and distributed implementations. One possible direction would be to investigate sequential distributed MPC schemes, as for example presented in Müller et al. (2012). However, because of the unknown setpoint and time-varying communication topology in MPF, this approach would require a centralized entity to compute a global terminal control law, which would no longer constitute a purely distributed scheme (Müller et al. 2012). In this context, the application of iterative approaches to MPF could be investigated as these may help addressing the challenges related to the prediction mismatch in distributed MPC (Lyu et al. 2021; Stomberg et al. 2022).

Furthermore, existing frameworks for MPF mainly consider ideal scenarios without effects such as model uncertainties, external disturbances, or communication delays in combination with safety-related constraints. However, to successfully utilize MPF in experimental scenarios, such effects should be considered in the MPF framework, for example by the use of robust control techniques (Rawlings et al. 2020).

In the context of elliptic flocking, predictive and non-predictive, one of the open challenges is the extension of the analysis to time-varying desired formations. While such schemes can be readily implemented, as demonstrated in Chapter 5, the explicit time dependency poses additional challenges for the analysis. The same holds for extending to state-dependent desired formations, for example by considering desired separations dependent on the relative velocities between agents (Wang et al. 2023).

Finally, it is worth noting that individual aspects of the presented frameworks can also be integrated into other MPC algorithms. For example, the elliptic obstacle avoidance capabilities presented in Chapter 6 have already been adapted into a multi-stage predictive separation algorithm in the context of urban air mobility (Berling et al. 2023).

Bibliography

Athanasiadis, I. N. and P. A. Mitkas (2004). "An agent–based intelligent environmental monitoring system". In: *Management of Environmental Quality: An International Journal* 15.3, pp. 238–249.

Attallah, A., A. Datar, and H. Werner (2020). "Flocking of Linear Parameter Varying Agents: Source Seeking Application with Underwater Vehicles". In: *IFAC-PapersOnLine* 53.2, pp. 7305–7311.

Barve, A. and M. J. Nene (2013). "Survey of Flocking Algorithms in multi-agent Systems". In: *International Journal of Computer Science Issues (IJCSI)* 10.6.

Beaver, L. E., C. Kroninger, and A. A. Malikopoulos (2020). "An Optimal Control Approach to Flocking". In: *2020 American Control Conference*. IEEE, pp. 683–688.

Beaver, L. E. and A. A. Malikopoulos (2020). "Beyond Reynolds: A Constraint-Driven Approach to Cluster Flocking". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 208–213.

Beaver, L. E. and A. A. Malikopoulos (2021). "An overview on optimal flocking". In: *Annual Reviews in Control* 51, pp. 88–99.

Berling, J., P. Hastedt, S. T. Wanniarachchi, A. Vieregg, C. Gertz, V. Turau, H. Werner, and V. Gollnick (2023). *A Modular Urban Air Mobility Simulation Toolchain with Dynamic Agent Interaction*. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V.

Boyd, S. P. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press.

Burmeister, B., A. Haddadi, and G. Matylis (1997). "Application of multi-agent systems in traffic and transportation". In: *IEE Proceedings - Software Engineering* 144.1, p. 51.

Cai, H., Y. Su, and J. Huang (2022). *Cooperative Control of Multi-agent Systems*. Springer International Publishing.

Cannon, M. (2004). "Efficient nonlinear model predictive control algorithms". In: *Annual Reviews in Control* 28.2, pp. 229–237.

Cannon, M., J. Buerger, B. Kouvaritakis, and S. Rakovic (2011). "Robust Tubes in Nonlinear Model Predictive Control". In: *IEEE Transactions on Automatic Control* 56.8, pp. 1942–1947.

Cao, Y., W. Yu, W. Ren, and G. Chen (2013). "An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination". In: *IEEE Transactions on Industrial Informatics* 9.1, pp. 427–438.

Diestel, R. (2017). *Graph Theory*. Vol. 173. Springer Berlin Heidelberg.

Drew, D. S. (2021). "Multi-Agent Systems for Search and Rescue Applications". In: *Current Robotics Reports* 2.2, pp. 189–200.

Dunbar, W. B. and D. S. Caveney (2012). "Distributed Receding Horizon Control of Vehicle Platoons: Stability and String Stability". In: *IEEE Transactions on Automatic Control* 57.3, pp. 620–633.

Dunbar, W. B. and R. M. Murray (2006). "Distributed receding horizon control for multi-vehicle formation stabilization". In: *Automatica* 42.4, pp. 549–558.

Ferrari-Trecate, G., L. Galbusera, M. Marciandi, and R. Scattolini (2009). "Model Predictive Control Schemes for Consensus in Multi-Agent Systems with Single- and Double-Integrator Dynamics". In: *IEEE Transactions on Automatic Control* 54.11, pp. 2560–2572.

Francis, B. A. and M. Maggiore (2016). *Flocking and Rendezvous in Distributed Robotics*. 1st ed. 2016. SpringerLink Bücher. Springer.

Grimm, G., M. J. Messina, S. E. Tuna, and A. R. Teel (2004). "Examples when nonlinear model predictive control is nonrobust". In: *Automatica* 40.10, pp. 1729–1738.

Hastedt, P., A. Datar, K. Kocev, and H. Werner (2024). "Distributed Flocking Control With Ellipsoidal Level Sets". In: *2024 Americal Control Conference (ACC)*, pp. 1018–1023.

Hastedt, P. and H. Werner (2023a). "Distributed Model Predictive Flocking with Obstacle Avoidance and Asymmetric Interaction Forces". In: *2023 American Control Conference (ACC)*, pp. 1177–1182.

Hastedt, P. and H. Werner (2023b). "Nonlinear Distributed Model Predictive Flocking with Obstacle Avoidance". In: *IFAC-PapersOnLine* 56.2, pp. 3794–3799.

Hastedt, P. and H. Werner (2023c). "On Stability Analysis of Predictive Flocking Using N-Paths". In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 5676–5681.

Heemels, W., K. H. Johansson, and P. Tabuada (2012). "An introduction to event-triggered and self-triggered control". In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 3270–3285.

Hespe, C., A. Datar, D. Schneider, H. Saadabadi, H. Werner, and H. Frey (2024). *WiMAS: Wireless Multi-Agent Simulations: v0.1*. Zenodo. URL: `https://doi.org/10.5281/zenodo.11118667`.

Hu, Y., J. Zhan, and X. Li (2018). "Self–triggered distributed model predictive control for flocking of multi–agent systems". In: *IET Control Theory & Applications* 12.18, pp. 2441–2448.

Hu, Y., J. Zhan, Q. Yuan, and X. Li (2017). "A multi-agent flocking system with communication delays via distributed model predictive control". In: *36th Chinese Control Conference (CCC)*, pp. 8449–8454.

Huang, D., Q. Yuan, and X. Li (2019). "Decentralized flocking of multi-agent system based on MPC with obstacle/collision avoidance". In: *2019 Chinese Control Conference (CCC)*, pp. 5587–5592.

Keviczky, T., F. Borrelli, and G. J. Balas (2004). "A study on decentralized receding horizon control for decoupled systems". In: *Proceedings of the 2004 American Control Conference*. IEEE, pp. 4921–4926.

Keviczky, T., F. Borrelli, and G. J. Balas (2006). "Decentralized receding horizon control for large scale dynamically decoupled systems". In: *Automatica* 42.12, pp. 2105–2115.

Keviczky, T., F. Borrelli, and G. J. Balas (2008). "Distributed Predictive Control: Synthesis, Stability and Feasibility". In: *Cooperative control of distributed multi-agent systems*. Ed. by J. S. Shamma. John Wiley & Sons, pp. 79–108.

Khalil, H. K. (2002). *Nonlinear systems*. 3. ed. Prentice Hall.

Kloock, C. and H. Werner (2020). "Prediction Mismatch in Distributed Model Predictive Control". In: *2020 European Control Conference (ECC)*. IEEE, pp. 1224–1229.

Kumar, M., D. P. Garg, and V. Kumar (2010). "Segregation of Heterogeneous Units in a Swarm of Robotic Agents". In: *IEEE Transactions on Automatic Control* 55.3, pp. 743–748.

Luenberger, D. G. and Y. Ye (2016). *Linear and Nonlinear Programming*. Vol. 228. Springer International Publishing.

Lyu, Y., J. Hu, B. M. Chen, C. Zhao, and Q. Pan (2021). "Multivehicle Flocking With Collision Avoidance via Distributed Model Predictive Control". In: *IEEE transactions on cybernetics* 51.5, pp. 2651–2662.

Mesbahi, M. and M. Egerstedt (2010). *Graph Theoretic Methods in Multiagent Networks*. Course Book. Princeton Series in Applied Mathematics. Princeton University Press.

Molčanov, I. S. (2017). *Theory of random sets*. Second edition. Vol. 87. Springer London.

Müller, M. A., M. Reble, and F. Allgöwer (2012). "Cooperative control of dynamically decoupled systems via distributed model predictive control". In: *International Journal of Robust and Nonlinear Control* 22.12, pp. 1376–1397.

Negenborn, R. R. and J. M. Maestre (2014). "Distributed Model Predictive Control: An Overview and Roadmap of Future Research Opportunities". In: *IEEE Control Systems Magazine* 34.4, pp. 87–97.

Nocedal, J. and S. J. Wright (2006). *Numerical Optimization*. Second Edition. SpringerLink. Springer New York.

Oh, K.-K., M.-C. Park, and H.-S. Ahn (2015). "A survey of multi-agent formation control". In: *Automatica* 53, pp. 424–440.

Olfati-Saber, R. (2006). "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory". In: *IEEE Transactions on Automatic Control* 51.3, pp. 401–420.

Preiss, J. A., W. Honig, N. Ayanian, and G. S. Sukhatme (2017). "Downwash-aware trajectory planning for large quadrotor teams". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 250–257.

Rawlings, J. B., D. Q. Mayne, and M. Diehl (2020). *Model predictive control: Theory, computation, and design*. 2nd ed. Nob Hill Publishing.

Ren, W. and Y. Cao (2011). *Distributed Coordination of Multi-agent Networks*. Springer London.

Reynolds, C. W. (1987). "Flocks, herds and schools: A distributed behavioral model". In: *ACM SIGGRAPH Computer Graphics* 21.4, pp. 25–34.

Rochefort, Y., S. Bertrand, H. Piet-Lahanier, D. Beauvois, and D. Dumur (2012). "Cooperative Nonlinear Model Predictive Control for Flocks of Vehicles". In: *IFAC Proceedings Volumes* 45.1, pp. 169–174.

Shamma, J. S., ed. (2008). *Cooperative control of distributed multi-agent systems*. John Wiley & Sons.

Stewart, B. T., A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia (2010). "Cooperative distributed model predictive control". In: *Systems & Control Letters* 59.8, pp. 460–469.

Stomberg, G., A. Engelmann, and T. Faulwasser (2022). "A compendium of optimization algorithms for distributed linear-quadratic MPC". In: *at - Automatisierungstechnik* 70.4, pp. 317–330.

Truszkowski, W. F., M. G. Hinchey, J. L. Rash, and C. A. Rouff (2006). "Autonomous and autonomic systems: a paradigm for future space exploration missions". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 36.3, pp. 279–291.

Wang, F., G. Wang, and Y. Chen (2022). "Adaptive Spacing Policy Design of Flocking Control for Multi-agent Vehicular Systems". In: *IFAC-PapersOnLine* 55.37, pp. 524–529.

Wang, G., M. Liu, F. Wang, and Y. Chen (2023). "A Novel and Elliptical Lattice Design of Flocking Control for Multi-Agent Ground Vehicles". In: *IEEE Control Systems Letters* 7, pp. 1159–1164.

Yang, T., X. Yi, J. Wu, Y. Yuan, Di Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson (2019). "A survey of distributed optimization". In: *Annual Reviews in Control* 47, pp. 278–305.

Yuan, Q., J. Zhan, and X. Li (2017). "Outdoor flocking of quadcopter drones with decentralized model predictive control". In: *ISA transactions* 71.Pt 1, pp. 84–92.

Zhan, J. and X. Li (2011a). "Decentralized flocking protocol of multi-agent systems with predictive mechanisms". In: *Proceedings of the 30th Chinese Control Conference*, pp. 5995–6000.

Zhan, J. and X. Li (2011b). "Flocking of Discrete-time Multi-Agent Systems with Predictive Mechanisms". In: *IFAC Proceedings Volumes* 44.1, pp. 5669–5674.

Zhan, J. and X. Li (2013). "Flocking of Multi-Agent Systems Via Model Predictive Control Based on Position-Only Measurements". In: *IEEE Transactions on Industrial Informatics* 9.1, pp. 377–385.

Zhang, H.-T., M. Chen, G.-B. Stan, T. Zhou, and J. Maciejowski (2008). "Collective behavior coordination with predictive mechanisms". In: *IEEE Circuits and Systems Magazine* 8.3, pp. 67–85.

Zhang, H.-T., Z. Cheng, G. Chen, and C. Li (2015). "Model predictive flocking control for second-order multi-agent systems with input constraints". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.6, pp. 1599–1606.

Zhang, H.-T., B. Liu, Z. Cheng, and G. Chen (2016). "Model Predictive Flocking Control of the Cucker-Smale Multi-Agent Model With Input Constraints". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 63.8, pp. 1265–1275.

Zheng, Y., S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick (2017). "Distributed Model Predictive Control for Heterogeneous Vehicle Platoons Under Unidirectional Topologies". In: *IEEE Transactions on Control Systems Technology* 25.3, pp. 899–910.

Zhou, L. and S. Li (2017). "Distributed model predictive control for multi-agent flocking via neighbor screening optimization". In: *International Journal of Robust and Nonlinear Control* 27.9, pp. 1690–1705.

Appendix

A Simulation and Tuning Parameters

A.1 Parameters for Chapter 4

Parameter Study - Implementation and Prediction Mismatch

n_d	N	τ	d	r	d_o	r_o	R_u	R_p	r_δ^+	r_δ^-	ϵ_α
2	20	0.2 s	7	8.4	6	8.4	$0.05I$	$0.01I$	0.15	1	0.001

r_β^-	ϵ_β	R_γ	d_r	N_SQP	u_max	u_min
1.2	0.001	$\left(\begin{bmatrix} 0.07 & 0 \\ 0 & 0.01 \end{bmatrix} \otimes I_{n_\mathrm{d}}\right) \otimes I$	1	1	1	-1

Parameter Study - SQP Iterations

n_d	N	τ	M	d	r	d_o	r_o	R_p	r_δ^+	r_δ^-	ϵ_α
2	20	0.2 s	5	7	8.4	6	8.4	$0.01I$	0.15	1	0.001

r_β^-	ϵ_β	R_γ	d_r	u_max	u_min
1.2	0.001	$\left(\begin{bmatrix} 0.07 & 0 \\ 0 & 0.01 \end{bmatrix} \otimes I_{n_\mathrm{d}}\right) \otimes I$	1	1	-1

R_u ($N_\mathrm{SQP} = 1$)	R_u ($N_\mathrm{SQP} = 2$)	R_u ($N_\mathrm{SQP} = 5$)	R_u ($N_\mathrm{SQP} = \{10, 50\}$)
0.05	0.47	0.49	0.45

Performance Comparison

Common simulation parameters:

n_d	N	d	r	d_o
2	20	7	8.4	6

Parameters for the proposed ADMPF algorithm:

τ	M	r_o	R_u	R_p	r_δ^+	r_δ^-	ϵ_α	r_β^-	ϵ_β
0.2 s	5	8.4	$0.001I$	$0.001I$	0.12	1.7	0.001	2.2	0.001

R_γ	d_r	N_SQP	u_max	u_min
$\left(\begin{bmatrix} 0.045 & 0 \\ 0 & 0.01 \end{bmatrix} \otimes I_{n_\mathrm{d}}\right) \otimes I$	3	1	1	-1

Parameters for algorithm in Huang et al. (2019) following the notation used therein:

τ	M	r_o	λ	c	C_1	C_2
0.2 s	5	6	0.003	0.1	0.25	0.3

Parameters for algorithm in Olfati-Saber (2006) following the notation introduced in Section 2.3:

τ	r_o	s	h_α	c_1^α	c_2^α	h_β	c_1^β	c_2^β	c_1^γ	c_2^γ
0.1 s	8.4	0.1	0.2	0.68	1.2	0.9	0.4	1.44	0.36	0.64

A.2 Parameters for Chapter 5

Scenario 1

Common simulation parameters:

τ	t_f	n_d	N
2×10^{-3} s	50 s	2	20

Parameters for the proposed algorithm for the scenario with small eccentricity:

T	d	r	s	h	c_1^α	c_2^α	c_1^γ	c_2^γ
$\begin{bmatrix} 0.816 & 0 \\ 0 & 1.224 \end{bmatrix}$	5.715	6.859	0.1	0.2	5	1	0.5	0.5

Parameters for the proposed algorithm for the scenario with large eccentricity:

T	d	r	s	h	c_1^α	c_2^α	c_1^γ	c_2^γ
$\begin{bmatrix} 0.447 & 0 \\ 0 & 2.236 \end{bmatrix}$	3.13	3.757	0.1	0.2	5	1	0.5	0.5

Parameters for algorithm in Wang et al. (2022) following the notation used therein for the scenario with small eccentricity:

d_a	d_b	r	ℓ	s	τ	h	c_α	c_ϕ^α	c_1^γ	c_2^γ
7	$\frac{14}{3}$	8.4	8.4	0.1	0.3	0.8	5	1	0.5	0.5

Parameters for algorithm in Wang et al. (2022) following the notation used therein for the scenario with large eccentricity:

d_a	d_b	r	ℓ	s	τ	h	c_α	c_ϕ^α	c_1^γ	c_2^γ
7	1.4	8.4	8.4	0.1	0.3	0.8	5	1	0.5	0.5

Scenario 2

Simulation parameters:

τ	t_f	n_d	N
$2 \times 10^{-3}\,\mathrm{s}$	$100\,\mathrm{s}$	3	20

Tuning parameters:

T	d	r	s	h	c_1^α	c_2^α	c_1^γ	c_2^γ
$\begin{bmatrix} 0.684 & 0 & 0 \\ 0 & 0.855 & 0 \\ 0 & 0 & 1.71 \end{bmatrix}$	4.788	5.746	0.1	0.8	5	1	0.5	0.5

Scenario 3

Simulation parameters:

τ	t_f	n_d	N
$1 \times 10^{-2}\,\mathrm{s}$	$150\,\mathrm{s}$	2	20

Tuning parameters:

T_β	d	r	d_o	r_o	s	h_α	h_β	c_1^α	c_2^α	c_1^γ	c_2^γ	c_1^β	c_2^β
I_2	6.261	7.513	6	8.4	0.1	0.2	0.9	5	1	1	0.5	2	0

The parameters h_α and h_β are the shaping parameters for the bump function $\rho_h(z)$ for inter-agent and agent-obstacle interactions, respectively.

A.3 Parameters for Chapter 6

Common simulation parameters:

n_d	N	d	r	d_o	r_o	T	T^β
2	20	7	8.4	6	8.4	$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

Parameters for EDMPF algorithm:

τ	M	$u_{\max}$	$u_{\min}$	R_u	R_p	r_δ^+	r_δ^-	ϵ_α	r_β^-	ϵ_β
0.2 s	5	2	-2	$0.04I$	$0.01I$	0.2	2	0.001	3.5	0.001

R_γ	d_r	N_SQP
$\left(\begin{bmatrix} 0.06 & 0 \\ 0 & 0.02 \end{bmatrix} \otimes I_{n_\mathrm{d}}\right) \otimes I$	2	2

Parameters for EF algorithm:

τ	s	h_α	h_β	c_1^α	c_2^α	c_1^γ	c_2^γ	c_1^β	c_2^β
0.02	0.1	0.2	0.9	1.5	1	0.2	0.4	1	0

The parameters h_α and h_β are the shaping parameters for the bump function $\rho_h(z)$ for inter-agent and agent-obstacle interactions, respectively.