Distributed Control for Complex Mission Scenarios With Non-Holonomic Agents - An LPV Approach

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing)

genehmigte Dissertation

von

Aly Saeed Aly Aly Attallah

aus

Menoufiya, Ägypten

2021

Prüfungsvorsitzende:	Prof. Dr.-Ing. Robert Seifried
Gutachter:	Prof. Dr. Herbert Werner Prof. Dr. rer. nat. Norbert Hoffmann
Tag der mündlichen Prüfung:	22. Oktober 2020

To my dear daughter, Talia

Acknowledgments

This thesis is the culmination of my research at the Institute of Control Systems (ICS) at Hamburg University of Technology (TUHH) in the period 2017-2020. Pursuing my research was in the same time a fascinating experience and the most challenging academic task I have had to face thus far. Many people encouraged and supported me during this period and here I would like to take a moment and acknowledge their help and support.

First of all, I would like to thank my advisor Prof. Dr. Herbert Werner for the guidance, trust and support throughout the years. I am grateful for the stimulating and open discussions, for the opportunity to learn a lot by tutoring some of his exercise classes, for the valuable experiences in analytic thinking and presentation skills. Furthermore, I thank him for his kindness and nice personality.

I would like to thank Prof. Dr. rer. nat. Norbert Hoffmann for his time and effort in examining this work, and Prof. Dr.-Ing. Robert Seifried for chairing the examination board.

Staying in Hamburg was financially supported by the Egyptian Government (Ministry of Defence) to whom I wish to express my gratitude, and to the members of the Egyptian Defence Office in the Egyptian Embassy in Berlin.

Studying at ICS was a wonderful experience and I would like to thank all members of the institute, for being good colleagues and friends. Special Thanks to Ms. Bettina Schrieber and Ms. Christine Kopf, the secretary of the ICS, for their enormous help and for the good advises and the valuable administrative support not only in my first days in Hamburg but also for their endless support up to last days, and thanks to Mr. Herwig Meyer, Mr. Uwe Jahns and Mr. Klaus Baumgart, the technical staff of the ICS, for their technical support.

I am grateful to Adwait Datar, Antonio Mèndez Gonzàlez, Patrick Göttsch, Lennart Heeren, Simon Heinke, Prima Aditya and Bindu Sharan, Furug Mirali and Muhammed Shaurkhan for their help, advises and interesting discussions throughout the years.

I am also obliged to my colleagues at Military Technical College in Egypt, specially Dr. Ahmed Taimour, for his endless support and continuous encouragement, specially in hard times during COVID-19 lockdown, and to my best friend whom I have ever met, Mohamed Zaki for his nice contacts and friendly calls which eliminate the homesickness and the long distance, wishing the best for him in his PhD studies. And all thanks for *those* who support indirectly by providing their friendship.

Finally, I am forever grateful to my parents, who despite the physical distance were always there for me when I needed them. I would like to manifest my deepest appreciation to parents in law, for their support. Special thanks to my brothers, Mohamed, Abd-Elrahman, Mostafa, and brother in law, Ahmed for their encouragement. And, I want to appreciate my wife, Nesma, for her unending patience, support and motivation. Without her understanding I would not have had the strength to pursue this work, I owe her and my lovely daughter, Talia, much more than I would ever be able to express.

Hamburg, October 2020
Aly Attallah

Summary

This work is concerned with cooperative control strategies for large groups of autonomous mobile agents such as Unmanned Aerial Vehicles (UAVs), wheeled mobile robots or Autonomous Underwater Vehicles (AUVs). As observed in swarms of animals, here it is assumed that agents exchange information with their neighbors according to a given communication topology that can be represented by an undirected or directed graph. The focus is on distributed control schemes that can be implemented based on locally available information only. Analysis and synthesis conditions are developed for stability of the network and for performance in the sense of the induced ℓ_2 norm; the complexity of solving these problems is independent of the network size.

The synthesis of distributed control schemes is based on dynamic models of agents that are assumed to be subject to non-holonomic constraints. While this excludes Linear Time-Invariant (LTI) agent models, it has been shown that modeling non-holonomic agents as Linear Parameter-Varying (LPV) systems enables the design of efficient gain-scheduled controllers. This approach is extended here to the design of distributed locally gain-scheduled control schemes for networks of autonomous agents.

Two cooperative control problems are studied in particular that both involve an underlying consensus protocol: the formation control problem and the flocking control problem. Concerning the formation control problem, the use of information flow filters for networks of LPV agents is investigated, and two contributions of this thesis are: (i) an extension of the information flow filter approach that allows direct feedback of transmitted agent data into the consensus protocol, and (ii) the simultaneous design and optimization of local controllers and network filter for stability and performance in the presence of obstacles and disturbances.

As for the flocking control problem, which is based on the Reynolds rules of *cohesion, separation* and *alignment*, the focus is again on the interaction between the dynamics of the information flow in the network and the local agent dynamics. As for the formation control problem, a two-layered network architecture is developed with a flocking filter at the top that generates reference trajectories, and local tracking control of agent dynamics at the bottom. The approach is demonstrated on realistic nonlinear LPV models of AUVs.

Results proposed in this thesis are illustrated in simulated mission control scenarios such as source seeking and level curve tracking, and also experimentally on a group of autonomous micro-quadcopters.

Abstract

This thesis considers cooperative control of non-holonomic autonomous agents which are modeled as linear parameter-varying (LPV) systems. Two approaches are considered. First, the formation control approach; here a coupled information flow filter is presented that enables feedback of local agent data into the consensus protocol. Moreover, a simultaneous design and optimization of filter and local controller is proposed. Secondly, extension of flocking control to LPV agents. A separation in design is achieved by designing a flocking filter that receives data from and provides reference trajectories to local agents. This approach is illustrated with source seeking and level curve tracking scenarios.

Kurze Zusammenfassung

Diese Arbeit untersucht kooperative Regelstrategien für nicht-holonome, autonome Agenten, die als linear parametervariierende (LPV) Systeme modelliert werden. Es werden zwei verschiedene Ansätze untersucht. Zum einen die Formationsregelung; hierfür wird ein gekoppelter *Information Flow Filter* vorgestellt, der Feedback von lokalen Agenten in das Konsensprotokoll erlaubt. Weiterhin wird eine Methode für den simultanen Entwurf von Filter und lokalen Reglern entwickelt. Zum anderen wird eine *Flocking*-basierte Regelung betrachtet und für LPV-Agenten erweitert. Hier wird eine Separation im Entwurf vorgestellt, in der ein Flocking-Filter entworfen wird, der Daten von lokalen Agenten empfängt und für diese Referenztrajektorien generiert. Diese Methode wird illustriert anhand von *Source-Seeking* und *Level-Curve-Tracking* Szenarien.

Contents

Summary **i**

Abstract **iii**

1 Introduction **1**

1.1 Multi-Agent Systems 1

1.1.1 Consensus: *brief discussion* 2

1.1.2 Formation Control: *overview and classifications* 3

1.1.3 Flocking Control: *brief survey* 5

1.2 Problem Description 6

1.3 Scope and Contribution 7

1.4 Thesis Structure 9

2 Preliminaries **13**

2.1 Facts about Graph Theory 14

2.2 Linear Parameter-Varying Systems 17

2.3 State Space Representation of LPV Systems 19

2.3.1 Polytopic Representation 19

2.3.2 LFT-LPV Representation 20

2.4 LPV Controller Design and Synthesis 20

2.4.1 Continuous-time LPV System Stability and Performance 21

2.4.2 Discrete-time LPV System Stability and Performance 22

3 Agent LPV Modeling and Control **27**

3.1 Dynamic Unicycle 27

3.1.1 Dynamic Unicycle: *LPV modeling* 28

3.1.2 Dynamic Unicycle: *LPV controller design* 30

3.2 Autonomous Underwater Vehicle 34
3.2.1 AUV: *LPV modeling* 36
3.2.2 AUV: *control loop design* 38

4 Formation Control of LPV Agents 43
4.1 Information Flow in Multi-Agent Systems 43
4.2 Coupled Information Flow in Network of LPV Agents 45
4.3 Coupled-Architecture Stability Analysis 46
4.4 Application to swarm of dynamic unicycles 51

5 Flocking Control of LPV Agents 57
5.1 Flocking Algorithms in Multi-Agent Systems 57
5.2 γ-Agent Dynamics 60
5.2.1 Source Seeking 60
5.2.2 Level Curve/Surface Tracking 61
5.2.3 Level Curve/Surface Monitoring 62
5.3 Flocking of LPV Agents 64
5.3.1 Simulation Scenarios 65
5.4 Flocking under Gradient and Hessian Estimation 70
5.4.1 Simulation Scenarios 72
5.5 Flocking Loop Test Experiment 79
5.5.1 Experiment Platform 79
5.5.2 Flocking Control Experiments 80

6 Conclusions and Outlook 87
6.1 Conclusions 87
6.2 Outlook 88

A Additional Material 91

B 3D Dynamic Unicycle 97

C Nomenclature 101

Publications 115

Curriculum Vitae 117

Chapter 1

Introduction

Detailed scientific research has been conducted on groups of biological creatures in nature, such as the collaborative division of labor between ant colonies, parades of fish schools, formation of bird groups, and cooperative hunting of herds. Considerable data observation and research has shown that the overall intelligent behavior and actions can be achieved through local or regional communication and cooperation between the individuals, although the individual's ability in the group is quite limited [Anderson et al., 2001, Garnier et al., 2007]. Without the centralized control from the outside world and internal global information exchange, these groups can show the overall complex behavior, such as maintaining formation, escaping natural enemies, collaborative attacks, and finding food, only through the information exchange with the surrounding individuals. Inspired by this natural collective behavior, the Multi-Agent Systems (MAS) appear to be a promising field of research and technical applications.

1.1 Multi-Agent Systems

In the past three decades, in particular, applications of control of multiple vehicles have received increasing attention because of the advantages that can be obtained when a single complicated vehicle is equivalently replaced by many simpler ones. In this attempt, two main categories are commonly adopted for approaching control of multiple vehicles: a *centralized control approach* and a *distributed control approach*. The centralized control approach essentially assumes that a central station is available and powerful to control a group of vehicles. It can be seen as a direct extension of the well-known single vehicle control philosophy and strategy in which each vehicle in the group communicates only with the central station and its actions are based on the commands from the central station. In contrast, although it is far more complex in structure and organization, the distributed control approach does not require a central station for control. The main objective is to have the whole group of vehicles working in a cooperative fashion where information sharing among vehicles plays a central role. Based on applications and situations, both approaches are considered to be practical; however, the distributed control approach is believed to be widely recognized and appreciated because of better performance, especially with physical constraints like limited resources, limited wireless communication ranges, and applicability for a large number of agents. Some excellent surveys for recent progress on distributed multi-agent coordination and networked systems can be found in [Chen and Ren, 2019, Li and Tan, 2019, Knorn et al., 2015, Oh et al., 2015, Ren and Cao, 2010].

1.1.1 Consensus: *brief discussion*

Consensus in MAS is defined as reaching an agreement regarding a certain quantity of interest which depends on the state of all agents in a group [Olfati-Saber et al., 2007]. The consensus problem plays the main role in distributed control of MAS. It has wide applications in formation control, flocking control, and sensor network design. According to [Li and Tan, 2019], research on consensus problems passes through three stages:

1. Simulation studies.
2. Theoretical research.
3. Development and applications.

In the simulation stage, one of the early successful approaches was proposed by [Reynolds, 1987], where simulation of the collective behavior of natural groups was performed, and an agent in a group was called the *Boid* model. In [Vicsek et al., 1995], a classical model, called Vicsek model, describing the phase shift of self-driven particles was presented based on the Boid model.

The second stage involves theoretical studies. In [Jadbabaie et al., 2003], a theoretical explanation of the Vicsek model was presented based on graph theory. In [Olfati-Saber and Murray, 2004], a MAS of integrator agents was investigated and analyzed based on graph theory, and the relation between convergence rate and graph algebraic connectivity in the consensus problem of time delay was presented.

Most research on the consensus problem considers the case where the agents are governed by first order dynamics [Cao et al., 2008, Wang and Chen, 2002, Yu et al., 2009b], where agents are controlled to agree on positions. Nevertheless, the second order consensus problem has become to be recognized as an interesting topic of research [Hong et al., 2006, 2008, Ren and Beard, 2008] where each agent is governed by second order dynamics, and the second order consensus problem is then to reach an agreement among a group of agents governed by second order dynamics in the position and velocity. The consensus problem of double integrator MAS is analyzed in [Ren and Beard, 2005], and the main result was that the communication graph including the directed spanning tree is important for asymptotic stability of the group. In this sense, necessary and sufficient conditions for second order consensus is derived in [Yu et al., 2010], while incorporating nonlinear dynamics in second order consensus is analyzed in [Yu et al., 2009a]. Investigation of second order consensus in lossy network is presented in [Cui et al., 2016].

Because graph theory is an important tool in theoretical analysis of consensus, more concern is directed to investigate consensus issues, and at this stage, a lot of work has been done to enhance consensus protocols, convergence, and application prospects. In this direction, some work in weighting strategies to improve convergence in MAS of first integrator agents are presented in [Xiao and Boyd, 2004, Hao and Barooah, 2012, Mirali and Werner, 2017, Mirali et al., 2017], whereas the consensus problem in the presence of packet dropouts is studied in [Wang et al., 2010, Vaidya et al., 2012]. A weighted memory protocol is applied to the group of first order integrator MAS to achieved better convergence in presence of packet losses in [Datar et al., 2018].

1.1.2 Formation Control: *overview and classifications*

Formation control, which is one of the most active research topics within the field of cooperative control systems, aims to force multiple vehicles to achieve a predetermined geometric shape. A key problem in formation control is how to stabilize and maintain a geometrical formation shape in a distributed manner. Therefore, one seeks to design a suitable controller to ensure that a group of vehicles move in an ordered manner along a desired reference trajectory or path while avoiding collisions with obstacles or other agents. In [Balch and Arkin, 1998], three different formation forms (line, diamond, wedge) and two reference techniques (leader reference, unit center) were compared. Controlling a group of spacecrafts with three different approaches, which are explained later, are considered in [Beard et al., 2001]. In [Fax and Murray, 2004], two main approaches were considered for homogeneous LTI modeled vehicles. The stability analysis was considered for SISO systems and based on decomposition, and the Nyquist criteria. Necessary and sufficient conditions on communication graphs for stabilizing the formation of nonholonomic vehicles are investigated in [Lin et al., 2005].

A local gradient based formation control law is designed to stabilize a group of robots to a target formation in [Krick et al., 2008]. Kinematic unicycle models are used in formation control to develop rotating geometric shapes around some targets in [Marshall et al., 2006]. More discussions regarding such type of synchronization problems can be found in [Sepulchre et al., 2007]. Investigating communication constraints in a formation control problem is presented in [Sepulchre et al., 2008]. Adding or removing vehicles from the network while maintaining information graph rigidity was explored in [Eren et al., 2002]. Based on the graph theory concepts, authors in [Olfati-Saber and Murray, 2002] showed how to create a large rigid network by combination of rigid smaller networks. In [Summers et al., 2011], a combination of leader-follower strategy and co-leaders was proposed to design a decentralized control law, in which each agent executes its control using only the relative position measurements. Most of these studies are based on agents which are considered as single-integrator, double-integrator, or LTI systems, and when agents are subject to nonholonomic constraints, modeling them as LPV systems shows a realistic behavior when moving in a swarm. Some early approaches that deal with the formation control problem with LPV agents are presented in [Gonzalez et al., 2015, Cisneros et al., 2016].

Based on the previous overview, the formation control problem can be categorized based on different aspects, such as sensing capability and interaction topology [Oh et al., 2015], and the formation control problem can be classified into

- Position-based formation control: Here, vehicles are required to commonly have a global coordinate system. Sensors can measure the absolute positions with respect to the global coordinate system. The desired formation is determined by the desired absolute positions for the vehicles. In this case, interactions among the vehicles can be introduced for control performance enhancement or for addressing additional objectives like maintaining the formation shape.

- Displacement-based formation control: This requires vehicles in the group to have their own local coordinate system, and knowledge regarding the origin of the global coordinate system is not necessary. In this case, each vehicle is equipped with sensors that can measure the relative displacements or its neighbors, which implies interaction existence. The geometric formation to be achieved is determined by the relative displacement among vehicles, and each vehicle in the group controls the relative displacement with its neighbor.

- Distance-based formation control: Here, each vehicle in the group has its own local coordinate system and is equipped with sensors that can measure the relative distance with respect to its neighbors, which also implies interaction among vehicles. Orientation of the vehicle local coordinate system is not required, unlike the displacement-based approach; thus, vehicles do not need to have a common sense of orientation. The required geometric shape is achieved by controlling the relative distance among vehicles.

From the above presentation, it is concluded that higher sensing capability allows lower interaction among vehicles. Another method of classification based on the requirement of whether or not formations are time-varying is given as follows [Ren and Cao, 2010, Oh et al., 2015],

- Formation producing problems e.g. [Ren and Cao, 2010, Fax and Murray, 2004]: Here, vehicles are required to achieve a certain predetermined geometric shape, which is performed by designing distributed controllers based on the matrix theory, graph theory, Lyapunov theory,. . . etc.

- Formation tracking problems e.g. [Péni et al., 2004, Gonzalez et al., 2015]: Here, vehicles are required to track a prescribed reference trajectory forming a geometric shape. Such problems have been investigated through matrix theory, potential function based controller design, Lyapunov theory,. . . etc.

Formation control can also be classified into leader-follower, behavioral, and virtual structure approaches [Scharf et al., 2004, Beard et al., 2001]:

- Leader-follower formation: In this case, at least one vehicle is considered to be a leader which tracks a predetermined trajectory, and the other vehicles are followers which are at prescribed relative distances from the leader [Consolini et al., 2008]. The leader in this case may be a real or a virtual one.

- Behavioral formation: In this case, in addition to forming a geometric shape, many desired behaviors are required like cohesion, collision avoidance, obstacle avoidance,…etc [Barogh et al., 2015].
- Virtual structure formation: In this approach, a vehicle formation is considered to be a single object. The desired motion of this object is given. The desired motions of vehicles are determined from that of the virtual object [Low, 2014].

Depending on whether or not the desired formation shapes are explicitly prescribed, one may also classify formation control problems as follows [Oh et al., 2015],

- Morphous formation: The desired formation shapes are explicitly determined by assigning vehicle positions, relative distances, and/or positions among vehicles.
- Amorphous formation: Without any predetermined formations, the geometric shapes are formed based on the behaviors among vehicles in the group, such as cohesion and collision avoidance. Amorphous formation is related to behavioral formation discussed above.

1.1.3 Flocking Control: *brief survey*

In the previous discussion, formation control is based on a first order consensus protocol, which is mainly based on (relative) positions. One step further is flocking control, which is considered as a nonlinear type of the second order consensus protocol which depends on not only positions but also velocities of vehicles in a group. Flocking is a collective behavior of a group of interacting, self-organized vehicles with a common group objective. One of first models that attempted to investigate this behavior was proposed in [Reynolds, 1987], in which the three main flocking rules are presented, known as Reynolds rules:

- Cohesion: stay close to nearby neighbors;
- Separation: avoid collisions with nearby neighbors;
- Alignment: match velocity with nearby neighbors.

Following Reynolds' model, many attempts have been made to describe the collective behavior mathematically [Vicsek et al., 1995, Shimoyama et al., 1996, Toner and Tu, 1998, Mogilner and Edelstein-Keshet, 1999]. In addition to the proposed models, other flocking algorithms have been proposed. Most of them are based on artificial potential fields to represent the first two rules of cohesion and separation, and the velocity alignment rule is simply represented as a velocity consensus protocol. In [Leonard and Fiorelli, 2001], a framework for collective motion control of vehicles was proposed modeled as a double-integrator system based on artificial potential functions and virtual leader(s). The artificial potential functions present the interaction among vehicles to maintain a certain offset. The task of the virtual leader(s) is to provide reference points that affect the motion of the vehicles in the group via additional potential functions.

In [Tanner et al., 2003a,b], the authors assumed that the relative position and velocity interaction topologies may be nonidentical. They presented a gradient-based control low combined with a velocity consensus term, which represent cohesion/ separation and alignment, respectively, and analyzed the stability under fixed topologies and switching interaction topology.

In [Olfati-Saber, 2006], different flocking algorithms are proposed based on the Reynolds rules. Underlying vehicles are considered as double integrators. In addition to the usage of the nonlinear potential function and velocity consensus protocol, the obstacle avoidance scheme is proposed. Through the proposed algorithms, the stability of the whole network is guaranteed and the error of relative distances between vehicles converges to zero asymptotically. In addition, the vehicle velocities converge to a common vector, and no collisions occur.

Following this work, different issues were studied, such as preserving network connectivity [Zavlanos et al., 2007, Jia and Wang, 2014], and several protocols have been developed like multi-target flocking [Luo et al., 2010], in which the whole group is divided into sub-groups to achieve different missions. The theoretical framework for flocking with time-varying velocity virtual leaders is presented in [Su et al., 2009b, Wen et al., 2012]. In [Zhao et al., 2017], a leader-follower flocking protocol is studied where the leader's dynamics and states are not available to the whole flock. A hybrid event-triggered flocking control scheme is presented in [Yu et al., 2016], where position information is transmitted continuously and velocities are used at discrete time instants, based on an event-trigger mechanism.

Different applications applying flocking rules include source seeking and level curve tracking using glowworm optimization techniques [Krishnanand and Ghose, 2009, Turgeman and Werner, 2018] . In addition, most existing flocking algorithms depend on information regarding both the relative position and the relative velocity. However, in practice, limited equipment and sensors are considered; thus, some work have formulated flocking based on the position, such as [Su et al., 2009a], to have the advantage of decreasing equipment cost and network traffic.

1.2 Problem Description

We consider cooperative control can be divided into two layers, the *abstraction layer* that represents the network dynamics and the *physical layer* which represents agents' dynamics. So, the MAS is dealt in this thesis mainly from point of view of interaction between these two layers. In this sense, this interaction is studied in two main approaches, *formation control* of LPV-agents and *flocking control* of LPV-agents. In formation control approach, a group of LPV-modeled agents can exchange information through Information Flow Filter (IFF), which is first presented in [Fax and Murray, 2004] and the task is to extend the IFF application to a network of LPV agents and achieve a certain geometric shape with better cooperative behavior among agents by simultaneous and coupled design of both agent LPV controller and IFF.

As a degree of flexibility, flocking rules are applied to a swarm of LPV-agents in complex missions like source seeking scenario and level curve/ surface monitoring. Extension of the information flow idea from formation control, (first order consensus), to flocking control, (second order consensus), is considered in a coupled interaction between abstraction and physical layers. Task of all agents is to find equilibrium around source/ multiple sources or to move along a certain concentration level. In each problem, agents can exchange information with their neighbours and the *coupled* interaction between the two layers improves the cooperative behavior of the swarm to achieve a mission. In all scenarios, it is assumed that agents are equipped with suitable sensors and has the computation capability to achieve the required task. So, we consider the following problems:

- **LPV modeling and control of autonomous underwater vehicle**: Consider a complex, twelve state nonlinear autonomous underwater vehicle actuated by four propellers, find the LPV representation of such dynamics and design an LPV output feedback controller.
- **Interaction of information and physical layers in formation control problem**: Consider a group of N identical agents, each of which is modeled as a discrete-time LPV system, exchanging information through a communication topology, find a distributed LPV formation controller with fully populated model matrices, i.e. there is coupling through the off-diagonal blocks of the controller between network dynamics and local agent dynamics.
- **Interaction of information and physical layers in flocking control problem**: Consider a group of N identical agents, each of which is modeled as a continuous-time LPV system, design a flocking filter and local LPV controller that allow the group to reach a required mission of:
  - **Source seeking Problem**: where agents are required to reach equilibrium at the source of certain pollution.
  - **Level Surface/ Curve Monitoring Problem**: where agents are required not to reach a source but to move along certain concentration level of such pollution.

1.3 Scope and Contribution

In order to solve the aforementioned problems, two main approaches are considered. In the first approach, a *coupled information flow architecture* is proposed to improve the co-operative behaviour among agents in a network where a distributed controller can receive both formation error and tracking error, and provides the reference to be tracked and the required tracking control effort, and extends the application of IFF from networks of LTI systems to nonlinear agents in LPV form. Secondly, a *coupled flocking loop architecture* is proposed in which a flocking filter receives realistic data and generates virtual references to local agents to achieve a required mission.

In both cases, representing agents as LPV models is considered and two LPV models are developed, one is an LPV representation of a dynamic unicycle and the other is modeling a realistic underwater vehicle as an LPV model. Thesis contributions can be summarized in the following points:

1. **Cascaded control structure of AUVs:** modeling the underwater vehicle as an LPV agent leads to an efficient gain scheduled controller. In this sense, the AUV under simulation studies in this work is modeled as LPV agent using Taylor series expansion, which results in decoupling the agent dynamics into inner dynamics that represents the orientations and its rates, and outer dynamics that represents the position and linear velocity [Attallah et al., 2020]. So, it enables to design inner and outer LPV output feedback controllers and this is presented in Chapter 3.

2. **Distributed coupled information flow scheme:** to solve the interaction between information and physical layers in formation control problem, the contribution is twofold. First, extend the application of information flow filter from LTI agents to LPV modeled ones. Second, modification is applied such that a coupled architecture is proposed in which the distributed coupled controller receives the formation error and tracking error and provides the local agent the corresponding local control effort and virtual reference to be tracked [Attallah and Werner, 2020]. These results are mainly presented in Corollary 4.1 and Theorem 4.2 in Chapter 4.

3. **Distributed source seeking algorithms:** to solve the source seeking problem, flocking control is investigated. The contribution in this direction is to propose a coupled flocking control loop architecture based on extension of IFF from formation control to flocking control application, in which the the whole architecture consists of main levels, the virtual network, *flocking filter*, that receives real data and provides references to be tracked to the physical level that represents the agents. In this case, agent dynamics is represented in LPV form and network dynamics represent the flocking behaviour where flocking rules are applied. In addition, the source seeking problem can be solved by navigational feedback control that depends on scalar field gradient and Hessian that are assumed available at agents' location [Attallah et al., 2020]. These results are presented in Proposition 5.1 and Theorem 5.1 in Chapter 5.

4. **Distributed level surface/curve monitoring:** under the above assumption for scalar field gradient and Hessian availability, and based on the coupled flocking control architecture, a navigational feedback control law is proposed to solve a level surface/ curve monitoring problem. The flocking filter is responsible for processing the data and providing virtual references to LPV local agents to achieve this mission. These results are presented in Proposition 5.3 based on coupled flocking loop stability in Theorem 5.1 in Chapter 5.

5. **Mission achievement under scalar field gradient and Hessian estimation:** estimation scheme for scalar field gradient and hessian calculation is proposed to get rid of the assumption of scalar filed gradient and Hessian availability at agents' location. The estimation process is based on least squares estimation. According to the required mission in coupled flocking architecture, Algorithm 2 and Algorithm

3 in Chapter 5 provide a suitable navigational feedback control law based on the estimated gradient and Hessian.

1.4 Thesis Structure

This thesis is structured as follows. Chapter 2 introduces some important facts and definitions regarding graph theory, which is key to the cooperative control field of research, and some basic aspects related to linear-parameter varying control, in particular, the design and synthesis of the output feedback controller.

In Chapter 3, LPV modeling and control of the dynamic unicycle, which is considered to be a generic model for nonholonomic agents like ground vehicles, space-rockets, fixed-wing unmanned aerial vehicles, and unmanned underwater vehicles. In addition, LPV modeling and control of autonomous underwater vehicles is used for three dimensional realistic and complex missions.

The information flow in formation control of the LPV-modeled agent problem is investigated in Chapter 4. Analysis results and sufficient conditions for (robust) stability with guaranteed performance of the network of LPV nonholonomic agents are detailed.

In Chapter 5, for more realistic and complex missions like source seeking and level surface monitoring, flocking control is used as a realistic flexible approach to control the swarm of realistic underwater agents modeled as LPV agents. The coupled architecture is shown to achieve a given mission. An estimation scheme is proposed to calculate the gradient and Hessian for a scalar field.

Finally, Chapter 6 draws the conclusions and outlines directions for future developments. The chapters of the thesis are organized in such a way that each of them builds upon its predecessors. Hence, for a thorough understanding of the matter, it is recommended that one reads them in the presented order. Nevertheless, readers interested in different aspects of this work might find it useful to concentrate only on particular chapters. Figure 1.1 shows two possible paths through this thesis. For readers interested in formation control of the swarm of LPV-modeled agents, a blue track is recommend. In contrast, for flocking control of such swarms, readers should refer to the red track.

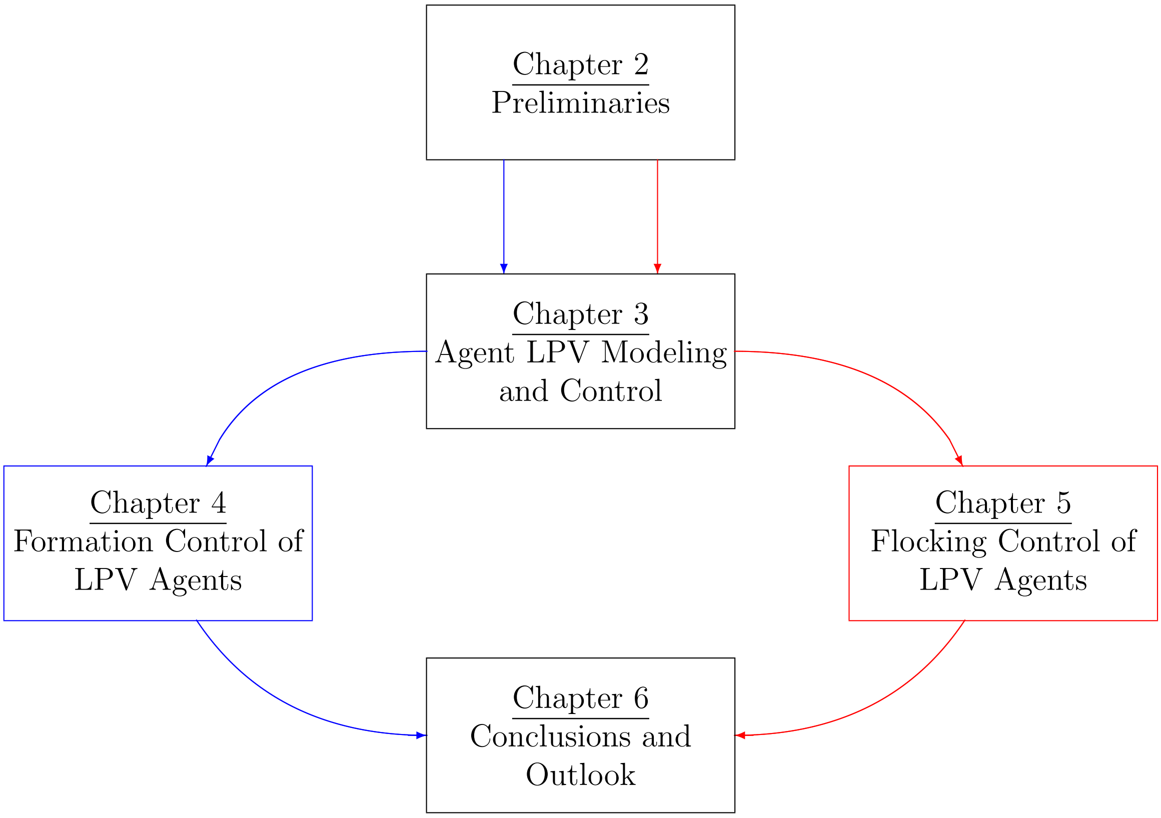
Chapter 2
Preliminaries
Chapter 3
Agent LPV Modeling and Control
Chapter 4
Formation Control of LPV Agents
Chapter 5
Flocking Control of LPV Agents
Chapter 6
Conclusions and Outlook

Figure 1.1: Reading tracks. (−) Formation Control Track, (−) Flocking Control Track

Chapter 2

Preliminaries

Notation

It is convenient to introduce some of the notations in advance. The fields of real numbers, positive integers, and complex numbers are denoted by $\mathbb{R}, \mathbb{N}$ and $\mathbb{C}$ respectively. Complex units are denoted as j, while scalar and vector variables are denoted by lower-case letters; upper-case letters are used for matrices. Furthermore, I_n denotes an $n \times n$ identity matrix; 0 will denote a zero matrix with the appropriate dimensions. The sets of $n \times m$ real and complex matrices are denoted by $\mathbb{R}^{n\times m}$ and $\mathbb{C}^{n\times m}$.

For a matrix X, its transpose is denoted by X^T, its pseudo-inverse is denoted by $X^\dagger$. The element in row i and column j of a matrix X is denoted by x_{ij}, and element i of a vector q will be denoted by q_i. Furthermore, a set of $n \times n$ symmetric matrices is denoted by $\mathcal{S}^{n\times n}$, and $S \succ 0$ ($S \succeq 0$) will denote that S is a symmetric positive definite (positive semi-definite) matrix.

The Kronecker product will be denoted by $\otimes$. For a matrix Y, the notation $\hat{Y}$ will be used to denote $\hat{Y} = I_N \otimes Y$ where N will be clear from the context, whereas $Y_{(m)} = Y \otimes I_m$. To represent $blkdiag\big(Z(\rho_1), \ldots, Z(\rho_N)\big)$, a shorthand notion of $\bar{Z}(\rho)$ is used. The $\mathcal{L}_2$-norm of a continuous-time signal $x(t)$ is denoted by $\|x(t)\|_{\mathcal{L}_2}$ and is defined as

$$\|x(t)\|_{\mathcal{L}_2} = \Big(\int_0^\infty x(t)^2 dt \Big)^{\frac{1}{2}}$$

and the ℓ_2-norm of the discrete-time signal $x(k) = [x_1(k), \ldots, x_n(k)]^T \in \mathbb{C}^n$ is denoted by $\|x(k)\|_{\ell_2}$ and defined as

$$\|x(k)\|_{\ell_2} = \Big(\sum_{i=1}^{n} |x_i(k)|^2 \Big)^{\frac{1}{2}}$$

The induced $\mathcal{L}_2$-norm of the LPV system $T_{zr}(\rho)$ with continuous time input signal $r(t)$, output signal $z(t)$, and scheduling parameter vector $\rho \in \mathcal{P} \subset \mathbb{R}^{n_\rho}$ is defined as

$$\|T_{zr}(\rho)\|_{\mathcal{L}_2} = \sup_{\rho\in\mathcal{P}} \sup_{\|r\|_{\mathcal{L}_2}\neq 0} \frac{\|z\|_{\mathcal{L}_2}}{\|r\|_{\mathcal{L}_2}},$$

For the discrete-time LPV system, the ℓ_2-norm is used with discrete input signal $r(k)$, output signal $z(k)$, and scheduling parameter vector $\theta \in \mathcal{P}_\theta \subset \mathbb{R}^{n_\theta}$ and is defined as

$$\|T_{zr}(\theta)\|_{\ell_2} = \sup_{\theta \in \mathcal{P}_\theta} \sup_{\|r\|_{\ell_2} \neq 0} \frac{\|z\|_{\ell_2}}{\|r\|_{\ell_2}}$$

2.1 Facts about Graph Theory

In an MAS, information exchange plays a vital role in achieving the group mission. This can be done by either through wireless communication where the agents can broadcast their information or through sensing capabilities where the agent can calculate data of the neighbors based on the sensed information from e.g. radar, ultrasonic sensors, or cameras. Based on how information is exchanged among agents, communication- or sensing-topology terms are used [Popov, 2012].

Graph theory has become an important tool in the field of cooperative control systems, because the information exchange among agents can be modeled as a graph. Some important concepts and results related to graph theory that are important in this thesis are provided; otherwise, the reader can refer to [Mesbahi and Egerstedt, 2010].

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as a combination of a set of nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$, $N \in \mathbb{N}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. In the interaction graph of a MAS, the nodes represent the agents and the edges represent the communication links. This means, for each existing edge (v_i, v_j), the head node v_i is receiving data from node v_j. For each node v_i , the neighbor set is defined as

$$\mathcal{N}_i = \{v_j | (v_i, v_j) \in \mathcal{E},\ i \neq j\} \subset \mathcal{V}$$

The cardinality $|\mathcal{N}_i|$ denotes the number of elements in $\mathcal{N}_i$. A graph $\mathcal{G}$ is called an undirected graph if and only if $(v_i, v_j) \in \mathcal{E} \iff (v_j, v_i) \in \mathcal{E}$ i.e. the information exchange is always bi-directional. Otherwise, it is called a directed graph, Figure 2.1.

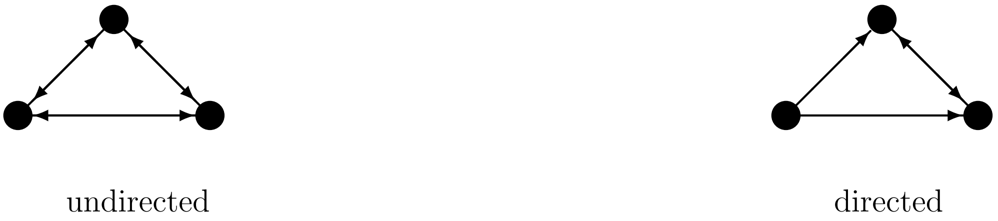

Figure 2.1: Undirected and directed graphs examples

In a directed graph $\mathcal{G}$, a sequence of edges $(v_i, v_{h_1}), (v_{h_1}, v_{h_2}), \ldots, (v_{h_k}, v_j)$, which starts from vertex v_j and ends at vertex v_i, is called a directed path from v_j to v_i. A vertex v_j

is called globally reachable if there exists a directed path from v_j to every other vertex. A graph is said to have a spanning tree if it has at least one globally reachable vertex. If every vertex is globally reachable, then the graph is strongly connected (or simply connected for undirected graphs).

The adjacency matrix $\mathcal{A}^0$= $[a^0_{ij}] \in \mathbb{R}^{N\times N}$ of the graph is defined as

$$a^0_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$$

and a degree matrix $\mathcal{D}^0$=$diag(d_1, \dots, d_N)$ where $d_i = \sum_{j=1}^{N} a^0_{ij},\ i = 1, \dots, N$. Based on that, the Laplacian matrix is defined as $\mathcal{L}^0$= $\mathcal{D}^0 - \mathcal{A}^0$. The second smallest eigenvalue λ_2 of the Laplacian is known as the algebraic connectivity of a graph [Fiedler, 1973]. If the graph ($\mathcal{G}$) is connected and undirected, then the eigenvalues of its Laplacian $\mathcal{L}^0$ satisfy $0 = \lambda_1 \leq \lambda_2 \leq \dots \lambda_N$ and the algebraic connectivity,

$$\lambda_2 = \min_{z \perp \mathbf{1}} \frac{z^T \mathcal{L} z}{\|z\|^2}$$

In addition, if a new graph $\mathcal{G}_{new}$ is produced by adding undirected edges to an undirected graph ($\mathcal{G}$) with the same vertices, then $\lambda_2(\mathcal{L}^0(\mathcal{G})) \leq \lambda_2(\mathcal{L}^0(\mathcal{G}_{new}))$. The normalized Laplacian matrix $\mathcal{L}$= $[l_{ij}]$ of a graph is defined as follows:

$$l_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } |\mathcal{N}_i| \neq 0 \\ -\frac{1}{|\mathcal{N}_i|} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

It is known that based on its construction, each row sum of $\mathcal{L}$ is equal to 0. Accordingly, the vector $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^N$ is the right eigenvector of $\mathcal{L}$ associated with the eigenvalue $\lambda_1 = 0$. Using Gershgorin's circle theorem, it can be shown that all eigenvalues of $\mathcal{L}$ are contained in the closed Perron disk $\mathbf{P}$ defined as

$$\mathbf{P} = \{z \in \mathbb{C} |\ |z - 1| \leq 1\}$$

and the remaining eigenvalues $\lambda_2, \cdots, \lambda_N$ are non-zero, if the graph is strongly connected. From $\mathcal{L} = I - \mathcal{A}$, it can be inferred that the eigenvalues of the adjacency matrix are located in the closed unit disk, $\mathbf{U}$= $\{z \in \mathbb{C} |\ |z| \leq 1\}$. In addition, the zero eigenvalue of $\mathcal{L}$ corresponds to eigenvalue 1 of $\mathcal{A}$, such that $\mathcal{A}$ has the right eigenvector $\mathbf{1}$ associated with the eigenvalue 1 [Popov, 2012, Bartels, 2019].

Example 2.1. *Consider an MAS with 6 agents and the topology shown below.*

This topology is described by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ which is directed, has no leader, and has a set of vertices

$$\mathcal{V} = \{1, 2, 3, 4, 5, 6\}$$

and edge set

$$\mathcal{E} = \{(1,5), (2,5), (2,6), (3,1), (3,4), (4,3), (4,6), (5,1), (5,3), (6,2), (6,3)\}$$

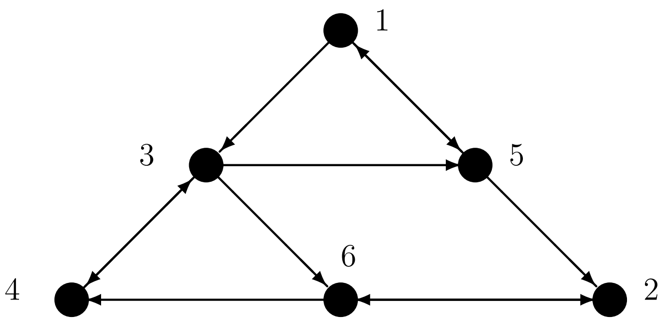

Figure 2.2: Graph example

For $i = 5$*, the following hold:* $\mathcal{N}_5 = \{1, 3\}$*,* $|\mathcal{N}_5| = 2$*. The normalized Laplacian matrix of the graph is given as follows:*

$$\mathcal{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & \frac{-1}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & 0 & -1 & 1 & 0 & 0 \\ \frac{-1}{2} & \frac{-1}{2} & 0 & 0 & 1 & 0 \\ 0 & \frac{-1}{2} & 0 & \frac{-1}{2} & 0 & 1 \end{bmatrix}$$

with corresponding eigenvalues,

$$\lambda_i \in \{0, 1.4358, 1.7782 \pm \mathrm{j}0.2844, 0.5039 \pm \mathrm{j}0.1801\}$$

The eigenvalues (marked in blue) together with the Perron disk are shown in Figure 2.3.

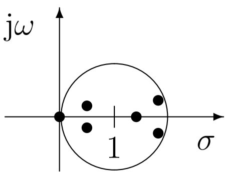

Figure 2.3: Normalized Laplacian eigenvalues

2.2 Linear Parameter-Varying Systems

Commonly in the literature [Slotine et al., 1991, Khalil, 2002], a class of nonlinear time-invarient systems is described in the following form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t)) \end{aligned} \tag{2.2}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$. The functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ are continuously differentiable functions and satisfy a Lipschitz condition. If it is possible to represent the nonlinear system (2.2) in the form of

$$\begin{aligned} \dot{x}(t) &= A(\rho(t))x(t) + B(\rho(t))u(t) \\ y(t) &= C(\rho(t))x(t) + D(\rho(t))u(t) \end{aligned} \tag{2.3}$$

where ρ is a time-varying vector of scheduling variables that may depend on $x(t)$, and $u(t)$, and $A(\cdot)$, $B(\cdot)$, $C(\cdot)$ and $D\ (\cdot)$ are continuous functions of ρ, then system (2.3) is called a *quasi*-linear-parameter-varying system [Shamma and Athans, 1990, Hoffmann and Werner, 2014]. This class can be used to represent general nonlinear systems in the LPV form and makes it possible to introduce modern linear synthesis techniques into a nonlinear design.

The scheduling vector ρ is usually assumed to be unknown *a priori* but to be measurable online, and it takes values in a known compact set $\mathcal{P} \subset \mathbb{R}^{n_\rho}$. To model the nonlinear system to be a *quasi-LPV* system, the scheduling variables are allowed to depend on the system state and input variables. Consequently, a set $\mathcal{P}$ cannot be assumed to be known *a priori* but has to be constructed when analyzing the stability and performance. We define a set

$$\mathcal{F}_{\mathcal{P}} = \{\rho(t) \in C^1(\mathbb{R}^+, \mathbb{R}^{n_\rho}) |\ \rho(t) \in \mathcal{P}\ \forall t \geq 0\} \tag{2.4}$$

where $\mathcal{F}_{\mathcal{P}}$ is a subset of a signal space of admissible scheduling trajectories. If the LPV system does not only depend on the values of the scheduling vector $\rho(t)$ but also on the rate of change $\dot{\rho}(t)$ such that

$$|\dot{\rho}_i| \leq \bar{\chi}_i \quad \forall t \geq 0,\ i = 1, \ldots, n_\rho$$

and $\chi = [\chi_1\ \ldots\ \chi_{n_\rho}]^T$, and we define the set,

$$\mathcal{X} = \{\chi \in \mathbb{R}^{n_\rho} |\ |\chi_i| \leq \bar{\chi}_i,\ i = 1, \ldots, n_\rho\} \tag{2.5}$$

then, the set of scheduling trajectories (2.4) can be re-defined as

$$\mathcal{F}_{\mathcal{P}}^{\mathcal{X}} = \{\rho(t) \in C^1(\mathbb{R}^+, \mathbb{R}^{n_\rho}) |\ \rho(t) \in \mathcal{P},\ \dot{\rho}(t) \in \mathcal{X}\ \ \forall t \geq 0\} \tag{2.6}$$

which satisfies the constraints of scheduling parameters and their rates of change. An LPV model is then constructed from a given nonlinear state space model using different techniques. The most common employed techniques to transform a nonlinear system into the LPV form are discussed in [Kwiatkowski, 2008, Tóth, 2010, Abbas, 2010, Lovera et al., 2013].

Local Modeling Method

The Jacobian linearization approach has been widely employed in the literature, e.g. [Reberga et al., 2005], to create a family of LTI models, using the first-order Jacobian linearization of (2.2) with respect to a set of equilibrium points, and the interpolation between them provides the LPV model, which is linear in the inputs and states.

$$\begin{aligned} \delta\dot{x} &= A(\rho)\delta x + B(\rho)\delta u \\ \delta y &= C(\rho)\delta x + D(\rho)\delta u \end{aligned} \tag{2.7}$$

where $\delta x = x - x_0,\ \delta u = u - u_0,$ and $\delta y = y - y_0$ and $x_0,\ u_0,\ y_0$ are the steady-state equilibrium points. The resulting model is a local approximation of the dynamics of the nonlinear plant around this set of equilibrium points. This implies that the LPV trajectories are limited to a subset of actual trajectories of the nonlinear system exactly at equilibrium.

Global Modeling Method: velocity-based linearization

A generalized approach proposed by [Leith and Leithead, 1998c,b,a] is the velocity-based linearization which provides a general method for transforming systems into the LPV/ *quasi*-LPV form. The velocity-based algorithms are developed for wide applications to control unmanned vehicles [Leith and Leithead, 2001], turbofan engines [DeCastro, 2007], and robotic arms using nonlinear model predictive control [Cisneros and Werner, 2021]. In contrast to the Jacobian linearization method, the velocity-based analysis and design framework associates a linear system with every operating point of a nonlinear system, not just the equilibrium operating points [Kaminer et al., 1995, Halás et al., 2003]. In this approach, the LPV model can be obtained by time differentiation of (2.7), and the perturbation quantities are no longer required to remain in the vicinity of the interpolation point; thus, the LPV states are identical to the actual system state derivatives:

$$\begin{aligned} \delta\ddot{x} &= A(\rho)\dot{x} + B(\rho)\dot{u} \\ \dot{y} &= C(\rho)\dot{x} + D(\rho)\dot{u} \end{aligned} \tag{2.8}$$

with a new state vector $\xi = [\dot{x}^T\ y^T]^T$. The main advantage of this method is that the equilibrium state terms present in (2.7) disappear because of differentiation. This method has some shortcomings, such as a higher dimension state vector, $n_x + n_y$, which may result in computational troubles if the number of system outputs is large. In addition, if the model has an additional disturbance input, the structure of the controller will depend on its rate, as the measured disturbance inputs to the LPV models are brought in as a derivative. In addition, if the inputs to this derivative are liable to noise or biasing, this will lead to uncertainties in amplifications with damaging effects [DeCastro, 2007].

Another method is to directly convert the continuous-time nonlinear model in (2.2) into a *quasi*-LPV in the form of (2.3): The main principle of this method is that virtual scheduling parameters are assigned to each nonlinear function element of the resulting matrix functions $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $D(\cdot)$ in (2.3). In spite of its ad-hoc nature, this method results in a *quasi*-LPV model which exactly represents the corresponding nonlinear one, and its efficiency has been demonstrated in many applications, e.g. [Abbas et al., 2009]. The main problems of this approach are non-uniqueness and overbounding, and the number of scheduling parameters can increase rapidly with system order.

2.3 State Space Representation of LPV Systems

An LPV system can be represented in different forms: *gridding* [Wu, 1995] where the operating envelop is divided into a grid of operating points and both plant model and controller are obtained by suitable interpolation within the grid, *polytopic representation* [Apkarian et al., 1995] where the models depend on the parameters affinely, or *LFT form* [Apkarian and Gahinet, 1995] where uncertainties and model parameters are considered, and the results are nominal generalized plants and uncertain blocks.

2.3.1 Polytopic Representation

The LPV system (2.3) is called a polytopic LPV model if it satisfies two conditions:

1. The set $\mathcal{P}$ is a polytope, i.e. it can be expressed as a convex hull

$$\mathcal{P} = \mathrm{Co}\left\{\rho_1, \rho_2, \ldots, \rho_n\right\} \tag{2.9}$$

   where the $\rho_i \in \mathbb{R}^{n_\rho}$ are the vertices of the polytope and n is the number of vertices. The representation (2.9) implies that

$$\mathcal{P} = \left\{ \rho \in \mathbb{R}^{n_\rho} \mid \rho = \sum_{i=1}^{n} \alpha_i \rho_i, \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0, i = 1, \ldots, n \right\}$$

   The coefficients α_i are referred to as convex coordinates. A typical occurrence of a polytopic set of parameters is a hyperrectangle, which arises when the individual parameters ρ_i are known to be confined to intervals $\underline{\rho}_i \leq \rho_i \leq \bar{\rho}_i, i = 1, \ldots, n$.

2. The model (2.3) depends affinely on the parameter vector ρ. In this case, the set of admissible LTI systems that is generated when ρ in (2.3) ranges over the polytope $\mathcal{P}$ is itself a polytope, i.e. it can be represented by

$$\begin{bmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \sum_{i=1}^{n} \rho_i \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$$

2.3.2 LFT-LPV Representation

The LPV system (2.3) can be represented as an LFT-LPV system, see Figure 2.4, if it can be represented in the form

$$\begin{aligned} \dot{x} &= Ax + B_\Delta q_\Delta + Bu \\ p_\Delta &= C_\Delta x + D_{\Delta\Delta} q_\Delta + D_{\Delta u} u \\ y &= Cx + D_{u\Delta} q_\Delta + Du \\ q_\Delta &= \Delta(\rho(t)) p_\Delta, \quad \rho(t) \in \mathcal{F}_\mathcal{P} \end{aligned} \tag{2.10}$$

where $q_\Delta \in \mathbb{R}^{n_{q_\Delta}}$, $p_\Delta \in \mathbb{R}^{n_{p_\Delta}}$ denote the parameter channel of the system. The parameter $\rho(t)$ is restricted to a set of admissible trajectories $\mathcal{F}_\mathcal{P}$, and the parameter block $\Delta(\rho(t)) \in \mathcal{C}^0\left(\mathbb{R}^{n_\delta}, \mathbb{R}^{n_{q_\Delta} \times n_{p_\Delta}}\right)$ is a continuous matrix-valued function of the LFT parameter vector $\rho(t)$.

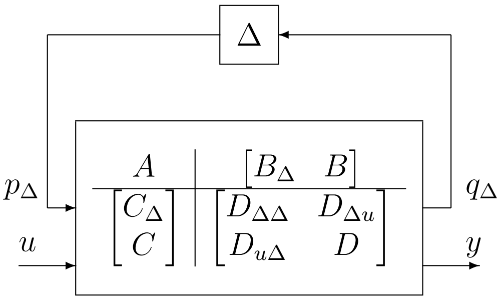
Δ
p_Δ
u
q_Δ
y
A
$\begin{bmatrix} B_\Delta & B \end{bmatrix}$
$\begin{bmatrix} C_\Delta \\ C \end{bmatrix}$
$\begin{bmatrix} D_{\Delta\Delta} & D_{\Delta u} \\ D_{u\Delta} & D \end{bmatrix}$

Figure 2.4: LFT-LPV system representation

2.4 LPV Controller Design and Synthesis

Several approaches have been proposed previously to design gain scheduled controllers for LPV systems. Most of these approaches are state-space-based and are extensions of $\mathcal{H}_\infty$ optimal control of LTI systems to LPV systems based on linear matrix inequality optimization for the design and synthesis of the gain scheduled controller [Wu, 1995, Gahinet and Apkarian, 1994]. The obtained performance is usually in terms of the induced $\mathcal{L}_2$-norm for continuous time systems [Wu et al., 1996] or ℓ_2-norm for discrete-time systems [De Caigny et al., 2012]. The two main categories of LPV controller design and synthesis are as follows:

1. Parameter independent Lyapunov function method, which is conservative but numerically cheap, which makes it attractive for practical implementation.
2. Parameter dependent Lyapunov function method, which offers a less conservative synthesis tools; however, it suffers from numerical complexity.

In this thesis, the output feedback LPV controller design and synthesis are considered for continuous-time systems and discrete-time ones.

2.4.1 Continuous-time LPV System Stability and Performance

The LPV system (2.3) can be augmented with a performance channel to yield a generalized plant $G(\rho)$ defined as

$$\begin{bmatrix} \dot{x} \\ z \\ e \end{bmatrix} = \begin{bmatrix} A(\rho) & B_1(\rho) & B_2(\rho) \\ C_1(\rho) & 0 & D_{12}(\rho) \\ C_2(\rho) & D_{21}(\rho) & 0 \end{bmatrix} \begin{bmatrix} x \\ r \\ u \end{bmatrix} \tag{2.11}$$

where we assume $C_1(\rho) = \begin{bmatrix} C_{11}^T(\rho) & C_{12}^T(\rho) \end{bmatrix}^T$, $B_1(\rho) = \begin{bmatrix} B_{11}(\rho) & B_{12}(\rho) \end{bmatrix}$, $D_{12} = \begin{bmatrix} 0 & I \end{bmatrix}$, $D_{21} = \begin{bmatrix} 0 & I \end{bmatrix}^T$, $x \in \mathbb{R}^{n_x}$, $r \in \mathbb{R}^{n_r}$, $u \in \mathbb{R}^{n_u}$, $z \in \mathbb{R}^{n_z}$, and $e \in \mathbb{R}^{n_e}$. This structure of a system plant $G(\rho)$ assumes that $D_{11} = 0$ and both D_{12} and D_{21} are full column rank $\forall \rho \in \mathcal{P}$. For case of $D_{11} \neq 0$, a loop transformation can be considered [Wu, 1995].

Consider an LPV controller $K(\rho)$ of the form

$$\begin{bmatrix} \dot{x}_C \\ u \end{bmatrix} = \begin{bmatrix} A_C(\rho) & B_C(\rho) \\ C_C(\rho) & D_C(\rho) \end{bmatrix} \begin{bmatrix} x_C \\ e \end{bmatrix} \tag{2.12}$$

where $x_C \in \mathbb{R}^{n_{x_C}}$. The controller depends linearly on the measurement e and has arbitrary dependence on the (measurable) parameter vector ρ. The closed-loop interconnection of $G(\rho)$ and $K(\rho)$ is given by a Linear Fraction Transformation (LFT) and is denoted by $\mathcal{F}_l(G(\rho), C(\rho))$, see Figure 2.5. A state space model is

$$\begin{bmatrix} \dot{x}_{cl} \\ z \end{bmatrix} = \begin{bmatrix} A_{cl}(\rho) & B_{cl}(\rho) \\ C_{cl}(\rho) & D_{cl}(\rho) \end{bmatrix} \begin{bmatrix} x_{cl} \\ r \end{bmatrix}, \tag{2.13}$$

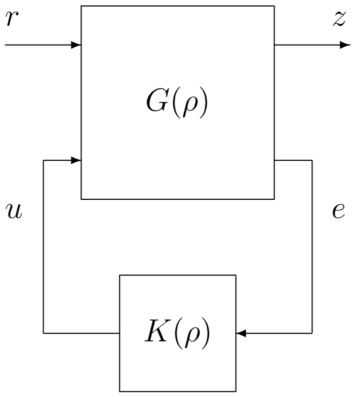

Figure 2.5: Feedback interconnection of generalized plant and controller

where $x_{cl} \in \mathbb{R}^{n_{x_{cl}}}$ where $n_{x_{cl}} = n_x + n_{x_C}$. The objective is to synthesize a controller $K(\rho)$ which minimizes the closed-loop induced $\mathcal{L}_2$ gain from the performance input r to performance output z, i.e. that solves

$$\min_{K(\rho)} \left\| \mathcal{F}_l \left(G(\rho), K(\rho) \right) \right\|_{\mathcal{L}_2} \tag{2.14}$$

The following theorem provides a sufficient condition that forms the basis for output feedback LPV controller design based on the dependent Lyapunov function method [Wu, 1995].

Theorem 2.1. *The parameter-dependent output feedback problem is solvable if and only if there exists* $P(\rho),\ Q(\rho) \in C^1(\mathbb{R}^{n_\rho}, \mathcal{S}^{n_{x_{cl}} \times n_{x_{cl}}})$ *such that* $\rho \in \mathcal{P}$ *and the following conditions are satisfied:*

$$P(\rho) > 0, \quad Q(\rho) > 0$$

$$\begin{bmatrix} P(\rho) & \frac{1}{\gamma} I \\ \frac{1}{\gamma} I & Q(\rho) \end{bmatrix} \geq 0$$

$$\begin{bmatrix} Q\bar{A}(\rho)^T + \bar{A}(\rho)Q - \sum_{i=1}^{n_\rho}\left(\pm\, \bar{\chi}_i \frac{\partial P}{\partial \rho_i}\right) - B_2(\rho)B_2(\rho)^T & QC_{11}(\rho)^T & \frac{1}{\gamma}B_1(\rho) \\ C_{11}(\rho)^T Q & -I & 0 \\ B_1(\rho)^T & 0 & -I \end{bmatrix} < 0 \quad (2.15)$$

$$\begin{bmatrix} \tilde{A}(\rho)^T P + P\tilde{A}(\rho) - \sum_{i=1}^{n_\rho}\left(\pm\, \bar{\chi}_i \frac{\partial Q}{\partial \rho_i}\right) - C_2(\rho)^T C_2(\rho) & PB_{11}(\rho) & \frac{1}{\gamma}C_1(\rho)^T \\ B_{11}(\rho)^T P & -I & 0 \\ C_1(\rho) & 0 & -I \end{bmatrix} < 0$$

where $\bar{A}(\rho) := A(\rho) - B_2(\rho)C_{12}(\rho)$ *and* $\tilde{A}(\rho) := A(\rho) - B_{12}(\rho)C_2(\rho)$*.*

Proof. see, [Wu, 1995] □

To construct the LPV controller $K(\rho)$, the following auxiliary variables are defined:

$$\begin{aligned} F(\rho) &= -\big(B_2^T(\rho)P^{-1} + C_{12}(\rho)\big) \\ L(\rho) &= -\big(Q^{-1}C_2(\rho) + B_{12}(\rho)\big) \\ H(\rho,\dot{\rho}) &= -\Big(P^{-1}A_F + A_F^T P^{-1} + \sum_{i=1}^{n_\rho} \dot{\rho}_i \frac{\partial P^{-1}}{\partial \rho_i} + C_F^T C_F - \gamma^{-2}P^{-1}B_1B_1^T P^{-1}\Big) \qquad (2.16) \\ X(\rho) &= Q - \gamma^{-2}P^{-1} > 0 \\ M(\rho,\dot{\rho}) &= H(\rho,\dot{\rho}) - \gamma^{-2}(QL(\rho)D_{21} + X(\rho)B_1)B_1^T P^{-1} \end{aligned}$$

where

$$A_F(\rho) = A(\rho) + B_2(\rho)F(\rho), \quad C_F(\rho) = C_1(\rho) + D_{12}F(\rho)$$

The LPV output feedback controller is obtained from

$$\begin{aligned} A_C(\rho) &= A + B_2F(\rho) + X^{-1}(\rho)QL(\rho)C_2 - \gamma^{-2}X^{-1}(\rho)M(\rho,\dot{\rho}) \\ B_C(\rho) &= -X^{-1}(\rho)QL(\rho) \\ C_C(\rho) &= F(\rho) \\ D_C(\rho) &= 0. \end{aligned} \qquad (2.17)$$

2.4.2 Discrete-time LPV System Stability and Performance

Consider a discrete-time LPV system $G(\theta(k))$,

$$\begin{bmatrix} x(k+1) \\ z(k) \\ e(k) \end{bmatrix} = \begin{bmatrix} A(\theta(k)) & B_r(\theta(k)) & B_u(\theta(k)) \\ C_z(\theta(k)) & D_{zr}(\theta(k)) & D_{zu}(\theta(k)) \\ C_e(\theta(k)) & D_{er}(\theta(k)) & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ r(k) \\ u(k) \end{bmatrix} \qquad (2.18)$$

where $x(k) \in \mathbb{R}^n, r(k) \in \mathbb{R}^{n_r}, u(k) \in \mathbb{R}^{n_u}, z(k) \in \mathbb{R}^{n_z}$ and $e(k) \in \mathbb{R}^{n_e}$ and

$$\theta = [\theta_1, \ldots, \theta_{n_\theta}]^T \in \ \mathcal{P}_\theta \in \mathbb{R}^{n_\theta}$$

Consider a discrete-time output feedback LPV controller $K(\theta(k))$,

$$\begin{bmatrix} x_C(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} A_C(\theta(k)) & B_C(\theta(k)) \\ C_C(\theta(k)) & D_C(\theta(k)) \end{bmatrix} \begin{bmatrix} x_C(k) \\ e(k) \end{bmatrix} \tag{2.19}$$

where $x_C \in \mathbb{R}^{n_{x_C}}$. The feedback interconnection between the controller (2.19) and the system plant (2.18) results in the closed loop system:

$$\begin{bmatrix} x_{cl}(k+1) \\ z(k) \end{bmatrix} = \begin{bmatrix} A_{cl}(\theta(k)) & B_{cl}(\theta(k)) \\ C_{cl}(\theta(k)) & D_{cl}(\theta(k)) \end{bmatrix} \begin{bmatrix} x_{cl}(k) \\ r(k) \end{bmatrix} \tag{2.20}$$

The stability and guaranteed ℓ_2 performance of the closed loop system (2.20) are summarized using the following theorem:

Theorem 2.2. *The LPV system (2.20) is stable and has ℓ_2-gain less than γ if there exist symmetric matrices $P(\theta)$ and $Q(\theta)$ such that*

$$\begin{bmatrix} P(\theta) & A_{cl}(\theta(k))Q(\theta(k)) & B_{cl}(\theta(k)) & 0 \\ \star & -P(\theta(k)) + Q^T(\theta(k)) + Q(\theta(k)) & 0 & Q^T(\theta(k))C_{cl}^T(\theta(k)) \\ \star & \star & \gamma I & D_{cl}^T(\theta(k)) \\ \star & \star & \star & \gamma I \end{bmatrix} > 0 \tag{2.21}$$

holds $\forall \theta(k) \in \mathcal{P}_\theta$.

Proof. see, [de Souza et al., 2006] □

In literature, numerous approaches deal with the discrete-time LPV controller synthesis [De Caigny et al., 2012, de Souza et al., 2006]. For a polytopic LPV model representation, a controller $K(\theta)$ can be constructed based on the following assumptions of [Ali and Werner, 2011],

Assumption 1. *The matrix $A(\theta)$ is affine on the scheduling parameter θ.*

Assumption 2. *$B_u(\theta), C_e(\theta),\ D_{zu}(\theta),\ D_{er}(\theta)$ are parameter independent.*

If assumption (2) is not satisfied, a pre- or post-filter can be used. The following algorithm is used based on the previous assumptions:

Algorithm 1: Discrete-time LPV controller synthesis for polytopic models

Step 1: Solve the optimization problem,

$$\min_{U_i,V_i,W_i,X_i,P_{x_i},P_{y_i},P_{z_i},N,J,S} \quad \gamma$$

$$\text{s.t.} \begin{bmatrix} P_{x_i} & P_{y_i} & AJ_i + B_uW_i & A + B_uX_iC_e & B_e + B_uX_iD_{er} & 0 \\ \star & P_{z_i} & U_i & N_iA + V_iC_e & N_iB_r + VD_{er} & 0 \\ \star & \star & J_i + J_i^T - P_{x_i} & I + S_i^T - P_{y_i} & 0 & J_i^TC_z^T + W_i^TD_{zu}^T \\ \star & \star & \star & N_i + N_i^T - P_{z_i} & 0 & C_z^T + C_e^TX_i^TD_{zu}^T \\ \star & \star & \star & \star & I & D_{zr}^T + D_{er}^TX_i^TD_{zu}^T \\ \star & \star & \star & \star & \star & \gamma I \end{bmatrix} > 0$$

$$\text{holds}, \forall i = 1, \ldots, n_\theta.$$

Step 2: Find

$$\begin{bmatrix} A_{C_i} & B_{C_i} \\ C_{C_i} & D_{C_i} \end{bmatrix}, \quad \forall i = 1, \ldots, n_\theta$$

such that

$$\begin{bmatrix} U_i & V_i \\ W_i & X_i \end{bmatrix} = \begin{bmatrix} R_i & N_iB_u \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{C_i} & B_{C_i} \\ C_{C_i} & D_{C_i} \end{bmatrix} \begin{bmatrix} L_i & 0 \\ C_{y_i}J_i & I \end{bmatrix} + \begin{bmatrix} N_i \\ 0 \end{bmatrix} A \begin{bmatrix} J_i & 0 \end{bmatrix}$$

$$S_i = N_iJ_i + R_iL_i$$

Step 3: Construct the LPV controller $K(\theta)$ where

$$\begin{bmatrix} A_C(\theta) & B_C(\theta) \\ C_C(\theta) & D_C(\theta) \end{bmatrix} = \sum_{i=1}^{n_\theta} \begin{bmatrix} A_{C_i} & B_{C_i} \\ C_{C_i} & D_{C_i} \end{bmatrix} \theta_i$$

Chapter 3

Agent LPV Modeling and Control

In an MAS, a single agent can be modeled as a single or double integrator using simple dynamics. In addition, they can be represented as LTI models. For nonlinear dynamics, simple models of nonholonomic agents can be used like kinematic or dynamic unicycles that describe the movement of realistic robots [Chen et al., 2004], fixed-wing aircrafts [Lugo-Cárdenas et al., 2017],. . . etc. In addition, the realistic nonlinear models of such complex agents can be used. In this sense, LPV modeling leads to efficient gain scheduled control schemes. In this chapter, a dynamic unicycle moving in a plan and realistic model of an autonomous underwater vehicle are represented as the *quasi*-LPV models, and the gain scheduled controllers are designed and synthesized based on LPV gridding techniques.

3.1 Dynamic Unicycle

The dynamic unicycle models cover the nonholonomic constraints in a simple way which can describe the movement of many vehicles. A state space representation of their Equations of Motion (EoMs) is given by

$$\begin{aligned}
\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \\ \dot{\psi} \\ \dot{v}_n \\ \dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & \cos\psi & 0 \\ 0 & 0 & 0 & \sin\psi & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ \psi \\ v_n \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} f \\ \tau_\psi \end{bmatrix} \\
y &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ \psi \\ v_n \\ \omega \end{bmatrix}
\end{aligned} \tag{3.1}$$

where (q_x,q_y) is the center of gravity (c.g.) position of the dynamic unicycle in the xy-plane, $\psi \in [0, 2\pi)$ is the yaw angle with respect to the x-axis, and v_n and ω are the linear and angular velocities. m is the mass of the dynamic unicycle, and I_z is the moment of inertia. f is the thrusting force, and τ_ψ is the yaw torque, Figure 3.1.

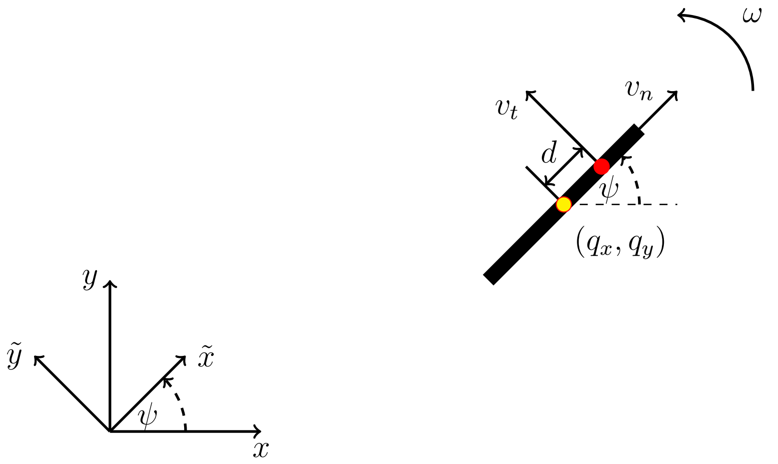

Figure 3.1: Dynamic Unicycle

3.1.1 Dynamic Unicycle: ***LPV modeling***

To obtain an LPV model of a dynamic unicycle, we first apply a translation of c.g. to the handle point (marked in red) at a small distance d ahead with coordinates

$$\begin{bmatrix} q_{x_d} \\ q_{y_d} \end{bmatrix} = \begin{bmatrix} q_x \\ q_y \end{bmatrix} + d \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix}. \tag{3.2}$$

Its dynamics are obtained by taking the time derivative of equation (3.2),

$$\begin{aligned} \dot{q}_{x_d} &= \dot{q}_x - d\omega \sin\psi = v_n \cos\psi - v_t \sin\psi \\ \dot{q}_{y_d} &= \dot{q}_x + d\omega \cos\psi = v_n \sin\psi + v_t \cos\psi \\ \dot{\psi} &= \frac{v_t}{d} \\ \dot{v}_n &= \frac{f}{m} \\ \dot{v}_t &= d\frac{\tau}{I_z} \end{aligned} \tag{3.3}$$

where $v_t = d\omega$ is the tangential velocity of the handle point, and its normal velocity is v_n because of rigid body properties. One method to obtain an LPV model is to choose the trigonometric functions as scheduling parameters; however, this will lead to a conservative solution. To overcome this, a Counter Clockwise (CCW) orthogonal transformation is applied to transform the system from xy-coordinates to the $\tilde{x}\tilde{y}$-coordinate system [Lawton et al., 2003, Cortés and Egerstedt, 2017, Gonzalez et al., 2015]. This can be done using the orthonormal transformation matrix

$$R_\psi = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}$$

and

$$\begin{bmatrix} \tilde{q}_{x_d} \\ \tilde{q}_{y_d} \end{bmatrix} = R_\psi \begin{bmatrix} q_{x_d} \\ q_{y_d} \end{bmatrix}. \tag{3.4}$$

Starting from equation (3.2) and multiplying from the right by R_ψ,

$$\begin{aligned} q_{x_d}\cos\psi + q_{y_d}\sin\psi &= q_x\cos\psi + q_y\sin\psi + d(\cos^2\psi + \sin^2\psi) \\ -q_{x_d}\sin\psi + q_{y_d}\cos\psi &= -q_x\sin\psi + q_y\cos\psi + d(-\sin\psi\cos\psi + \cos\psi\sin\psi) \end{aligned} \tag{3.5}$$

which leads to

$$\begin{aligned} \tilde{q}_{x_d} &= q_x\cos\psi + q_y\sin\psi + d \\ \tilde{q}_{y_d} &= -q_x\sin\psi + q_y\cos\psi. \end{aligned} \tag{3.6}$$

Take the time derivative,

$$\begin{aligned} \dot{\tilde{q}}_{x_d} &= \dot{q}_x\cos\psi - q_x\dot{\psi}\sin\psi + \dot{q}_y\sin\psi + q_y\dot{\psi}\cos\psi \\ \dot{\tilde{q}}_{y_d} &= -\dot{q}_x\sin\psi - q_x\dot{\psi}\cos\psi + \dot{q}_y\cos\psi - q_y\dot{\psi}\sin\psi \end{aligned} \tag{3.7}$$

and substitute from equation (3.1) for $\dot{q}_x$ and $\dot{q}_y$ in equation (3.7) to obtain

$$\begin{aligned} \dot{\tilde{q}}_{x_d} &= v_n + \dot{\psi}(-q_x\sin\psi + q_y\cos\psi) \\ \dot{\tilde{q}}_{y_d} &= -v_n\cos\psi\sin\psi + v_n\sin\psi\cos\psi - \dot{\psi}(q_x\cos\psi + q_y\sin\psi) \end{aligned} \tag{3.8}$$

Finally, substitute from equation (3.6) in equation (3.8) and use $\dot{\psi} = \omega = \frac{v_t}{d}$ to get

$$\begin{aligned} \dot{\tilde{q}}_{x_d} &= v_n + \frac{v_t}{d}\tilde{q}_{y_d} \\ \dot{\tilde{q}}_{y_d} &= -\frac{v_t}{d}(\tilde{q}_{x_d} - d). \end{aligned} \tag{3.9}$$

A *quasi*-LPV model of the dynamic unicycle is then given as follows:

$$\begin{aligned} \begin{bmatrix} \dot{\tilde{q}}_{x_d} \\ \dot{\tilde{q}}_{y_d} \\ \dot{v}_n \\ \dot{v}_t \end{bmatrix} &= \begin{bmatrix} 0 & \frac{v_t}{d} & 1 & 0 \\ \frac{-v_t}{d} & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{q}_{x_d} \\ \tilde{q}_{y_d} \\ v_n \\ v_t \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{d}{I_z} \end{bmatrix} \begin{bmatrix} f \\ \tau_\psi \end{bmatrix} \\ \tilde{y} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{q}_{x_d} \\ \tilde{q}_{y_d} \\ v_n \\ v_t \end{bmatrix} \end{aligned} \tag{3.10}$$

It has only one scheduling parameter $\rho = v_t$. $d \neq 0$ guarantees controllability of the system for frozen parameters. A discrete-time *quasi*-LPV model is obtained by Euler discretization with sampling time T_s, and with scheduling parameter $\theta(k) = v_t(k)$,

$$\begin{aligned} \begin{bmatrix} \tilde{q}_{x_d}(k+1) \\ \tilde{q}_{y_d}(k+1) \\ v_n(k+1) \\ v_t(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & \frac{T_s}{d}v_t(k) & T_s & 0 \\ \frac{-T_s}{d}v_t(k) & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{q}_{x_d}(k) \\ \tilde{q}_{y_d}(k) \\ v_n(k) \\ v_t(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T_s\frac{1}{m} & 0 \\ 0 & T_s\frac{d}{I_z} \end{bmatrix} \begin{bmatrix} f \\ \tau_\psi \end{bmatrix} \\ \tilde{y}(k) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{q}_{x_d}(k) \\ \tilde{q}_{y_d}(k) \\ v_n(k) \\ v_t(k) \end{bmatrix} \end{aligned} \tag{3.11}$$

A three-dimensional equivalent *quasi*-LPV model for a dynamic unicycle moving in three-dimensional space, is provided in Appendix B.

3.1.2 Dynamic Unicycle: ***LPV controller design***

A continuous-time output feedback LPV controller $K(\rho)$ is designed and synthesized using Theorem 2.1, and to meet the conditions of the generalized plant structure in (2.11), loop transformation is considered, which is automatically performed when using LPVtools [Hjartarson et al., 2015]. Closed loop reference tracking is shown in Figure 3.2 with added performance channel z and shaping filters, W_s and W_{ks}, for shaping sensitivity and control sensitivity, respectively. This approach aims at shaping the sensitivity using a low-pass filter that enforces integral action in the loop. In addition, the cut-off frequency of this filter dictates the closed-loop bandwidth while limiting control action using another high pass filter that constrains the control action at low frequencies and enforces roll-off at high frequencies.

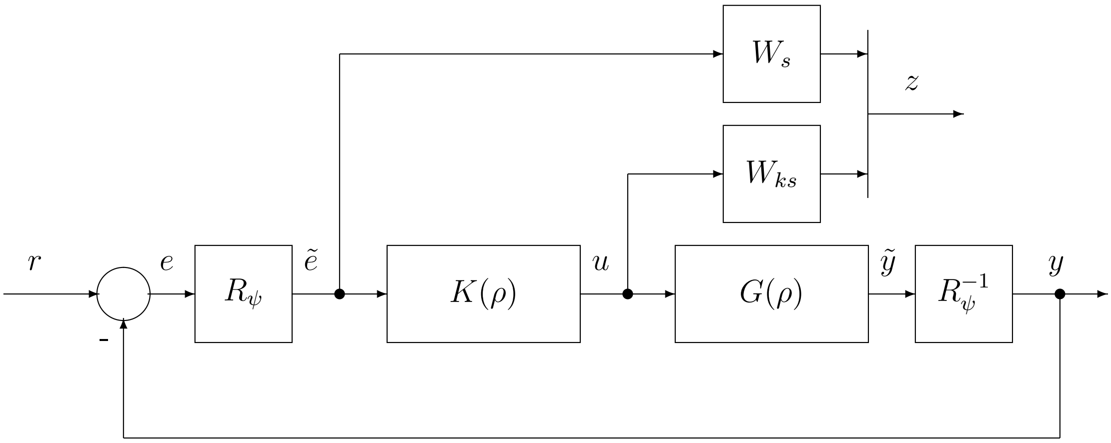

Figure 3.2: Reference tracking loop for dynamic unicycle

To form the generalized plant $P(\rho)$, both orthonormal transforming blocks are pulled out from the loop, Figure 3.3. The closed loop remains the same under orthonormal transformation, and the generalized plant is shown in Figure 3.4.

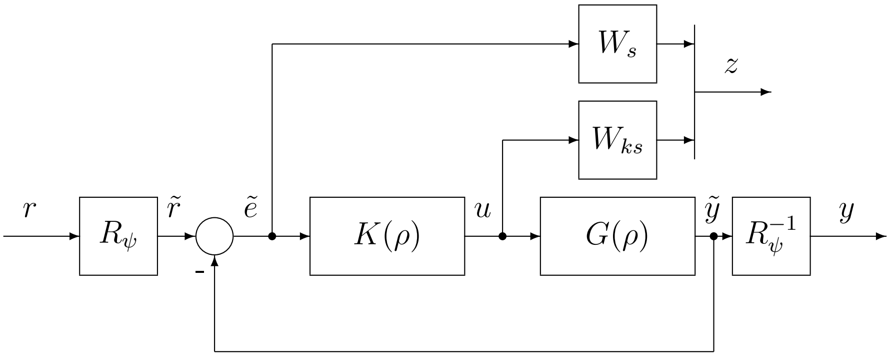

Figure 3.3: Loop after pulling the orthonormal blocks out

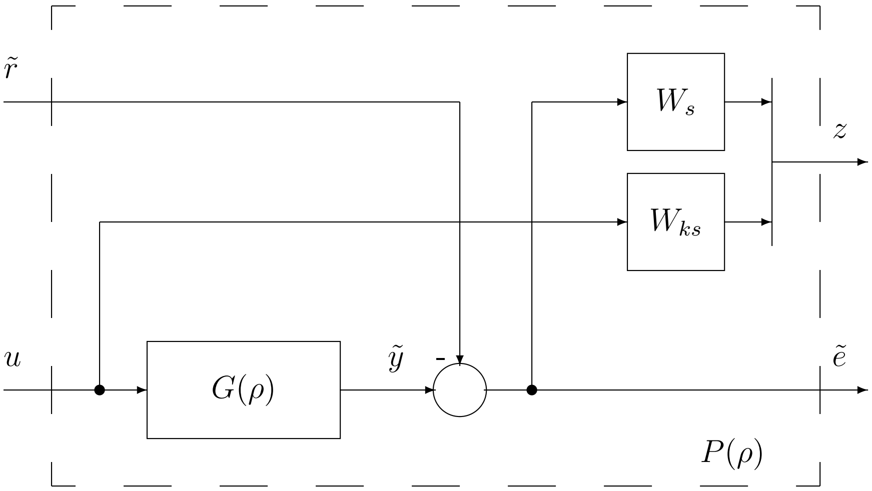

Figure 3.4: Generalized plant for the dynamic unicycle

The closed loop performance is given by $\gamma = 0.8455$. The tuned shaping filters are

$$W_s = \frac{0.8s + 10^{-1}}{s + 10^{-4}}, \quad W_{ks} = \frac{0.7s + 10}{s + 10^4} \tag{3.12}$$

The controller implementation is shown in Figure 3.5, and Figure 3.6 shows the sigma plots of loop shaping for parameter range $\rho = [-1 \ 1]$. A trajectory tracking simulation for the dynamic unicycle is shown in Figure 3.7, and the orientation during the trajectory is shown in Figure 3.8.

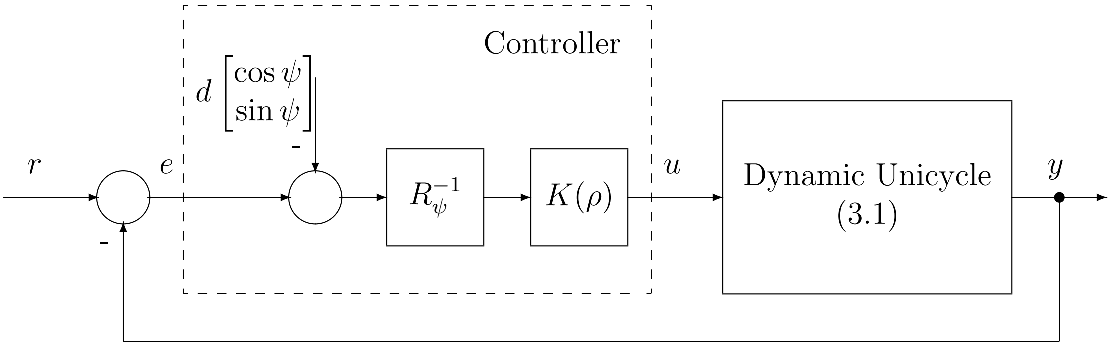

Figure 3.5: Closed loop with controller implementation

In Figure 3.9 , simulation results with different speeds are shown. In these cases, the maximum required speed is limited to $\leq 1.25\ m/sec$. Furthermore, the tracking error in the different cases for the same track shape is shown in Figure 3.10.

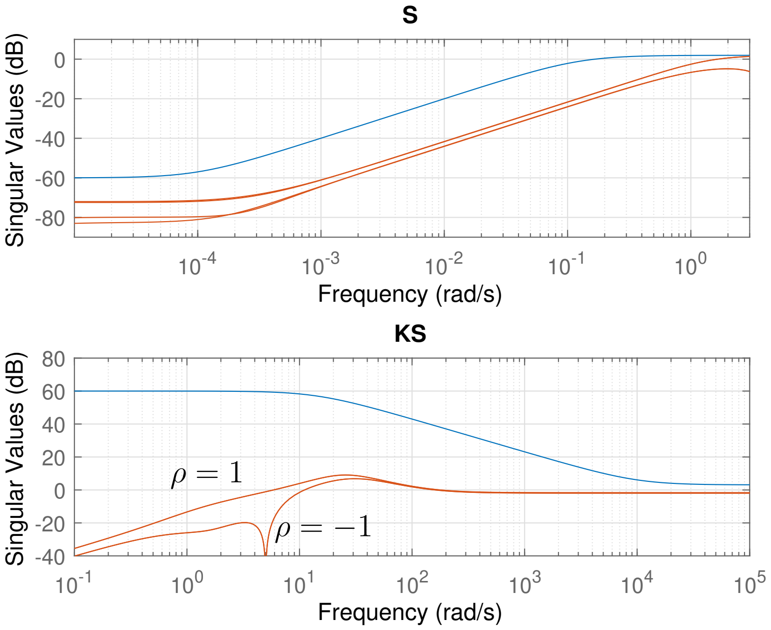

Figure 3.6: Sensitivity and control sensitivity sigma plots, closed loop sensitivity/control sensitivity (—), inverted filters, (—)

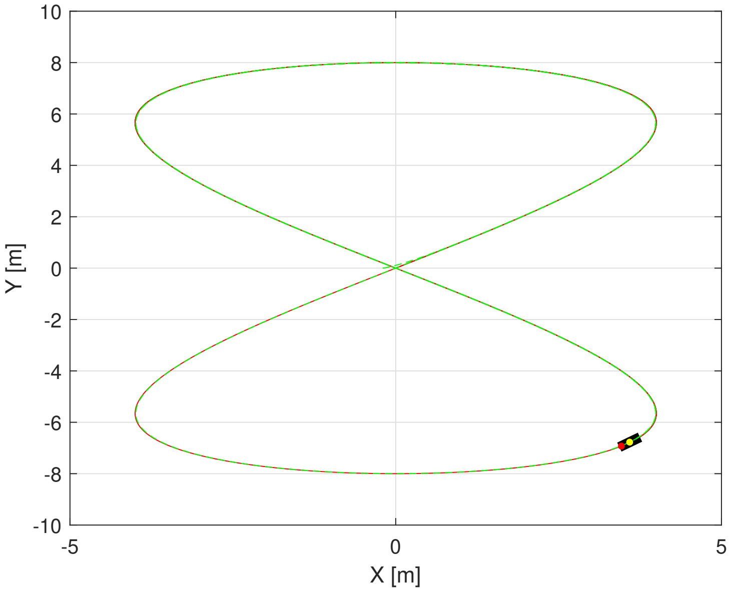

Figure 3.7: Trajectory tracking of the dynamic unicycle. Reference (—), and trajectory (−−)

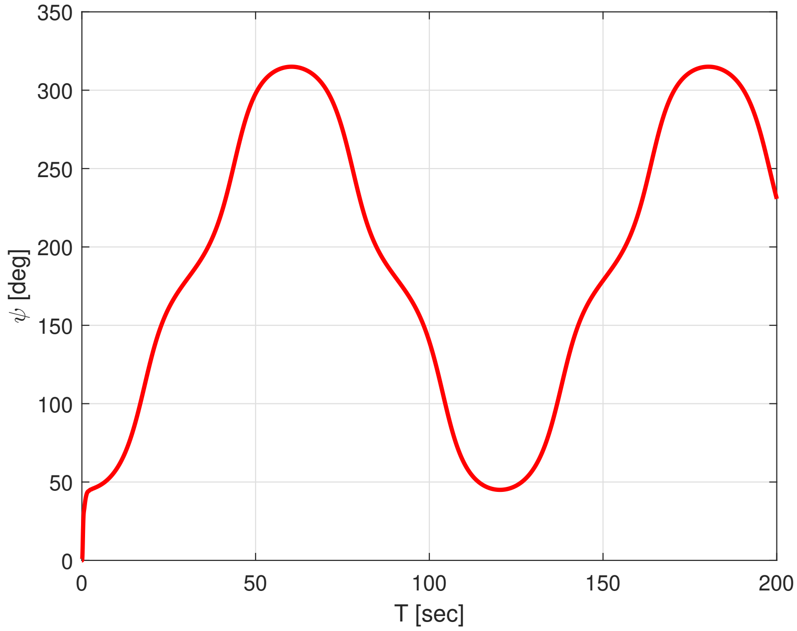

Figure 3.8: Orientation of the dynamic unicycle during the trajectory

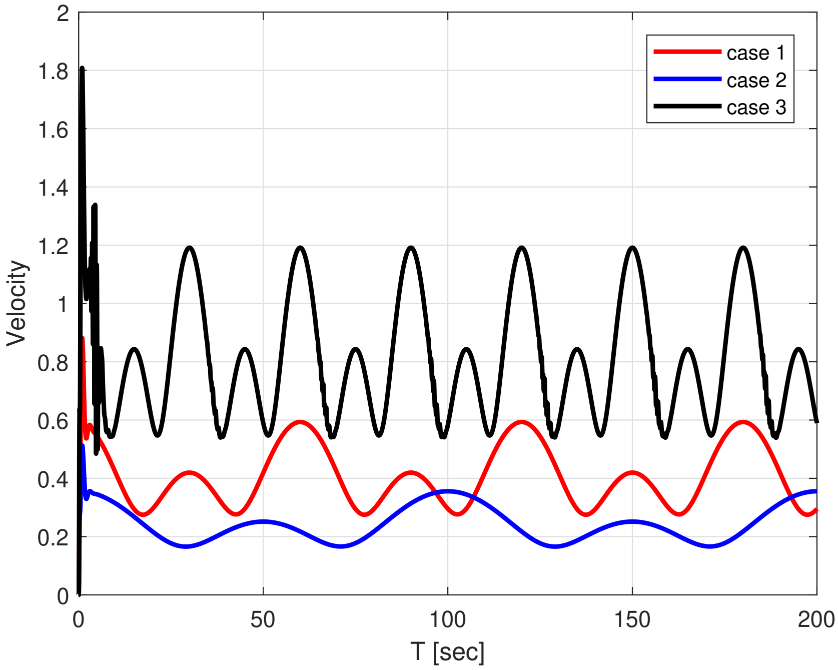

Figure 3.9: Linear velocity of dynamic unicycle

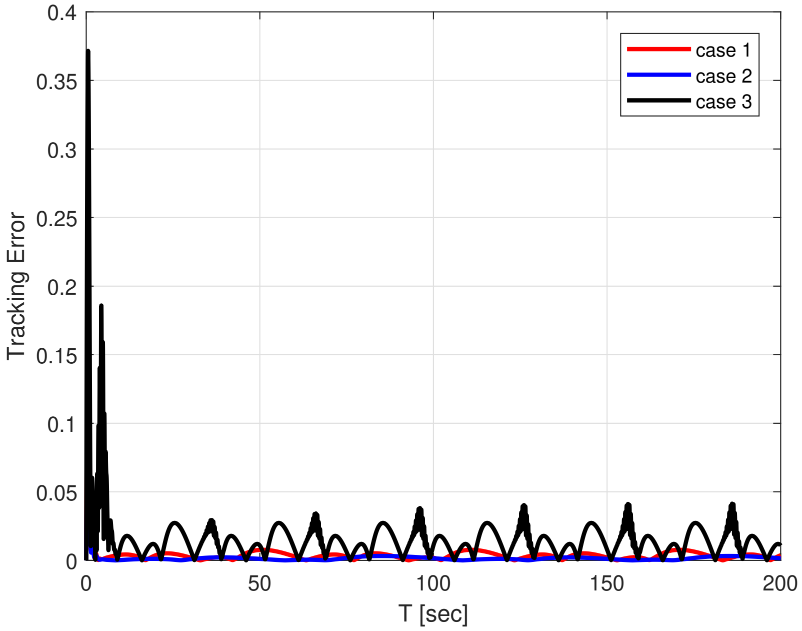

Figure 3.10: Tracking error during trajectory

3.2 Autonomous Underwater Vehicle

Another model for unmanned vehicles considered in this context is an autonomous underwater vehicle. The general nonlinear dynamics of an autonomous under water vehicle are formulated in [Fossen, 2011] in North-East-Down inertial coordinates $\{\mathbf{I}\}$ with orthonormal vectors $\{i_N, i_E, i_D\}$ and in body-fixed coordinates $\{\mathbf{B}\}$ with orthonormal vectors $\{x_B, y_B, z_B\}$ centered at body center of gravity see Figure 3.2.

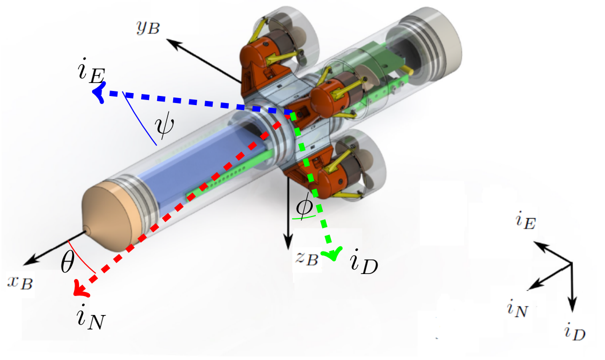

Figure 3.11: *Hippocampus* [Duecker et al., 2018] with inertial and body coordinates

Position and orientation are combined into the vector $\eta = [q_x \; q_y \; q_z \; \phi \; \theta \; \psi]^T$, and linear and angular velocities are combined into the vector $\nu = [v_{x_B} \; v_{y_B} \; v_{z_B} \; \omega_{x_B} \; \omega_{y_B} \; \omega_{z_B}]^T$. The dynamics of the AUV is controlled by four propellers that produce the thrust force and moments, which are combined into the vector $\tau = [f \;\; 0 \;\; 0 \;\; \tau_\phi \;\; \tau_\theta \;\; \tau_\psi]^T$. A nonlinear state space model is then given as

$$\begin{bmatrix} \dot{\eta} \\ \dot{\nu} \end{bmatrix} = \begin{bmatrix} 0 & J(\eta) \\ -M^{-1}G(\eta) & -M^{-1}(C(\nu) + D(\nu)) \end{bmatrix} \begin{bmatrix} \eta \\ \nu \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \tau, \tag{3.13}$$

This nonlinear model is adapted for the experimental *Hippocampus* vehicle [Duecker et al., 2018], where $J(\eta) = blkdiag(R_{\phi\theta\psi}, T_{\phi\theta\psi})$ is a block diagonal matrix of the rotational tensor $R_{\phi\theta\psi}$ and the angular velocity transformation tensor $T_{\phi\theta\psi}$ such that

$$R_{\phi\theta\psi} = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ s_\psi c_\theta & c_\psi c_\phi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + s_\theta s_\psi c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

$$T_{\phi\theta\psi} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}$$

where $c., s., t.$ denotes $\cos(\cdot), \sin(\cdot), \tan(\cdot)$, respectively. $G(\eta)$ represents the hydrostatic load and is defined as

$$G(\eta) = diag\Big(0, 0, 0, z_g mgc_\theta sinc(\phi), z_g mgsinc(\theta), 0\Big)$$

where m is the AUV mass and g the gravitational acceleration; $M = M_{RB} + M_A$ is the inertia matrix due to rigid body M_{RB} and added mass due to hydrodynamic loads M_A, where

$$M_{RB} = diag(m, m, m, I_x, I_y, I_z)$$

$$M_A = -diag(X_{\dot{v}_{x_B}}, Y_{\dot{v}_{y_B}}, Z_{\dot{v}_{z_B}}, K_{\dot{\omega}_{x_B}}, M_{\dot{\omega}_{y_B}}, N_{\dot{\omega}_{z_B}}).$$

I_x,I_y, and I_z are the moments of inertia around the different axes, and $D(\nu)$ is the damping matrix. Here,

$$D(\nu) = -diag(X_{|v_{x_B}|}|v_{x_B}|, Y_{|v_{y_B}|}|v_{y_B}|, Z_{|v_{z_B}|}|v_{z_B}|, K_{|\omega_{x_B}|}|\omega_{x_B}|, M_{|\omega_{y_B}|}|\omega_{y_B}|, N_{|\omega_{z_B}|}|\omega_{z_B}|)$$

and $C(\nu)$ is the Coriolis matrix. The parameters of the *Hippocampus* model are identified based on experimental measurements and are shown in Table 3.1 [Duecker et al., 2018]. Because of the symmetric shape of the *Hippocampus* and low speed operation environment ($v_{x_{B\,max}} = 1.5 m/sec$), some coupling hydrodynamic parameters have been removed. The buoyancy force is acting on the *Hippocampus* in a point at a distance of z_g above its center of gravity, which results in an inverse moment if it rotates around its longitudinal body axis x_B; thus, the roll angle ϕ is maintained at nearly zero.

Table 3.1: *Hippocampus* parameters

Parameter	Value	Parameter	Value
m	$1.43kg$	$X_{\lvert v_{x_B}\rvert}$	$-4.56kg/m$
I_x	$0.00241kgm^2$	$Y_{\lvert v_{y_B}\rvert} = Z_{\lvert v_{z_B}\rvert}$	$-17.36kg/m$
I_y	$0.01072kgm^2$	$K_{\lvert \omega_{x_B}\rvert}$	$-0.0028kgm^2$
I_z	$0.01072kgm^2$	$M_{\lvert \omega_{y_B}\rvert} = N_{\lvert \omega_{z_B}\rvert}$	$-0.0476kgm^2$
$X_{\dot{v}_{x_B}}$	$-1.11kg$	$K_{\dot{\omega}_{x_B}}$	$-0.0018kgm^2$
$Y_{\dot{v}_{y_B}} = Z_{\dot{v}_{y_B}}$	$-2.80kg$	$M_{\dot{\omega}_{y_B}} = N_{\dot{\omega}_{z_B}}$	$-0.0095kgm^2$

3.2.1 AUV: *LPV modeling*

Quasi-LPV representations of nonlinear systems are not unique, and the selection of scheduling parameters affects both the synthesis complexity and performance [Kwiatkowski et al., 2006]. One approach is a Taylor series expansion for trigonometric functions which allows the selection of the angle itself as a scheduling parameter instead of a trigonometric function [Hoffmann, 2016]. As a trade-off between model accuracy and complexity, a second order Taylor expansion is selected. One can approximate the rotational tensors by

$$R_{\phi\theta\psi} \approx \begin{bmatrix} 1-\frac{\psi^2}{2}-\frac{\theta^2}{2}+\frac{\theta^2\psi^2}{4} & -\psi+\frac{\psi^3}{6} & \theta-\frac{\theta\psi^2}{2}-\frac{\theta^3}{6}+\frac{\theta^3\psi^2}{12} \\ \psi-\frac{\psi\theta^2}{2}-\frac{\psi^3}{6}+\frac{\theta^2\psi^3}{12} & 1-\frac{\psi^2}{2} & \psi\theta-\frac{\psi\theta^3}{6}-\frac{\theta\psi^3}{6}+\frac{\theta^3\psi^3}{36} \\ -\theta+\frac{\theta^3}{6} & 0 & 1-\frac{\theta^2}{2} \end{bmatrix};$$

$$T_{\phi\theta\psi} \approx \begin{bmatrix} 1 & 0 & \theta+\frac{\theta^3}{3} \\ 0 & 1 & 0 \\ 0 & 0 & 1-\frac{\theta^2}{2} \end{bmatrix}.$$

In addition,

$$G(\eta) \approx diag\Big(0,0,0,z_g mg(1-\frac{\theta^2}{2}), z_g mg(\theta-\frac{\theta^3}{6}),0\Big).$$

A closer look at the approximated nonlinear equations shows that terms like $\frac{\psi^2 v_{x_B}}{2}, \frac{\psi\theta^3 v_{z_B}}{6}, ...$ contain products of state variables, and one can decide which variable will be used as the state and which one as the scheduling parameter. To obtain a model with the most coupling information possible, these terms can be divided equally between the state variables as

$$\frac{\psi^2 v_{x_B}}{2} = \frac{\psi^2}{4} v_{x_B} + \frac{v_{x_B}\psi}{4}\psi.$$

Higher order terms like $\frac{\theta^3\psi^2 z_{x_B}}{12}, \frac{\theta^2\psi^3 v_{x_B}}{12}, \ldots$ are not divided due to their low contribution. The resulting model shows that the position dynamics depend on the orientation states and the thrust force. This makes it possible to decompose the whole system into a position subsystem $\mathcal{G}_{out}$ with state vector $\zeta_{out} = [q_x\ q_y\ q_z\ v_{x_B}\ v_{y_B}\ v_{z_B}]^T$, position control input vector $\tau_{out} = [f\ \theta\ \psi]^T$, output vector $y = [q_x\ q_y\ q_z]^T$ and a scheduling parameter vector $\rho_{out} = [\theta\ \psi\ v_{x_B}\ v_{y_B}\ v_{z_B}]^T$; and an orientation subsystem $\mathcal{G}_{in}$ with state vector $\zeta_{in} = [\phi\ \theta\ \psi\ \omega_{x_B}\ \omega_{y_B}\ \omega_{z_B}]^T$, orientation control input vector $\tau_{in} = [\tau_\phi\ \tau_\theta\ \tau_\psi]^T$, and its scheduling parameter vector $\rho_{in} = [\theta\ \omega_{x_B}\ \omega_{y_B}\ \omega_{z_B}]^T$. The resulting open loop *quasi*-LPV subsystems are represented in equations (3.14)-(3.15).

$$\mathcal{G}_{out}:\ \dot{\zeta}_{out} = \left[\begin{array}{c|ccc} 0 & 1-\frac{\psi^2}{4}-\frac{\theta^2}{4}+\frac{\theta^2\psi^2}{4} & -\frac{\psi}{2}+\frac{\psi^3}{6} & \frac{\theta}{2}-\frac{\theta\psi^2}{2}-\frac{\theta^3}{6}+\frac{\theta^3\psi^2}{12} \\ 0 & \frac{\psi}{2}-\frac{\psi\theta^2}{2}-\frac{\psi^3}{6}+\frac{\theta^2\psi^3}{12} & 1-\frac{\psi^2}{4} & \frac{\psi\theta}{3}-\frac{\psi\theta^3}{6}-\frac{\theta\psi^3}{6}+\frac{\theta^3\psi^3}{36} \\ 0 & -\frac{\theta}{2}+\frac{\theta^3}{6} & 0 & 1-\frac{\theta^2}{4} \\ 0 & \frac{-X_{|v_{x_B}|}|v_{x_B}|}{m-X_{\dot{v}_{x_B}}} & 0 & 0 \\ 0 & 0 & \frac{-Y_{|v_{y_B}|}|v_{y_B}|}{m-Y_{\dot{v}_{y_B}}} & 0 \\ 0 & 0 & 0 & \frac{-Z_{|v_{z_B}|}|v_{z_B}|}{m-Z_{\dot{v}_{z_B}}} \end{array}\right]\zeta_{out} + \begin{bmatrix} 0 & -\frac{u\theta}{4}+\frac{w}{2} & -\frac{u\psi}{4}-\frac{v}{2} \\ 0 & \frac{w\psi}{3} & \frac{u}{2}-\frac{v\psi}{4}+\frac{w\theta}{3} \\ 0 & -\frac{u}{2} & -\frac{w\theta}{4} \\ \frac{1}{m-X_{\dot{v}_{x_B}}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\tau_{out} \tag{3.14}$$

$$\mathcal{G}_{in}:\ \dot{\zeta}_{in} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & \theta+\frac{\theta^3}{3} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-\frac{\theta^2}{2} \\ z_g mg(1-\frac{\theta^2}{2}) & 0 & 0 & \frac{-K_{|\omega_{x_B}|}|\omega_{x_B}|}{I_x-K_{\dot{\omega}_{x_B}}} & 0 & 0 \\ 0 & z_g mg(\theta-\frac{\theta^3}{6}) & 0 & 0 & \frac{-M_{|\omega_{y_B}|}|\omega_{y_B}|}{I_y-M_{\dot{\omega}_{y_B}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-N_{|\omega_{z_B}|}|\omega_{z_B}|}{I_z-N_{\dot{\omega}_{z_B}}} \end{bmatrix}\zeta_{in} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_x-K_{\dot{\omega}_{x_B}}} & 0 & 0 \\ 0 & \frac{1}{I_y-M_{\dot{\omega}_{y_B}}} & 0 \\ 0 & 0 & \frac{1}{I_z-N_{\dot{\omega}_{z_B}}} \end{bmatrix}\tau_{in} \tag{3.15}$$

3.2.2 AUV: *control loop design*

Based on the cascaded structure, Figure 3.12, the inner and outer LPV controllers are designed and synthesized for an autonomous underwater vehicle.

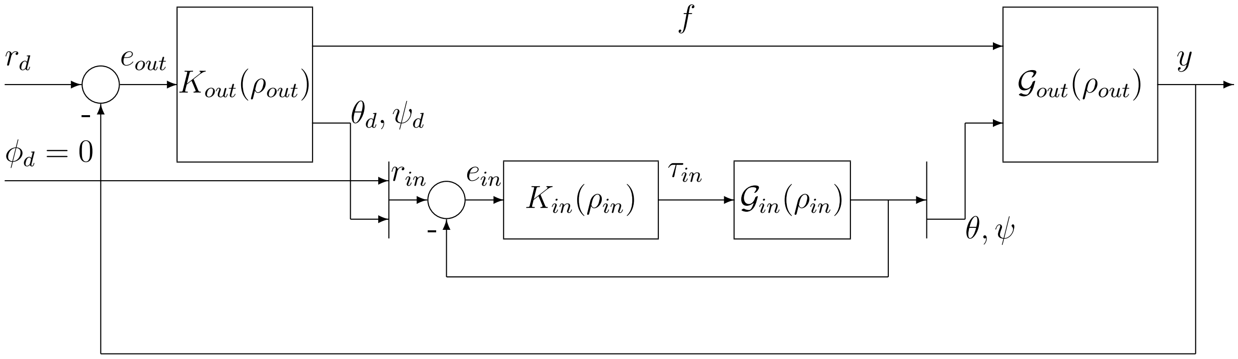

Figure 3.12: Cascaded tracking loop for the AUV

Both inner $K_{in}(\rho_{in})$ and outer $K_{out}(\rho_{out})$ controllers are designed based on the well-known S/KS approach, taking into account that the inner closed-loop bandwidth is chosen to be much faster than the outer closed-loop bandwidth. For the inner loop, shaping filters

$$W_s^{in} = 0.8\frac{s+75}{s+0.3},\ W_{ks}^{in} = 10^3\frac{s+1517}{s+1.011\times 10^7}. \tag{3.16}$$

are used to shape the sensitivity and control sensitivity, see Figure 3.13. corresponding to the required closed-loop bandwidth ω_s^{in}, required controller bandwidth ω_{ks}^{in}, sensitivity upper bound M_s^{in}, control sensitivity upper bound M_{ks}^{in}, and arbitrary constants c_s^{in}, c_{ks}^{in} are used as tuning knobs for achieving satisfactory performance.

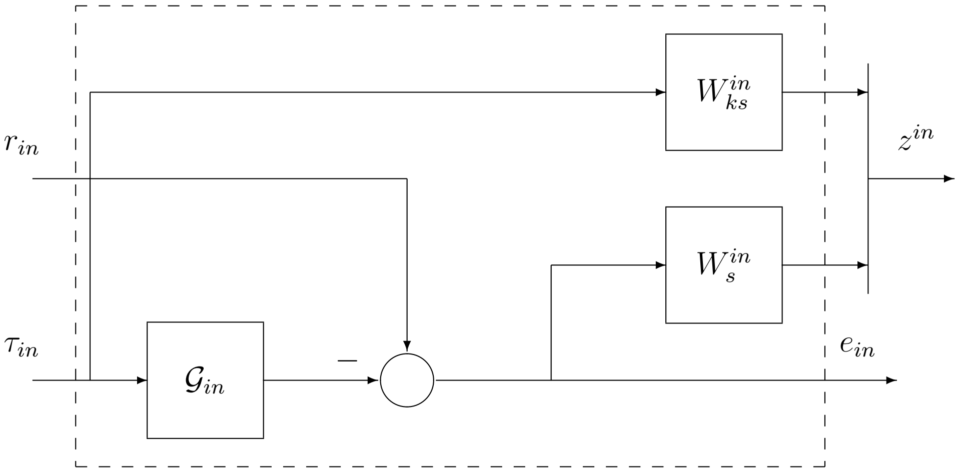

Figure 3.13: LPV S/KS loop shaping inner loop generalized plant

The design of the outer loop controller is based on the preceding inner loop design with shaping filters:

$$W_s^{out} = 0.8\frac{s+0.75}{s+0.0015},\ W_{ks}^{out} = 10^3\frac{s+60.3}{s+60.3\times 10^5}. \tag{3.17}$$

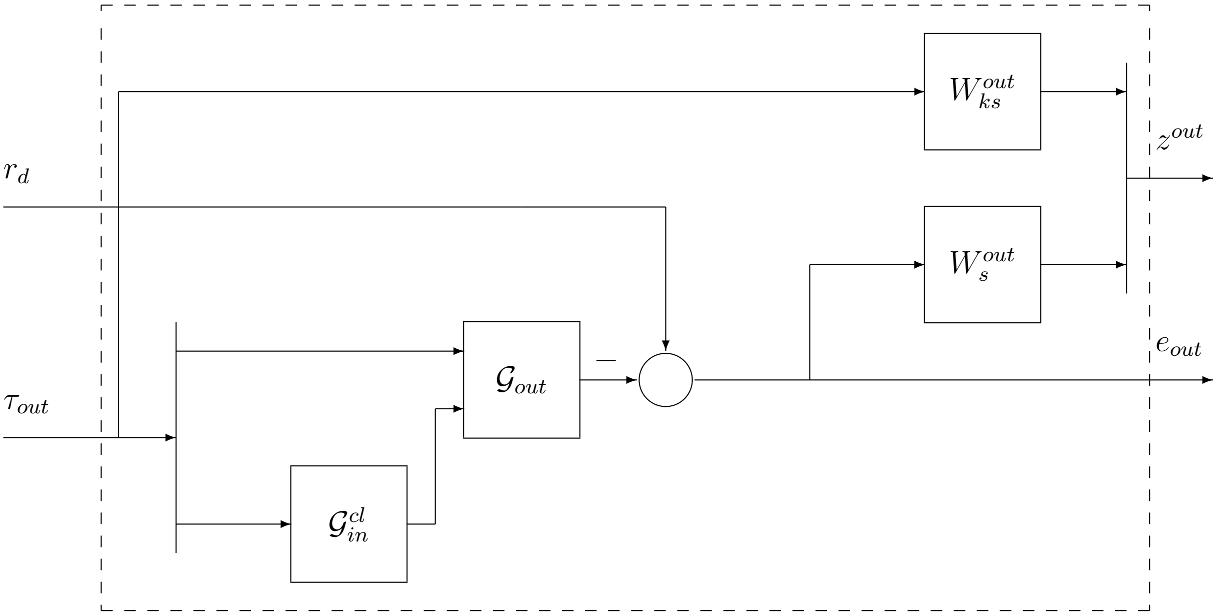

Figure 3.14: LPV S/KS loop shaping outer loop generalized plant

In cascaded design, the shaping filters are tuned using genetic algorithms, which achieve an inner closed-loop bandwidth $\omega^{in} = 163\ rad/sec$ and a performance index $\gamma_{in} = 4.133$ and for the outer loop, a closed-loop bandwidth $\omega^{out} = 1.02\ rad/sec$ and the overall performance $\gamma = \gamma_{out} = 2.6031$. Sigma plots of the sensitivity and control sensitivity are shown in Figure 3.15.

Simulation of a single underwater vehicle tracking a spiral trajectory with a radius of 3 m is shown in Figure 3.16. The simulation was carried out with the plant model (3.14),(3.14). The tracking errors when moving with different speeds of nearly 1.2 m/sec, 0.6 m/sec and 0.3 m/sec are shown Figure 3.17. In the first case, the tracking error is relatively high compared with that at low speeds.

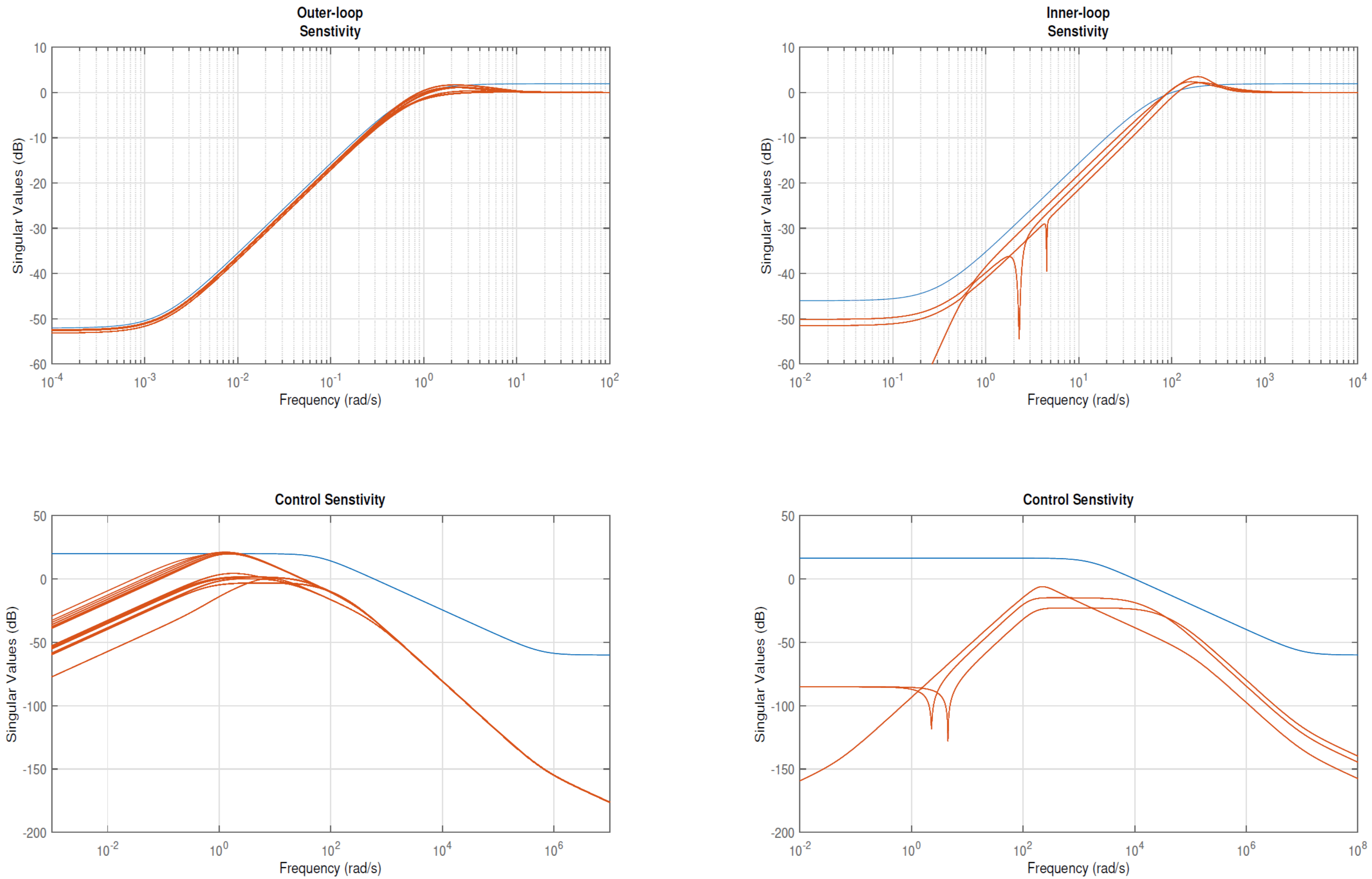

Figure 3.15: Sigma plots showing inner and outer sensitivity and control sensitivity; inverse shaping filters are in blue

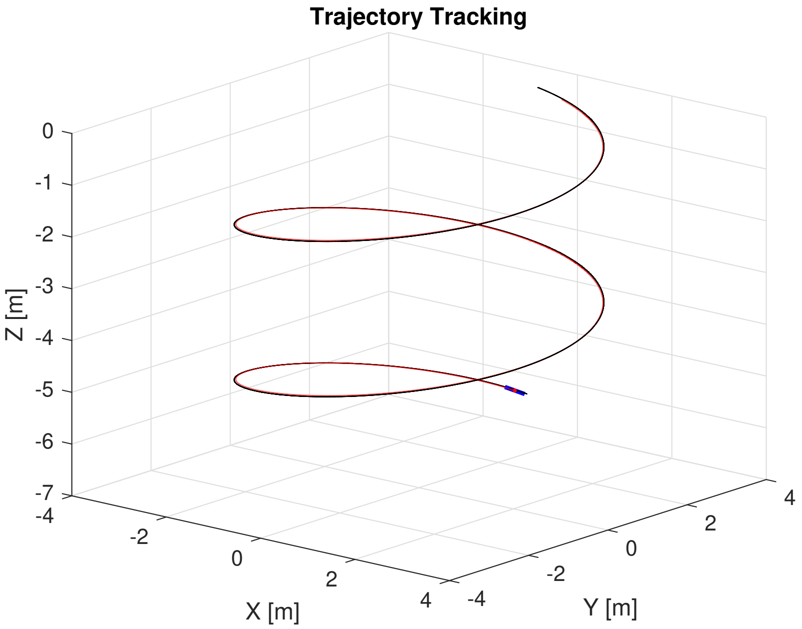

Figure 3.16: Tracking spiral trajectory, – for AUV path and – for reference trajectory

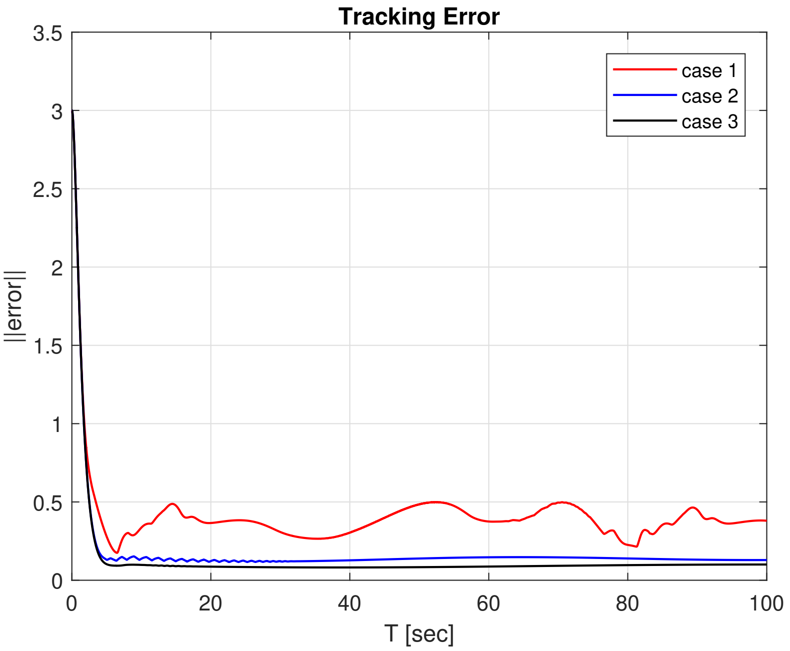

Figure 3.17: Tracking error

Chapter 4

Formation Control of LPV Agents

A distributed formation control scheme is proposed for a network of identical nonholonomic agents modeled as linear parameter-varying (LPV) systems. The design problem is formulated in the framework of discrete-time LPV control solved by a gridding technique. The agents exchange information through a communication graph which is time-invariant. In the robust design, only the minimum and maximum eigenvalues of the graph Laplacian are considered. A decomposition approach is employed which makes it possible to solve a synthesis problem that has the size of a single agent, regardless of the number of agents. The results presented here extend previous work on information flow through a group of LTI agents to LPV agents [Popov and Werner, 2012, Pilz et al., 2011, Bartels and Werner, 2014]. Moreover, a novel control architecture is used to synthesize the distributed control scheme and the information flow filter simultaneously, while allowing feedback from agents into the information flow dynamics. The design guarantees (marginal) stability and overall ℓ_2 performance of the closed-loop system. The simulation results illustrate the practicality and benefits of the proposed extended information flow scheme by applying it to a network of dynamic unicycles.

4.1 Information Flow in Multi-Agent Systems

The idea of an information flow filter is proposed for LTI agents to improve the cooperative behavior in MAS in [Fax and Murray, 2004], Figure 4.1. Each agent i in the network is equipped with a stabilizing controller K that provides a control signal u_t to the agent G, network filter R, and prediction filter H. Agents can communicate through a graph Laplacian $\mathcal{L}$ and adjacency matrix $\mathcal{A}$. The rule is based on agent position y_i and its predicted position w_i in addition to filtered data p_i from the network; the agent can take its new position in the formation. The analysis results are based on the assumption that an information flow filter $F = \frac{R}{1-R}$ is marginally stable and strictly proper; then, a filter $H = 1 - R$ can be obtained directly, which enables the separation in design for local controller on one hand and filters for each agent in the network. On the other hand, the framework is shown for LTI agents with the Nyquist criteria design method which is used for Single-Input-Single-Output (SISO) agents, assuming decoupled filters for each dimension when agents are moving in 2 or 3 dimensional space.

In such diagram, the relative formation error is filtered through filter R and added to the output of predictor filter H which is given to the controller K to produce a suitable control effort to the agent to adapt its position in the formation [Fax and Murray, 2004].

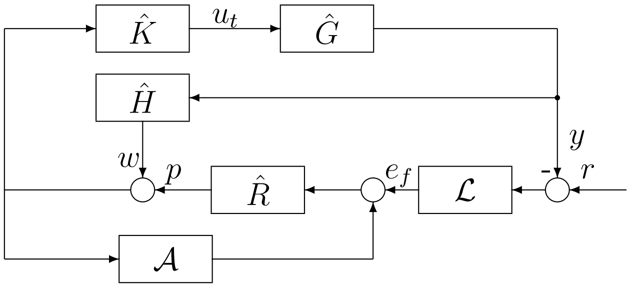

Figure 4.1: Information flow in a multi-agent system [Fax and Murray, 2004]

Following the same assumptions proposed in [Fax and Murray, 2004], previous studies [Pilz et al., 2011, Bartels and Werner, 2014] present a systematic way of designing both the local controller and filters for LTI agents based on μ-synthesis techniques. In this case, the whole network is separated into two loops, Figure 4.2, *abstraction level* that represents the virtual network loop and *physical layer* that represents the local tracking loop.

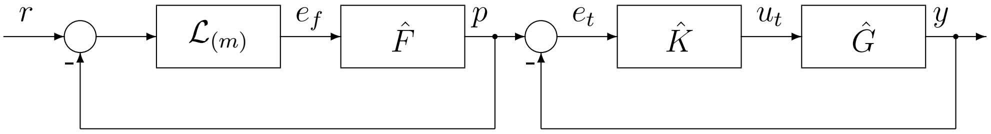

Figure 4.2: Information Flow in network (Separation Principle) [Pilz et al., 2011]

A major feature of this approach, from the viewpoint of controller design, is that information flow filter F and the controller K are designed independently. Thus, the controller is independent of the information flow filter and topology uncertainties and thus can be designed using standard methods. Unlike the heuristic approach of designing the filter in [Fax and Murray, 2004] which is realistic only for SISO systems, the μ-synthesis technique is proposed to design a filter, Figure 4.3, that guarantee robust stability of the MAS for an arbitrary number of agents and arbitrary communication topologies [Popov and Werner, 2012, Pilz et al., 2011].

Theorem 4.1. *Given a controller K that stabilizes G and a filter F, a MAS is stable for any number of agents N and for any communication topology if there exists an invertible matrix $D \in \mathbb{R}^{m \times m}$ such that $\|DT_{vz_\delta}D^{-1}\|_\infty < 1$, where T_{vz_δ} is the closed-loop transfer function from w_i to z_{δ_i}.*

Proof. [Popov and Werner, 2012, Pilz et al., 2011] □

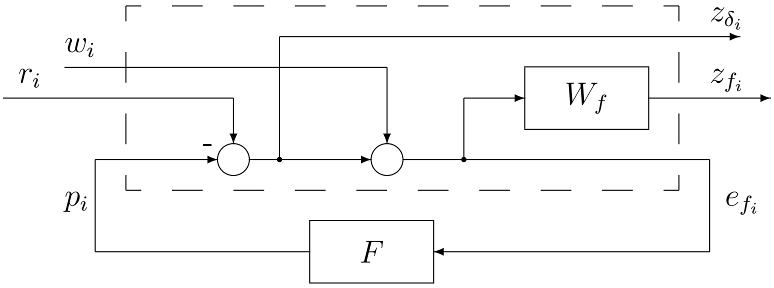

Figure 4.3: Robust filter design loop [Pilz et al., 2011]

4.2 Coupled Information Flow in Network of LPV Agents

When designing a filter and controller separately, extension to the LPV systems is easily achieved by designing a local LPV controller for a physical layer and a robust LTI filter for the network loop. However, the disadvantage is that the network level is not aware of what happens at the agent level, which remains in addition to the nonholonomic constraints which are not reflected in the virtual network dynamics.

In this section, coupling between the physical layer and network level is achieved through a coupling controller $K_C(\rho)$. In this case, the controller inputs are the formation error e_{f_i} and the tracking error e_{t_i}; the controller output is the filter output (virtual reference signal) p_i and the local control input u_{t_i}, Figure 4.4.

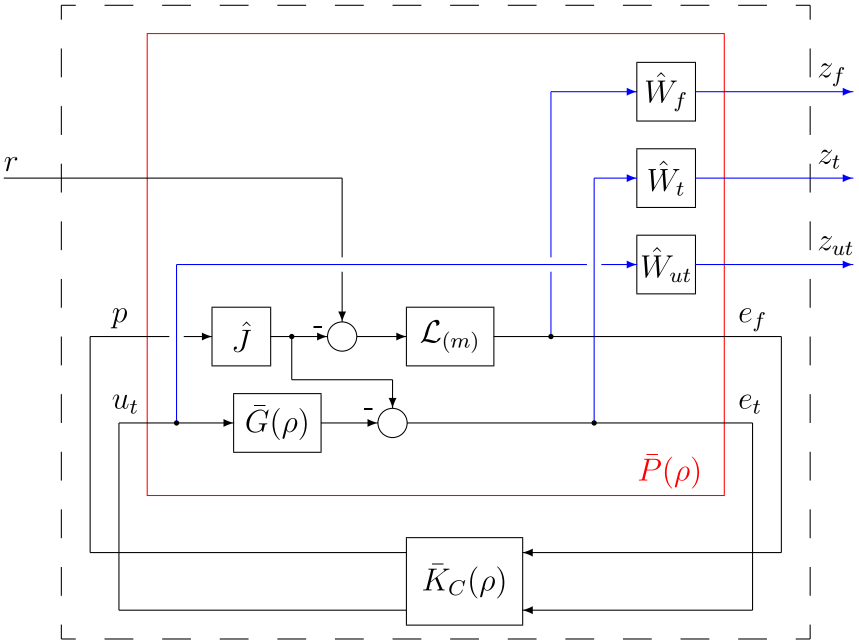

Figure 4.4: Closed loop coupled architecture

In the proposed coupled architecture, the agent model is augmented with different shaping filters, W_f, W_t, W_{ut}, to penalize the formation error, tracking error, and local agent control effort, respectively. A low-pass filter J is used as pre-filter and is chosen to have a sufficiently high bandwidth to avoid direct feedthrough through the generalized plant from its control input to measured output, which would lead to difficulties in the design procedure. For a group of N agents moving in space $\mathbb{R}^m$, each agent is equipped with filters with the following state space representation:

$$\begin{aligned} x_{\alpha_i}(k+1) &= A_\alpha x_{\alpha_i}(k) + B_\alpha u_{\alpha_i} \\ y_{\alpha_i}(k) &= C_\alpha x_{\alpha_i}(k) + D_\alpha u_{\alpha_i} \end{aligned} \tag{4.1}$$

where the subscript α stands for $f,\ t,$ and u_t. The filter J is

$$\begin{aligned} x_{J_i}(k+1) &= A_J x_{J_i}(k) + B_J u_{J_i} \\ y_{J_i}(k) &= C_J x_{J_i}(k) \end{aligned} \tag{4.2}$$

The LPV representation of each agent is of the form

$$\begin{aligned} x_{g_i}(k+1) &= A_g(\rho_i) x_{g_i}(k) + B_g(\rho_i) u_{t_i} \\ y_{g_i}(k) &= C_g x_{g_i}(k) \end{aligned} \tag{4.3}$$

and the LPV output feedback controller has state space realization

$$\begin{aligned} x_{C_i}(k+1) &= A_C(\rho_i) x_{C_i}(k) + B_C(\rho_i) e_i(k) \\ u_i(k) &= C_C(\rho_i) x_{C_i}(k) + D_C(\rho_i) e_i(k) \end{aligned} \tag{4.4}$$

4.3 Coupled-Architecture Stability Analysis

To simplify the stability analysis of the proposed scheme, we ignore the shaping filters and assume that the LPV agents are scheduled homogeneously i.e. $\rho_i = \rho_j$ for all agents i and j. The state space representation of the overall network plant of N agents moving in m-dimensional space is,

$$\begin{aligned} \begin{bmatrix} x_g(k+1) \\ x_J(k+1) \end{bmatrix} &= \begin{bmatrix} \hat{A}_g(\rho) & 0 \\ 0 & \hat{A}_J \end{bmatrix} \begin{bmatrix} x_g(k) \\ x_J(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} r(k) + \begin{bmatrix} 0 & \hat{B}_g(\rho) \\ \hat{B}_J & 0 \end{bmatrix} \begin{bmatrix} p(k) \\ u_t(k) \end{bmatrix} \\ \begin{bmatrix} e_f(k) \\ e_t(k) \end{bmatrix} &= \begin{bmatrix} 0 & -\mathcal{L}_{(m)} \hat{C}_J \\ -\hat{C}_g & \hat{C}_J \end{bmatrix} \begin{bmatrix} x_g(k) \\ x_J(k) \end{bmatrix} + \begin{bmatrix} \mathcal{L}_{(m)} \\ 0 \end{bmatrix} r(k) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p(k) \\ u_t(k) \end{bmatrix} \end{aligned} \tag{4.5}$$

which can be written in compact form,

$$\begin{aligned} x(k+1) &= \hat{A}(\rho) x(k) + \hat{B}_u(\rho) u(k) \\ e(k) &= \breve{C}_e x(k) + \breve{D}_e r(k) \end{aligned} \tag{4.6}$$

The state space representation of the LPV controller $\hat{K}_C(\rho)$ is

$$\begin{aligned} x_C(k+1) &= \hat{A}_C(\rho) x_C(k) + \hat{B}_C(\rho) e(k) \\ u(k) &= \hat{C}_C(\rho) x_C(k) + \hat{D}_C(\rho) e(k) \end{aligned} \tag{4.7}$$

The closed loop system is

$$\begin{bmatrix} x(k+1) \\ x_C(k+1) \end{bmatrix} = \begin{bmatrix} \hat{A}(\rho) + \hat{B}_u(\rho)\hat{D}_C(\rho)\breve{C}_e & \hat{B}_u(\rho)\hat{C}_C(\rho) \\ \hat{B}_C(\rho)\breve{C}_e & \hat{A}_C \end{bmatrix} \begin{bmatrix} x(k) \\ x_C(k) \end{bmatrix} + \begin{bmatrix} \breve{B}_u(\rho)\hat{D}_C(\rho)\breve{D}_e \\ \hat{B}_C(\rho)\breve{D}_e \end{bmatrix} r(k) \tag{4.8}$$

with output equation

$$y = \begin{bmatrix} \hat{C}_g & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_C(k) \end{bmatrix} \tag{4.9}$$

where $x_{cl}(k) = [x(k)^T \; x_C(k)^T]^T \in \mathbb{R}^{n_{x_{cl}}}$, $r(k) \in \mathbb{R}^{n_r}$, and $y \in \mathbb{R}^{n_y}$. By using a suitable state-permuting similarity transformation Π, the closed loop permuted-system can be written in compact form as follows:

$$\begin{aligned} \zeta(k+1) &= A_F(\rho)\zeta(k) + B_F(\rho)r(k) \\ y &= C_F\zeta(k) \end{aligned} \tag{4.10}$$

where $\zeta(k) = \begin{bmatrix} x_1(k)^T & x_{C_1}(k)^T & \dots & x_N(k)^T & x_{C_N}(k)^T \end{bmatrix}^T$,

$$\begin{aligned} A_F(\rho) &= I_N \otimes A_F^d(\rho) + \mathcal{L} \otimes A_F^c(\rho) \\ B_F(\rho) &= I_N \otimes B_F^d(\rho) + \mathcal{L} \otimes B_F^c(\rho) \\ C_F &= I_N \otimes C_F^d + \mathcal{L} \otimes C_F^c \end{aligned} \tag{4.11}$$

and

$$A_F^d(\rho) = \begin{bmatrix} A(\rho) + B_u(\rho)D_C(\rho) \begin{bmatrix} 0 & 0 \\ -C_g & C_J \end{bmatrix} & B_u(\rho)C_C(\rho) \\ B_C(\rho) \begin{bmatrix} 0 & 0 \\ -C_g & C_J \end{bmatrix} & A_C(\rho) \end{bmatrix},$$

$$A_F^c(\rho) = \begin{bmatrix} B_u(\rho)D_C(\rho) \begin{bmatrix} 0 & -C_J \\ 0 & 0 \end{bmatrix} & 0 \\ B_C(\rho) \begin{bmatrix} 0 & -C_J \\ 0 & 0 \end{bmatrix} & 0 \end{bmatrix},$$

$$B_F^d(\rho) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; B_F^c(\rho) = \begin{bmatrix} B_u(\rho)D_C(\rho) \begin{bmatrix} I_m \\ 0 \end{bmatrix} \\ B_C(\rho) \begin{bmatrix} I_m \\ 0 \end{bmatrix} \end{bmatrix},$$

$$C_F^d = \begin{bmatrix} C_g & 0 \end{bmatrix}, \; C_F^c = \begin{bmatrix} 0 & 0 \end{bmatrix},$$

$$A(\rho) = \begin{bmatrix} A_g(\rho) & 0 \\ 0 & A_J \end{bmatrix}, \; B_u(\rho) = \begin{bmatrix} 0 & B_g(\rho) \\ B_J & 0 \end{bmatrix}.$$

Moreover, if $\mathcal{L}$ is diagonalizable and $Z^{-1}\mathcal{L}Z = \Lambda$, where $\Lambda = diag(\lambda_1, \dots, \lambda_N)$, and extending the decoupling transformation of decomposable systems [Massioni and Verhaegen, 2009, Hoffmann et al., 2013] to LPV systems, yields

$$\begin{aligned}\tilde{A}_F(\rho) &= I_N \otimes A_F^d(\rho) + \Lambda \otimes A_F^c(\rho)\\ \tilde{B}_F(\rho) &= \Lambda \otimes B_F^c(\rho)\\ \tilde{C}_F &= I_N \otimes C_F^d\end{aligned} \tag{4.12}$$

where $A_F(\rho) = Z_{(n_{x_{cl}})}\tilde{A}_F(\rho)Z^{-1}_{(n_{x_{cl}})}$, $B_F(\rho) = Z_{(n_{x_{cl}})}\tilde{B}_F(\rho)Z^{-1}_{(n_r)}$, $C_F = Z_{(n_y)}\tilde{C}_F Z^{-1}_{(n_{x_{cl}})}$ substitute in model (4.10),

$$\begin{aligned}Z^{-1}_{(n_{x_{cl}})}\zeta(k+1) &= \tilde{A}_F(\rho)Z^{-1}_{(n_{x_{cl}})}\zeta(k) + \tilde{B}_F(\rho)Z^{-1}_{(n_r)}r(k)\\ Z^{-1}_{(n_y)}y &= \tilde{C}_F Z^{-1}_{(n_{x_{cl}})}\zeta(k).\end{aligned} \tag{4.13}$$

Define new signals, $\tilde{\zeta}(k) = Z^{-1}_{(n_{x_{cl}})}\zeta(k)$, $\tilde{r}(k) = Z^{-1}_{(n_r)}r(k)$, and $\tilde{y}(k) = Z^{-1}_{(n_y)}y(k)$,

$$\begin{aligned}\tilde{\zeta}(k+1) &= \tilde{A}_F(\rho)\tilde{\zeta}(k) + \tilde{B}_F(\rho)\tilde{r}(k)\\ \tilde{y} &= \tilde{C}_F\tilde{\zeta}(k)\end{aligned} \tag{4.14}$$

which is a diagonal system of N modal closed loop subsystems, each parameterized by $\lambda_i,\ i = 1, \dots, N$, which is of the form

$$\begin{aligned}\begin{bmatrix}\tilde{x}_i(k+1)\\ \tilde{x}_{C_i}(k+1)\end{bmatrix} &= \begin{bmatrix} A(\rho) + B_u(\rho)D_C(\rho)\begin{bmatrix}0 & -\lambda_i C_J\\ -C_g & C_J\end{bmatrix} & B_u(\rho)C_C(\rho)\\ B_C(\rho)\begin{bmatrix}0 & -\lambda_i C_J\\ -C_g & C_J\end{bmatrix} & A_C(\rho)\end{bmatrix}\begin{bmatrix}\tilde{x}_i(k)\\ \tilde{x}_{C_i}(k)\end{bmatrix}\\ &+ \begin{bmatrix}\lambda_i B_u(\rho)D_C(\rho)\begin{bmatrix}I_m\\ 0\end{bmatrix}\\ \lambda_i B_C(\rho)\begin{bmatrix}I_m\\ 0\end{bmatrix}\end{bmatrix}\tilde{r}_i(k)\\ \tilde{y}_i &= \begin{bmatrix}C_g & 0\end{bmatrix}\begin{bmatrix}\tilde{x}_i(k)\\ \tilde{x}_{C_i}(k)\end{bmatrix}\end{aligned} \tag{4.15}$$

with $i = 1, \dots, N$. The following corollary summarizes the stability results.

Corollary 4.1. *The LPV controller (4.4) stabilizes the network of LPV agents if it stabilizes the modal subsystems,*

$$\begin{aligned}\tilde{x}_i(k+1) &= A(\rho)\tilde{x}_i(k) + B_u(\rho)u_i(k)\\ e_i(k) &= C_e(\lambda_i)\tilde{x}_i(k) + D_e(\lambda_i)\tilde{r}_i(k)\end{aligned} \tag{4.16}$$

simultaneously for λ_2 *and* λ_N.

For controller design and synthesis, shaping filters are added to the modal subsystem, Figure 4.5. A state space model of the modal subsystem generalized plant is

$$\left[\begin{array}{c} \tilde{x}_{P_i}(k+1) \\ \hline \tilde{x}_{z_i}(k) \\ \hdashline e_{f_i}(k) \\ e_{t_i}(k) \end{array}\right] = \left[\begin{array}{ccccc|c|cc} A_g(\rho) & 0 & 0 & 0 & 0 & 0 & 0 & B_g(\rho) \\ 0 & A_J & 0 & 0 & 0 & 0 & B_J & 0 \\ 0 & -\lambda_i B_f C_J & A_f & 0 & 0 & \lambda_i B_J & 0 & 0 \\ -B_t C_g & B_t C_J & 0 & A_t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{ut} & 0 & 0 & B_{ut} \\ \hline 0 & -\lambda_i D_f C_J & C_f & 0 & 0 & \lambda_i D_f & 0 & 0 \\ -D_t C_g & D_t C_J & 0 & C_t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{ut} & 0 & 0 & D_{ut} \\ \hdashline 0 & -\lambda_i C_J & 0 & 0 & 0 & \lambda_i I_m & 0 & 0 \\ -C_g & C_J & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} \tilde{x}_{P_i}(k) \\ \hline \tilde{r}_i(k) \\ \hline p_i(k) \\ u_{t_i}(k) \end{array}\right] \tag{4.17}$$

which can be written in form,

$$\begin{aligned} \tilde{x}_{P_i}(k+1) &= A_P(\rho,\lambda_i)\tilde{x}_i(k) + B_{P_r}(\lambda_i)\tilde{r}_i(k) + B_{P_u}(\rho)u_i(k) \\ \tilde{z}_i(k) &= C_z(\lambda_i)\tilde{x}_i(k) + D_{zr}(\lambda_i)\tilde{r}_i(k) + D_{zu}u_i(k) \\ v_i(k) &= C_v(\lambda_i)\tilde{x}_i(k) + D_{vr}(\lambda_i)\tilde{r}_i(k) \end{aligned} \tag{4.18}$$

The modal subsystem (4.17) is affine in the uncertain parameter λ_i. The uncertain parameter λ_i, via the relationship $\lambda_i = 1 + \delta_i$, can be replaced by the uncertain parameter δ_i such that $|\delta_i| \leq 1$, based on the properties of the normalized graph Laplacian defined in Chapter 2. This can be used when solving the robust performance problem.

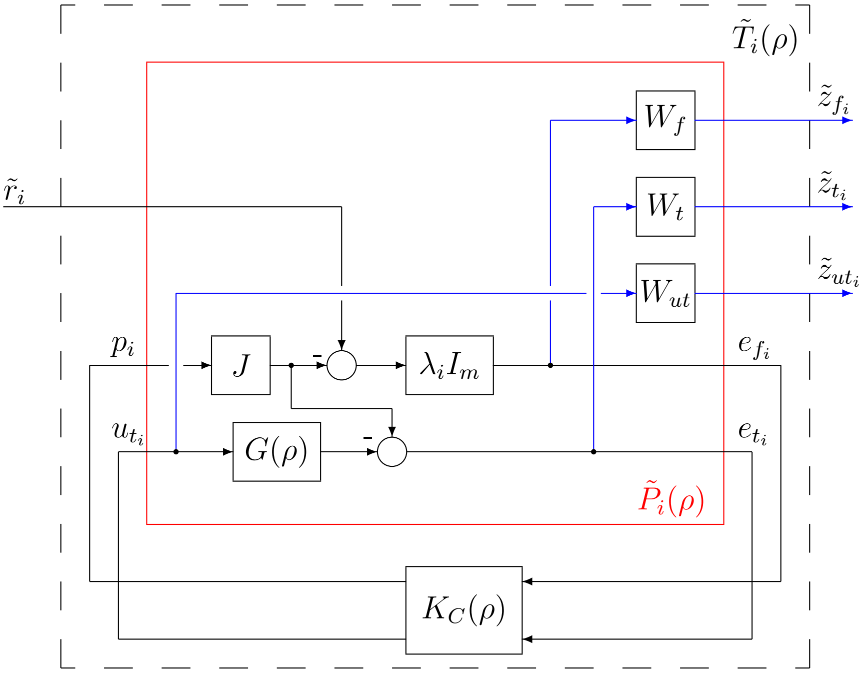

Figure 4.5: Closed loop modal subsystem with performance channel

The following result summarizes the above discussion and can be used to design a distributed controller with guaranteed performance.

Theorem 4.2. *The LPV controller (4.4) stabilizes the network of LPV agents and achieves performance bound* $\frac{1}{\kappa(Z)}\|\tilde{T}\|_{\ell_2} \leq \|T\|_{\ell_2} \leq \kappa(Z)\|\tilde{T}\|_{\ell_2}$ *if it stabilizes the modal subsystem (4.17) with performance* $\|\tilde{T}\|_{\ell_2}$ *simultaneously for* λ_2 *and* λ_N*, where* $\kappa(Z)$ *is the condition number of the diagonalizing transformation matrix* Z*, and* $\tilde{T}$ *is the closed loop modal subsystem.*

Proof. Starting with the whole closed-loop LPV network shown in Fig. 4.4 with shaping filters augmented, we apply a suitable state-permutation matrix Π such that $\Pi^T\Pi = I$, which groups in the closed-loop state vector; the states of each agent and its performance states together with the states of its associated controller. In addition, under the assumption that the graph Laplacian $\mathcal{L}$ is diagonalizable, we obtain a transformed system with block diagonal model matrices. Thus, the transformed network decomposes into N modal subsystems, each of the form

$$\begin{aligned}\begin{bmatrix}\tilde{x}_{P_i}(k+1)\\ \tilde{x}_{C_i}(k+1)\end{bmatrix} &= \begin{bmatrix}A_P(\rho,\lambda_i)+B_{P_u}(\rho)D_C(\rho)C_v(\lambda_i) & B_{P_u}(\rho)C_C(\rho)\\ \\ B_C(\rho)C_v(\lambda_i) & A_C(\rho)\end{bmatrix}\begin{bmatrix}\tilde{x}_{P_i}(k)\\ \tilde{x}_{C_i}(k)\end{bmatrix}\\ &\quad+\begin{bmatrix}B_{P_r}(\lambda_i)+B_{P_u}(\rho)D_C(\rho)D_{vr}(\lambda_i)\\ \\ B_C(\rho)D_{vr}(\lambda_i)\end{bmatrix}\tilde{r}_i(k) \qquad (4.19)\end{aligned}$$

$$\tilde{y}_i = [C_g \quad 0]\begin{bmatrix}\tilde{x}_{P_i}(k)\\ \tilde{x}_{C_i}(k)\end{bmatrix}$$

Thus, the stability of the whole network is equivalent to the stability of each subsystem (4.19). Note that the modal subsystems are identical up to the Laplacian eigenvalues λ_i and that the model matrices depend affinely on λ_i. The modal subsystem performance is obtained by

$$\|\tilde{T}\|_{\ell_2} = \max_i \|\tilde{T}_i\|_{\ell_2}$$

and the original network performance is bounded by

$$\frac{1}{\kappa(Z)}\|\tilde{T}\|_{\ell_2} \leq \|T\|_{\ell_2} \leq \kappa(Z)\|\tilde{T}\|_{\ell_2}$$

because the block-diagonalizing transformation also transforms the input and output signal of the performance channel. □

This theorem shows that one can use mixed-sensitivity loop shaping to tune the distributed formation control scheme for good performance. The synthesis of a controller that can guarantee stability and achieve ℓ_2 performance is shown in [De Caigny et al., 2012]. For tightening the performance bound, one can scale the transformation matrix Z to minimize its condition number, see [Eichler and Werner, 2013].

4.4 Application to swarm of dynamic unicycles

To illustrate the practicality and the benefits of the proposed approach, a leader-follower protocol is considered for a group of $N = 15$ agents in the presence of obstacles and is compared with the separated architecture. Each agent is modeled as discrete time LPV dynamic unicycles (3.11) moving in a plane ($m = 2$). In this scenario, the agents communicate through the directed communication graph, which is shown in Figure 4.6.

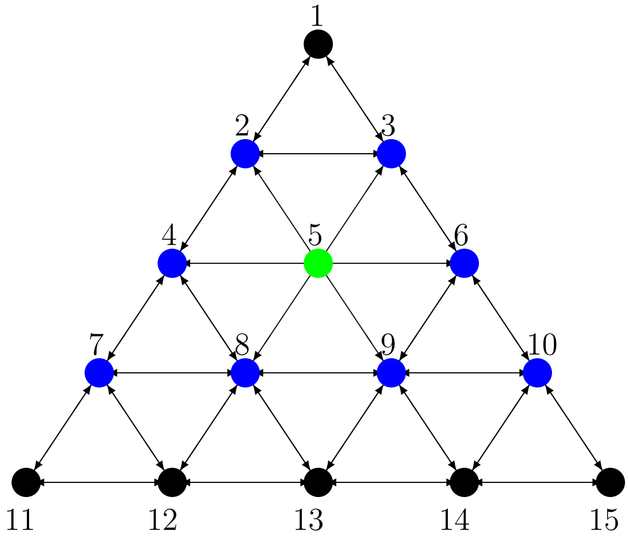

Figure 4.6: Interaction graph

This graph is represented by the normalized graph Laplacian $\mathcal{L}$ with second smallest (Fiedler) eigenvalue $\lambda_2 = 0.0821$, and maximum eigenvalue $\lambda_{max} = 1.5$.

$$\mathcal{L} = \begin{bmatrix}
1 & \frac{-1}{2} & \frac{-1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{-1}{4} & 1 & \frac{-1}{4} & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{-1}{4} & \frac{-1}{4} & 1 & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-1}{4} & 0 & 1 & \frac{-1}{4} & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{-1}{4} & 0 & \frac{-1}{4} & 1 & 0 & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{-1}{4} & 0 & 0 & 1 & \frac{-1}{4} & 0 & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{-1}{6} & \frac{-1}{6} & 0 & \frac{-1}{6} & 1 & \frac{-1}{6} & 0 & 0 & \frac{-1}{6} & \frac{-1}{6} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{-1}{6} & \frac{-1}{6} & 0 & \frac{-1}{6} & 1 & \frac{-1}{6} & 0 & 0 & \frac{-1}{6} & \frac{-1}{6} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{-1}{4} & 0 & 0 & \frac{-1}{4} & 1 & 0 & 0 & 0 & \frac{-1}{4} & \frac{-1}{4} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{2} & 0 & 0 & 0 & 1 & \frac{-1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & \frac{-1}{4} & 1 & \frac{-1}{4} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & \frac{-1}{4} & 1 & \frac{-1}{4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{4} & \frac{-1}{4} & 0 & 0 & \frac{-1}{4} & 1 & \frac{-1}{4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{2} & 0 & 0 & 0 & \frac{-1}{2} & 1
\end{bmatrix} \qquad (4.20)$$

In case of the separate-architecture approach, a robust filter is obtained based on algorithms shown in [Pilz et al., 2011] using the shaping filter

$$W_f = \frac{0.8 - 0.7z^{-1}}{1 - z^{-1}} I_2 \tag{4.21}$$

with performance index $\gamma_F = 1.4509$, and the filter is

$$F = \frac{0.1182 + 1.821 \times 10^{-5} z^{-1}}{1 - z^{-1}} I_2 \tag{4.22}$$

For the agent level, the $A_g(\rho)$ matrix of the *quasi*-LPV dynamic unicycle is parameter dependent, and the dependence is affine in only one parameter $v_t(k)$; in addition, B_g and C_g are parameter independent. Therefore, assumptions in [Ali and Werner, 2011] are satisfied and the LPV controller can be designed for the agent model using dilated LMI's. Shaping filters for sensitivity W_t and control sensitivity W_{ut} are chosen as follows:

$$W_t = \frac{0.001(1 + z^{-1})}{1 - 0.999z^{-1}} I_2, \; W_{ut} = \frac{0.13(1 - z^{-1})}{1 - 0.71z^{-1}} I_2 \tag{4.23}$$

The resulting controller interpolates between the two vertices and the obtained performance index $\gamma_g = 1.3079$. A polytopic discrete-time LPV controller is then synthesized, as shown in Chapter 2. In case of the coupled-architecture approach, a full order discrete time LPV controller is designed based on Algorithm 1 using shaping filters to penalize the formation error, tracking error, and local control effort simultaneously.

$$W_f = \frac{0.002}{1 - z^{-1}} I_2, \; W_t = \frac{0.0021(1 + z^{-1})}{1 - 0.9998z^{-1}} I_2, \; W_{ut} = \frac{0.39(1 - z^{-1})}{1 - 0.6z^{-1}} I_2 \tag{4.24}$$

A filter $J = \frac{0.6321}{1 - 0.3679z^{-1}} I_2$ is used and the performance index of the closed loop modal subsystem is $\gamma_c = 2.82$. This performance is the one achievable for the network controller; the difference from the lower performance of the local agent controller is due to the need for the agents to interact. In both scenarios (see Figure 4.7 and Figure 4.8) the leader is marked in green and its task is to move the swarm followers in a certain path while they are in a certain formation. The leader is considered to be virtual, to have the same dynamics as the followers, and to be equipped with the same local controller as in the separated architecture.

The virtual leader can penetrate obstacles, whereas the followers have to avoid them; this fact is used to illustrate the different behavior of formation control without coupling between the network and agent dynamics on one hand and the formation control as proposed here. The follower agents are assumed to be equipped with sensors to detect obstacles and with a collision avoidance scheme based on a potential field that generates a repulsive force when an agent approaches an obstacle. In both cases, agents start at the same initial positions and attain a formation that is then driven by a leader along a trajectory. This is shown in Figure 4.7 for formation control without coupling and in Figure 4.8 with the proposed coupled-loop.

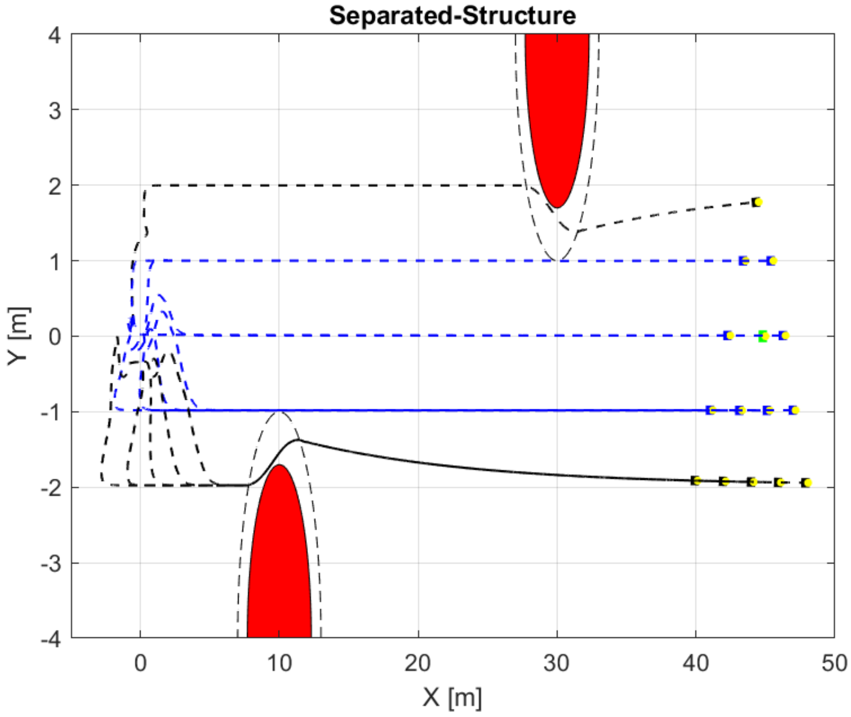

Figure 4.7: Simulation for separated-architecture

The agents achieve consensus in orientation when they are in an obstacle-free area. Along their journey, some followers (the black ones) detect the presence of obstacles. In the case of separated-loop design, Figure 4.7, the blue followers do not react to their neighbors evasive maneuvers when moving near obstacles. In contrast, in the coupled-loop approach, Figure 4.8, the blue followers react to their neighbors' evasive motions. Also, the network stability is robust against the external disturbances where heterogeneous scheduling appears due to differences in lateral velocities for agents subject to obstacles. Otherwise, all agents are scheduled homogeneously.

Comparing the trajectories of the agents for both approaches clearly shows that agents with coupled control design react in a cooperative manner, when obstacles are encountered. In contrast, in the separate-loop design, there is no *real* cooperative behavior. Figure 4.9 shows the ℓ_2-norm of the formation error for both design approaches. It shows that the coupled-architecture design gives better performance especially when agents are avoiding obstacles.

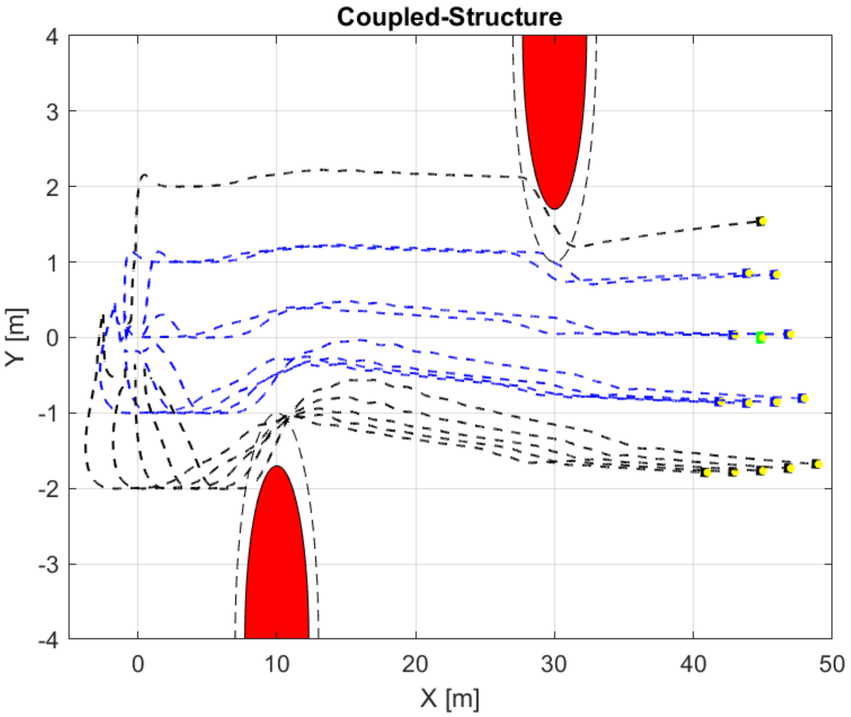

Figure 4.8: Simulation for proposed coupled-architecture

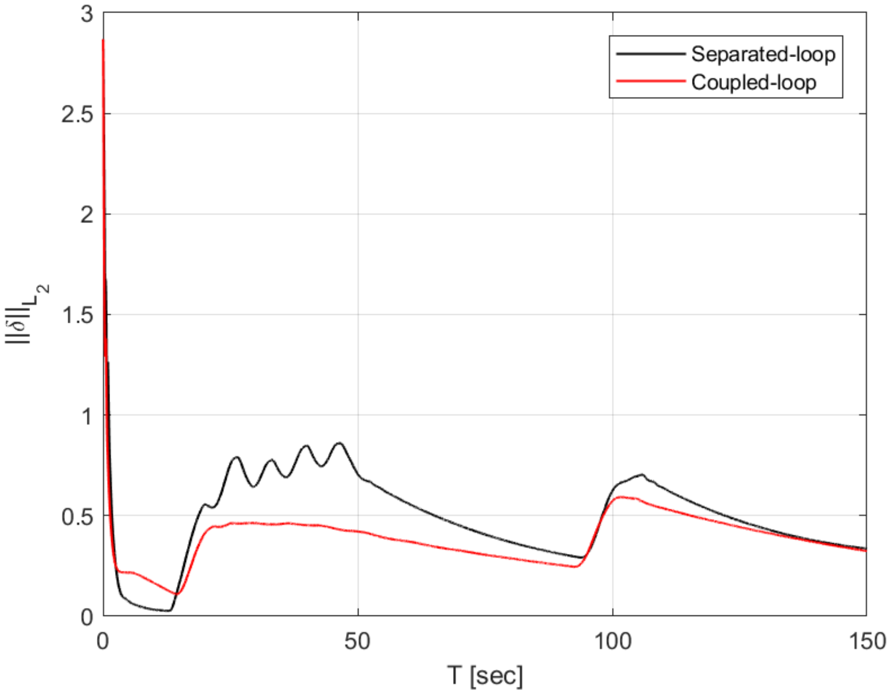

Figure 4.9: ℓ_2-norm of formation error for both approaches

Chapter 5

Flocking Control of LPV Agents

A distributed flocking control scheme is presented in this chapter for a swarm of nonholonomic vehicles modeled as LPV systems. The vehicles are assumed to perform tasks that arise when exploring scalar fields on $\mathbb{R}^m$, which may represent, e.g. concentration levels of a toxic substance in air or water. Complex missions like source seeking, level surface tracking, and level surface monitoring are considered. First, we make the somewhat unrealistic assumption that agents can sense gradients and Hessian information of the scalar field at their current position. However, at the end of this chapter, a gradient/ Hessian estimation scheme is presented to achieve the same missions. The control architecture of each vehicle is structured into two modules: a flocking filter which represents a *virtual layer* of the network, and a local tracking controller for vehicles on the *physical layer.* The flocking filter receives data from neighboring vehicles and generates a reference signal. An LPV tracking controller for the vehicles and a network flocking filter that stabilizes the whole network are designed. Simulated mission scenarios with swarms of autonomous underwater vehicles demonstrate the practicality of the proposed approach.

5.1 Flocking Algorithms in Multi-Agent Systems

In this section, flocking algorithms are reviewed. Consider N agents moving in m-dimensional space (where $m = 2$ or $m = 3$); the dynamics of each agent can be described as

$$\begin{aligned} \dot{q}_i &= p_i \\ \dot{p}_i &= u_i, \qquad i = 1, 2, \ldots, N \end{aligned} \tag{5.1}$$

where $q_i, p_i \in \mathbb{R}^m$ are the position and velocity of an agent i in the flock, respectively, and $u_i \in \mathbb{R}^m$ is the control input for this agent. In addition, we define $q = \begin{bmatrix} q_1^T & q_2^T & \ldots & q_N^T \end{bmatrix}^T$ and $p= \begin{bmatrix} p_1^T & p_2^T & \ldots & p_N^T \end{bmatrix}^T$. Each agent has a limited sensing range r, and the neighbor set of agent i is defined as

$$\mathcal{N}_i = \{j : \|q_i - q_j\| \leq r, j = 1, 2, \ldots, N, j \neq i\}, \tag{5.2}$$

where $\|\cdot\|$ is the Euclidean norm. A flocking protocol is defined as [Olfati-Saber, 2006]

$$u_i = -\sum_{j\in\mathcal{N}_i} \nabla_{q_i}\psi_\alpha(\|q_i - q_j\|_\sigma) + \sum_{j\in\mathcal{N}_i} a_{ij}(q)(p_j - p_i) + cp_i + u_i^\gamma \tag{5.3}$$

The first term defines the cohesion and collision avoidance among agents and is based on a smooth collective potential function $V(q)$ which is the main flocking interaction potential field,

$$V(q) = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \psi_\alpha(\|q_j - q_i\|_\sigma) \tag{5.4}$$

where $\|z\|_\sigma$ is defined as

$$\|z\|_\sigma = \frac{1}{\varepsilon}(\sqrt{1+\varepsilon\|z\|^2} - 1), \qquad \varepsilon > 0 \tag{5.5}$$

and the function $\psi_\alpha : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a non-negative smooth pairwise potential function that reaches its minimum when the distance between the agents i and j reaches the desired value $\|d\|_\sigma$; and $d \leq r$. The flocking interaction potential field works as a repulsive field when the relative distance between the agents i and j is less than $\|d\|_\sigma$ and attractive field when $\|d\|_\sigma \leq \|q_i - q_j\|_\sigma \leq \|r\|_\sigma$, Figure 5.1.

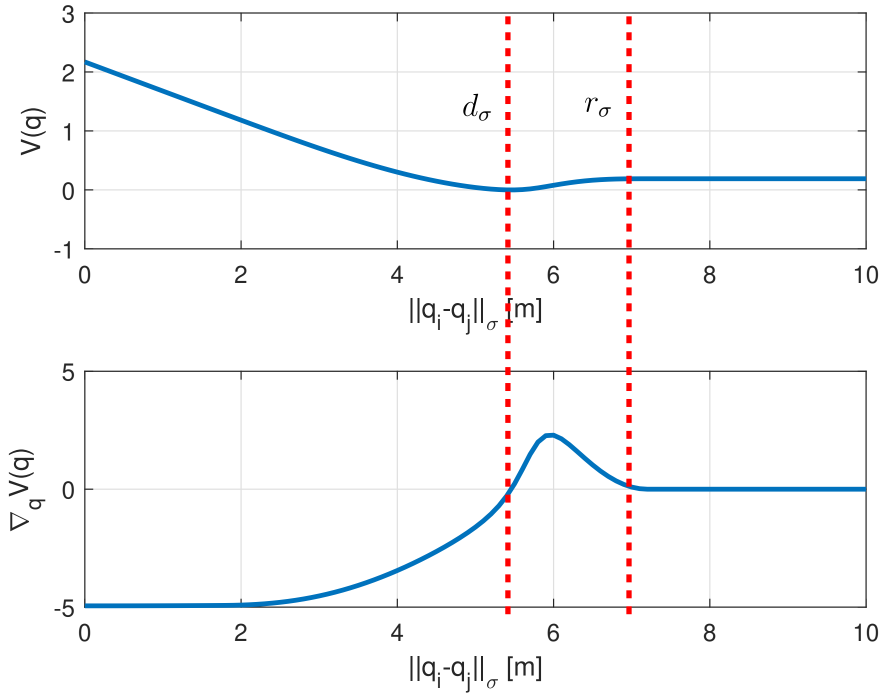

Figure 5.1: Attractive-Repulsive flocking interaction potential field

In some cases, this field is required to be repulsive like in surface tracking and distribution mission, and this is achieved by setting $d = r$, Figure 5.2.

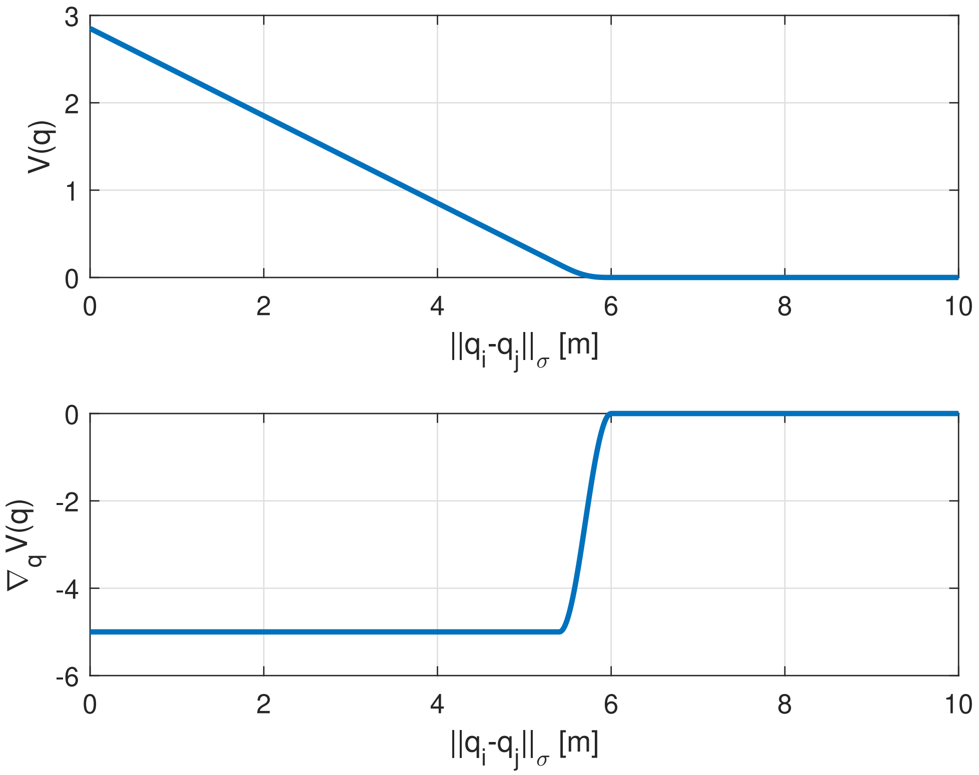

Figure 5.2: Repulsive flocking interaction potential field

The second term in (5.3) represents velocity matching in the flock, where the adjacency matrix $\mathcal{A}(q) = \left[a_{ij}(q)\right]$ is defined by

$$a_{ij}(q) = \begin{cases} 0 & i = j \\ \rho_h\left(\frac{\|q_j - q_i\|_\sigma}{\|r\|_\sigma}\right) & i \neq j \end{cases} \tag{5.6}$$

Here, $\rho_h(z)$ is a bump function with parameter $h \in (0, 1)$; it is defined as

$$\rho_h(z) = \begin{cases} 1 & z \in [0, h) \\ \frac{1}{2}\left(1 + \cos\left(\pi \frac{z-h}{1-h}\right)\right) & z \in [h, 1] \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

Assumption 3. *$\exists c_V > 0$ such that the flocking interaction potential field $V(q) : \mathbb{R}^m \rightarrow \mathbb{R}$ satisfies the Lipschitz condition*

$$\|\nabla V(q) - \nabla V(\tilde{q})\| \leq c_V \|q - \tilde{q}\| \quad \forall q, \tilde{q} \in \mathbb{R}^m$$

The constant c, in (5.3), represents damping, and the last term u^γ is the navigational feedback. The dynamics of the γ-agent represent the group objective, which is defined as

$$\begin{aligned} \dot{q}_r &= p_r \\ \dot{p}_r &= u_r = f(q_r, p_r), \end{aligned} \tag{5.8}$$

where $\begin{bmatrix} q_r^T & p_r^T \end{bmatrix}^T$ is the state vector of the γ-agent. If the γ-agent is static or linear (when a swarm is required to reach a fixed target point or move on a line, respectively), the navigational feedback is [Olfati-Saber, 2006]

$$u_i^\gamma = -c_1(q_i - q_r) - c_2(p_i - p_r), \quad c_1,\ c_2 > 0. \tag{5.9}$$

For nonlinear (dynamic) γ-agents, where the swarm is required to move with time-varying velocity, a navigational feedback control law was proposed [Su et al., 2009b] as

$$u_i^\gamma = -c_1(q_i - q_r) - c_2(p_i - p_r) + f(q_r, p_r), \quad c_1,\ c_2 > 0. \tag{5.10}$$

In both [Olfati-Saber, 2006, Su et al., 2009b], a detailed stability analysis is presented for flocking of double integrator agents with static/dynamic γ-agents.

5.2 γ-Agent Dynamics

As the γ-agent plays the main role in flocking missions, different control laws for γ-agent dynamics are presented according to the required task. The tasks presented in this chapter are *source seeking scenarios*, in which the maximum concentration of a toxic substance has to be found. In addition, concentration tracking for source localization may not required but reaching a certain concentration value is the goal; this is known as *level curve/surface tracking.* Finally, concentration monitoring is performed, where the agents have to move along a level surface with the required velocity; it is known as *level curve/surface monitoring.*

In such missions, the polluted area can be represented by a scalar field $\Psi(q)$: $\mathbb{R}^m \to \mathbb{R}$, which is an unknown smooth scalar function that is twice continuously differentiable and assumed to have an isolated global maximum Ψ_{max} located at $q_s \in \mathbb{R}^m$. In this chapter, the following assumptions are made:

Assumption 4. $\exists c_\Psi > 0$ *such that*

$$\|\nabla\Psi(q) - \nabla\Psi(\tilde{q})\| \leq c_\Psi \|q - \tilde{q}\| \quad \forall q, \tilde{q} \in \mathbb{R}^m$$

Assumption 5. *The scalar field $\Psi(q)$ is time invariant and twice differentiable. Furthermore, there exists a bounded set $\mathcal{B} \subset \mathbb{R}^m$ where $\|\nabla\Psi(q)\| \neq 0$.*

Remark 5.1. *The following problems are converted into a minimization problem by inverting the sign of the scalar field $\Psi(q)$.*

5.2.1 Source Seeking

For the source seeking scenario, the following control law is considered:

$$u_r = -\mu_1 \nabla\Psi(q_r) - \beta_1 \nabla^2\Psi(q_r) p_r, \quad \mu_1,\ \beta_1 > 0. \tag{5.11}$$

The following proposition establishes the stability of the γ-agent dynamics under this control law.

Proposition 5.1. *The γ-agent under control law (5.11) reaches the source of the field $\Psi(q_r)$ from any initial location if* $\nabla^2\Psi(q_r) \succeq 0 \ \forall \ \mu_1, \ \beta_1 > 0$.

Proof. Define the concentration error $e_\psi = \Psi(q_r) - \Psi_{max}$; the dynamics (5.8) is transformed to the equivalent dynamics of

$$\begin{aligned} \dot{e}_\psi &= \nabla\Psi(q_r)^T\dot{q}_r = \nabla e_\psi^T p_r \\ \dot{p}_r &= -\mu_1\nabla e_\psi - \beta_1\nabla^2 e_\psi p_r, \quad \mu_1 > 0, \ \beta_1 > 0 \end{aligned} \tag{5.12}$$

Now, consider the Lyapunov function candidate

$$V(e_\psi, p_r) = \frac{\mu_1}{2}e_\psi^2 + \frac{1}{2}p_r^T p_r \tag{5.13}$$

where $V(e_\psi, p_r) \geq 0$, and $V(e_\psi, p_r) = 0 \iff e_\psi = 0$ and $p_r = 0$. Taking the derivative w.r.t. time yields

$$\begin{aligned} \dot{V}(e_\psi, p_r) &= \mu_1 e_\psi \nabla e_\psi^T \dot{q}_r + p_r^T\dot{p}_r \\ &= \mu_1 e_\psi \nabla e_\psi^T p_r + p_r^T\Big(- \mu_1 e_\psi\nabla e_\psi - \beta_1\nabla^2 e_\psi p_r\Big) \\ &= -\beta_1 p_r^T\nabla^2 e_\psi p_r \leq 0 \impliedby \nabla^2 e_\psi \succeq 0. \end{aligned} \tag{5.14}$$

Asymptotic stability can then be shown using LaSalle's invariance principle [Slotine et al., 1991]. Let Ω be the set of all points where $\dot{V}(e_\psi, p_r) = 0$, which is true $\iff p_r = 0$, i.e.

$$\Omega = \{[e_\psi, p_r^T]^T \in \mathbb{R}^{m+1} | \ \dot{V}(e_\psi, p_r) = 0\} = \{[e_\psi \in \mathbb{R}, p_r^T = 0]^T\}$$

Now, from (5.12), $p_r = 0$ implies that $\dot{p}_r = -\mu_1 e_\psi \nabla e_\psi$, which is nonzero when $e_\psi \neq 0$, i.e., the field maximum is not reached. Hence, it cannot get stuck at a point other than $e_\psi = 0$. Because $[e_\psi, p_r^T]^T = [0, 0]^T$ is an equilibrium point of system (5.12), the largest invariant set $\mathbf{M} \in \Omega$ contains only one point, namely $[e_\psi, e_p^T]^T = [0, 0]^T$. Then, the equilibrium point is asymptotically stable $\forall\mu_1, \ \beta_1 > 0$. □

The γ-agent dynamics to be used for reaching the source Ψ_{max} is then

$$\begin{aligned} \dot{q}_r &= p_r \\ \dot{p}_r &= -\mu_1\nabla\Psi(q_r) - \beta_1\nabla^2\Psi(q_r)p_r \end{aligned} \tag{5.15}$$

5.2.2 Level Curve/Surface Tracking

For level curve/surface tracking, consider the control law for the γ-agent

$$u_r = -\mu_2\big(\Psi(q_r) - \Psi_{ref}\big)\nabla\Psi(q_r) - \beta_2\nabla^2\Psi(q_r)p_r, \quad \mu_2, \ \beta_2 > 0. \tag{5.16}$$

The following proposition establishes the stability of the γ-agent dynamics under this control law.

Proposition 5.2. *The γ-agent under control law (5.16) reaches the level set corresponding to Ψ_{ref} from any initial location if* $\nabla^2\Psi(q_r) \succeq 0 \ \forall \ \mu_2, \ \beta_2 > 0$.

Proof. Define the concentration error $e_\psi = \Psi(q_r) - \Psi_{ref}$; the dynamics (5.8) is transformed to the equivalent dynamics of

$$\begin{aligned} \dot{e}_\psi &= \nabla\Psi(q_r)^T\dot{q}_r = \nabla e_\psi^T p_r \\ \dot{p}_r &= -\mu_2 e_\psi \nabla e_\psi - \beta_2\nabla^2 e_\psi p_r, \quad \mu_2 > 0, \ \beta_2 > 0. \end{aligned} \tag{5.17}$$

Now, consider the Lyapunov function candidate

$$V(e_\psi, p_r) = \frac{\mu_2}{2}e_\psi^2 + \frac{1}{2}p_r^T p_r \tag{5.18}$$

where $V(e_\psi, p_r) \geq 0$ and $V(e_\psi, p_r) = 0 \iff e_\psi = 0$ and $p_r = 0$. Taking the derivative w.r.t. time yields

$$\begin{aligned} \dot{V}(e_\psi, p_r) &= \mu_2 e_\psi \nabla e_\psi^T \dot{q}_r + p_r^T \dot{p}_r \\ &= \mu_2 e_\psi \nabla e_\psi^T p_r + p_r^T\Big(- \mu_2 e_\psi \nabla e_\psi - \beta_2 \nabla^2 e_\psi p_r\Big) \\ &= -\beta_2 p_r^T \nabla^2 e_\psi p_r \leq 0 \impliedby \nabla^2 e_\psi \succeq 0 \end{aligned} \tag{5.19}$$

The asymptotic stability can then be shown using LaSalle's invariance principle [Slotine et al., 1991]. Let Ω be the set of all points where $\dot{V}(e_\psi, p_r) = 0$, which is true iff $p_r = 0$, i.e.

$$\Omega = \{[e_\psi, p_r^T]^T \in \mathbb{R}^{m+1} | \dot{V}(e_\psi, p_r) = 0\} = \{[e_\psi \in \mathbb{R}, p_r^T = 0]^T\}$$

Now, from (5.17), $p_r = 0$ implies that $\dot{p}_r = -\mu_2 e_\psi \nabla e_\psi$, which is nonzero when $e_\psi \neq 0$. Hence, it cannot get stuck at a point other than $e_\psi = 0$. Because $[e_\psi, p_r^T]^T = [0, 0]^T$ is an equilibrium point of system (5.17), the largest invariant set $\mathbf{M} \in \Omega$ contains only one point, namely $[e_\psi, \ e_p^T]^T = [0, 0]^T$. Then, the equilibrium point is globally asymptotically stable. □

The γ-agent dynamics to be used for reaching the concentration level Ψ_{ref} is then

$$\begin{aligned} \dot{q}_r &= p_r \\ \dot{p}_r &= -\mu_2\big(\Psi(q_r) - \Psi_{ref}\big)\nabla\Psi(q_r) - \beta_2\nabla^2\Psi(q_r)p_r \end{aligned} \tag{5.20}$$

5.2.3 Level Curve/Surface Monitoring

For level curve/surface monitoring, consider the control law

$$u_r = -\mu_3\big(\Psi(q_r) - \Psi_{ref}\big)\nabla\Psi(q_r) - \beta_3\big(p_r - p_{ref}\big) + \dot{p}_{ref}, \quad \mu_3, \ \beta_3 > 0 \tag{5.21}$$

where Ψ_{ref} is the specified concentration level, $p_{ref} = \alpha R\frac{\nabla\Psi(q_r)}{\|\nabla\Psi(q_r)\|}$, α is the required speed, and R is an orthonormal matrix such that $R\nabla\Psi(q_r) \perp \nabla\Psi(q_r)$. The following proposition establishes the stability of a γ-agent that describes the motion along the level surface with the given velocity.

Proposition 5.3. *A γ-agent under the control law (5.21) will reach a specified level surface corresponding to Ψ_{ref} starting from any initial locations and moves along it with the required velocity p_{ref} $\forall$ μ_3, $\beta_3 > 0$.*

Proof. Define the error in concentration measurement e_ψ and error in velocity e_p as

$$\begin{aligned} e_\psi &= \Psi(q_r) - \Psi_{ref} \\ e_p &= p_r - p_{ref} \end{aligned} \tag{5.22}$$

The error dynamics are

$$\begin{aligned} \dot{e}_\psi &= \nabla\Psi(q_r)^T\dot{q}_r \\ &= \nabla\Psi(q_r)^T p_r = \nabla\Psi(q_r)^T(e_p + p_{ref}) = \nabla\Psi(q_r)^T e_p + \alpha\nabla\Psi(q_r)^T R\frac{\nabla\Psi(q_r)}{\|\nabla\Psi(q_r)\|} \\ &= \nabla\Psi(q_r)^T e_p = \nabla e_\psi^T e_p \\ \dot{e}_p &= \dot{p}_r - \dot{p}_{ref} = u_r - \dot{p}_{ref} = -\mu_3 e_\psi \nabla e_\psi - \beta_3 e_p, \quad \mu_3 > 0,\ \beta_3 > 0 \end{aligned} \tag{5.23}$$

Now, consider the Lyapunov function candidate

$$V(e) = \frac{\mu_3}{2}e_\psi^2 + \frac{1}{2}e_p^T e_p \tag{5.24}$$

where $e = \begin{bmatrix} e_\psi & e_p^T \end{bmatrix}^T$, $V(e) > 0$ and $V(e) = 0 \iff e = 0$. Taking the derivative of $V(e)$ w.r.t. time yields

$$\begin{aligned} \dot{V}(e) &= \mu_3 e_\psi \dot{e}_\psi + e_p^T\dot{e}_p \\ &= \mu_3 e_\psi \nabla e_\psi^T e_p + e_p^T\big(-\mu_3 e_\psi\nabla e_\psi - \beta_3 e_p\big) \\ &= -\beta_3 e_p^T e_p \leq 0 \end{aligned} \tag{5.25}$$

To use LaSalle's invariance principle to prove the asymptotic stability of (5.23) [Slotine et al., 1991], let Ω be the set of all points where $\dot{V}(e) = 0$ (which is true iff $e_p = 0$), i.e.

$$\Omega = \{e \in \mathbb{R}^{m+1} | \dot{V}(e) = 0\} = \{[e_\psi \in \mathbb{R}, e_p^T = 0]^T\}.$$

Now from (5.23), $e_p = 0$ implies that $\dot{e}_p = -\mu_3 e_\psi \nabla e_\psi$, which is nonzero when $e_\psi \neq 0$. Hence, it cannot get stuck at a point other than $e_\psi = 0$. Because $[e_\psi, e_p^T]^T = [0,0]^T$ is an equilibrium point of system (5.23), the largest invariant set $\mathbf{M} \in \Omega$ contains only one point, namely $[e_\psi, e_p^T]^T = [0,0]^T$. Then, the equilibrium point is globally asymptotically stable. □

The nonlinear dynamics of a γ-agent for moving on a level surface/curve corresponding to Ψ_{ref} with velocity p_{ref} is then

$$\begin{aligned} \dot{q}_r &= p_r \\ \dot{p}_r &= -\mu_3\big(\Psi(q_r) - \Psi_{ref}\big)\nabla\Psi(q_r) - \beta_3\big(p_r - p_{ref}\big) + \dot{p}_{ref} \end{aligned} \tag{5.26}$$

5.3 Flocking of LPV Agents

In this section, a two-layered control architecture for a network of LPV agents is presented, see Figure 5.3. A virtual layer consists of a flocking filter and represents the dynamics of the information flow in the network; a second layer represents the dynamics of physical agents. Both layers are interacting; the flocking filter receives the positions x and velocities v of physical agents and calculates the references p and q, based on the mission to be performed. In this approach, the physical layer can represent the nonholonomic agents modeled as LPV systems. The local tracking controllers for the agents and the flocking filter are designed and analyzed separately. The overall coupled network dynamics can be written as

$$\begin{aligned} \dot{q} &= p \\ \dot{p} &= -\nabla V(x) - (\mathcal{L}_{(m)}(x) + cI)v + u^{\gamma}(x,v) = U(x,v) \\ \dot{z} &= \bar{A}_{cl}(\rho)z + \bar{B}_{cl}(\rho)q, \quad z = [x^T \ v^T]^T \end{aligned} \tag{5.27}$$

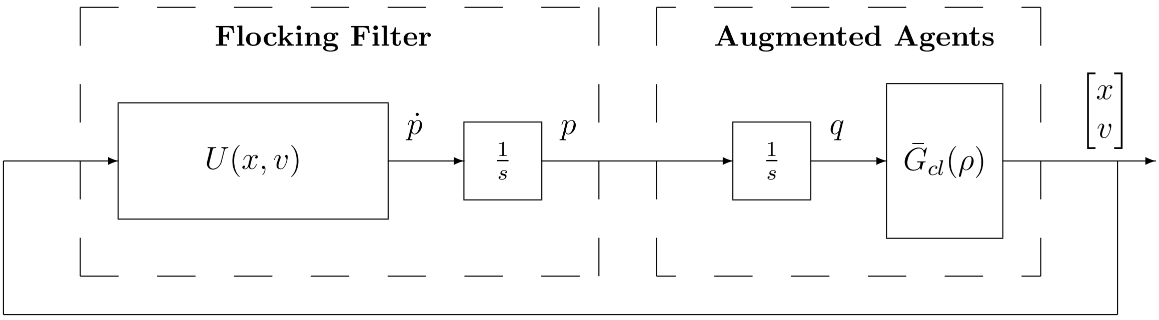

Figure 5.3: Two-layer control architecture loop.

A stability analysis of the proposed architecture was presented in [Attallah et al., 2020], in which stable local closed loop-dynamics is achieved for agents, and the flocking filter is analyzed to guarantee stability in the sense of Lyapunov, i.e., the boundedness of agents' trajectories. The main result is formulated in the following theorem.

Theorem 5.1. *For a network of LPV agents, under Assumptions 3-5 and if the LPV controller* $K(\rho)$ *stablizes the LPV plant* $G(\rho)$ *such that* $\|\mathcal{F}_l\big(G(\rho), K(\rho)\big)\|_{\mathcal{L}_2} \leq \gamma$, *then the flocking filter will provide The bounded references for all initial conditions if*

$$c > \Big((c_V + c_\Psi) + (\|\mathcal{L}_{(m)}\| + c_\Psi + c)\Big)\gamma.$$

Proof. [Datar, 2019, Attallah et al., 2020], see Appendix A □

5.3.1 Simulation Scenarios

Simulation results illustrate the practicality and benefits of the proposed flocking architecture scheme when applied it to a network of realistic AUVs that are equipped with LPV local controllers as shown in Chapter 3. The polluted space is represented by a scalar field $\Psi(q) : \mathbb{R}^3 \to \mathbb{R}$, with its source located at $q_s \in \mathbb{R}^3$.

$$\Psi(q) = \kappa[q^T - q_s^T]^T[q - q_s], \quad \kappa \in \mathbb{R}_+ \tag{5.28}$$

Mission I: Source Seeking Scenario

In the first mission, a swarm of 20 AUVs is considered which is collaborating to find the source of a three dimensional underwater pollution represented by a scalar field located at $q_s = [1\ 8\ 5]^T$ and the parameter $\kappa = 1$. The flocking filter is designed such that the flocking filed $V(q)$ is designed to be attractive-repulsive and reaches its minimum at $\|d\|_\sigma = 1m$, $r = 1.2d$, with parameters $\varepsilon = 0.1$, $h = 0.2$ and $\mu_1 = 1$, $\beta_1 = 0.01$. Agents start from random initial positions and move through the field up to its origin which represents the source, Figure 5.4.

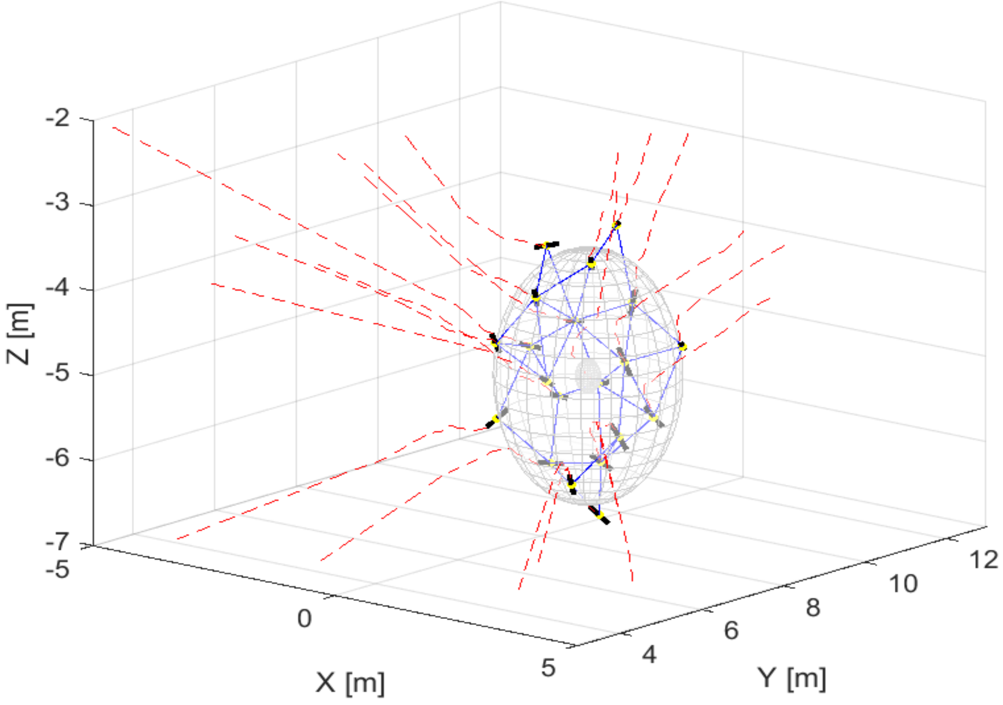

Figure 5.4: 3D source seeking with 20 AUVs. Link

In addition, the evolution of the algebraic connectivity λ_2 of the resulting interaction graph is shown in Figure 5.5.

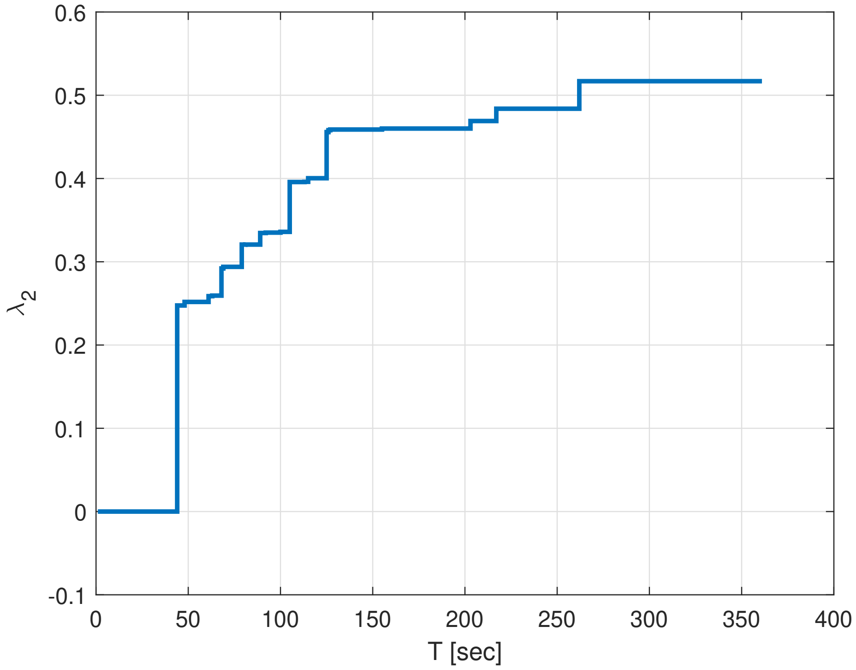

Figure 5.5: Evolution of λ_2 for Laplacian graph of AUVs.

Mission II: Level surface tracking and distribution

In this mission, a group of 20 LPV underwater vehicles with γ-agent dynamics (5.20) will reach the level surface and distribute themselves on it. The field parameters are $\kappa = 0.1,\ q_s = [30\ 40\ 10]^T$ and $\Psi_{ref} = 4.9$. The flocking filter parameters are $\varepsilon = 0.1,\ h = 0.9$, and $c_1 = 15,\ c_2 = 7.75$ and $\mu_2,\ \beta_2$ are chosen to be $8,\ 50$, respectively. Agents start from random initial positions, and because they are equipped with a repulsive field, they can distribute themselves on the surface, Figure 5.6.

In this scenario, the flocking filter is designed such that the flocking interaction field $V(q)$ is selected to be pairwise attractive-repulsive in the first part of the mission with $d = 4m,\ r = 6m$ where they are required to track a certain concentration Ψ_{ref}. Then, it is switched to repulsive field once they reach the concentration surface; thus, they can distribute themselves while staying on the required concentration; this is achieved by setting $d = r = 6m$. Evolution of the algebraic connectivity λ_2 of the resulting interaction graph is shown in Figure 5.7.

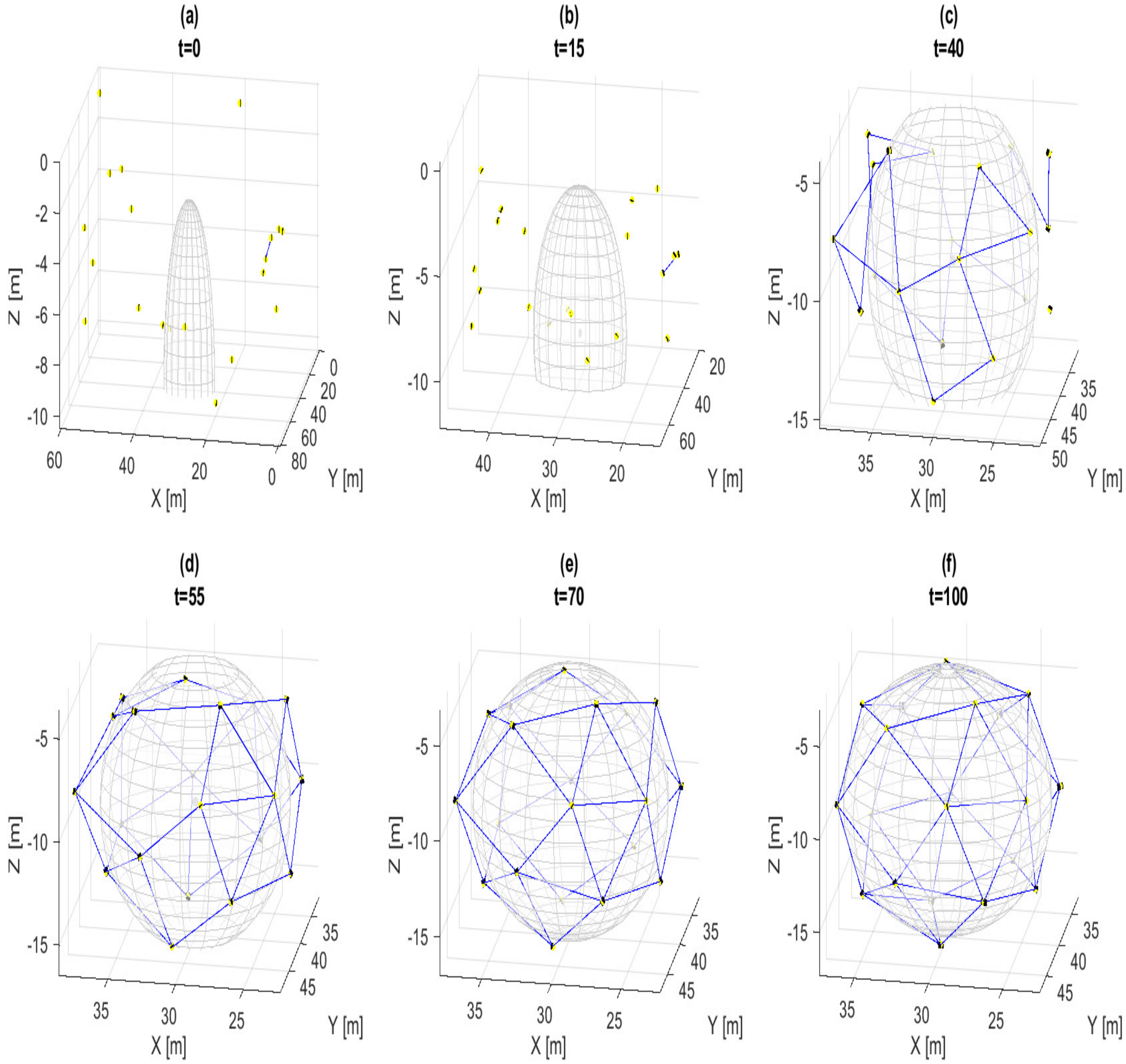

Figure 5.6: Agents distributing themselves on the level surface Link

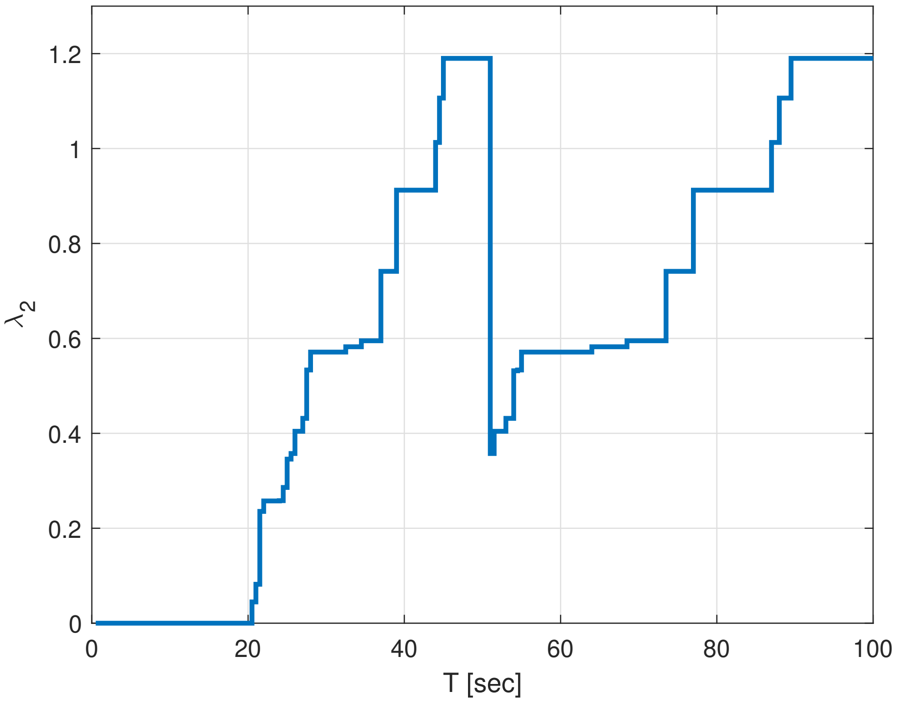

Figure 5.7: Network connectivity λ_2 in the level surface tracking mission

One can observe that as agents start to approach the level surface, new links among them are generated and the network connectivity increases until they reach the required concentration surface. Because of switching from the attraction-repulsion field into a repulsion one, some links are broken which leads to a drop in the network connectivity as shown in Figure 5.7, and agents start to distribute themselves over the surface with newly produced links.

Mission III: Level surface monitoring

In this scenario, a group of 20 LPV underwater vehicles are required to achieve the mission described in γ-agent dynamics (5.26). The scalar field parameters are $\kappa = 0.01$, $q_s = [30\ 30\ 8]^T$ and $\Psi_{ref} = 3.6$. The flocking filter parameters are chosen to be $d = 2\ m$, $r = 1.2d$, $\varepsilon = 0.1$, $h = 0.2$ and $c_1 = 2$, $c_2 = 5$, $\mu_3 = 3$, $\beta_3 = 7$. Agents start from random initial positions; they reach the level surface and move as a flock with required speed $\alpha = 0.65 m/sec$ and with

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

which is in a direction perpendicular to the gradient and parallel to the horizontal plan in the CCW direction, see Figure 5.8. In Figure 5.9, the network connectivity evolution is shown.

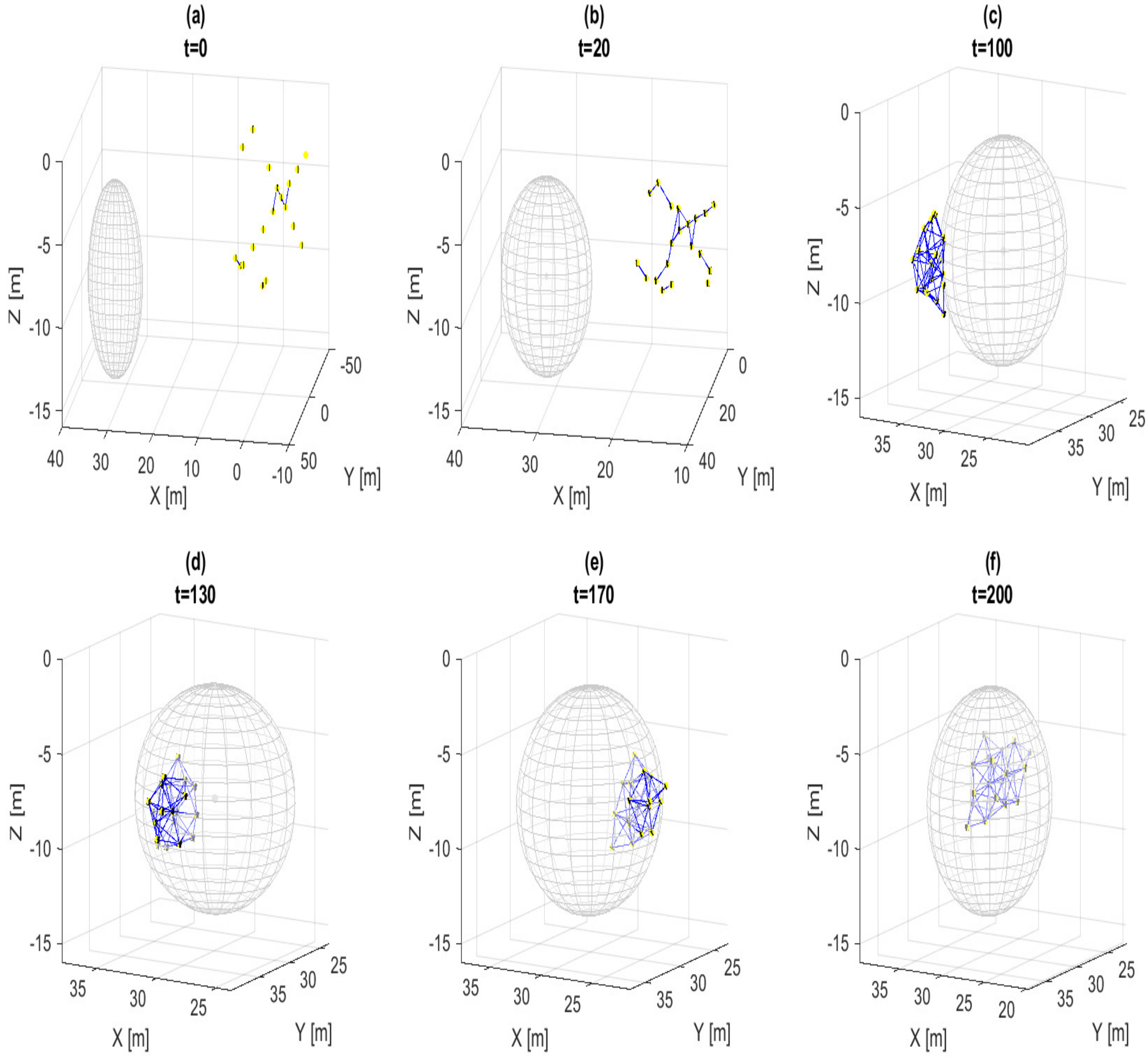

Figure 5.8: Agents flocking along the level surface Link

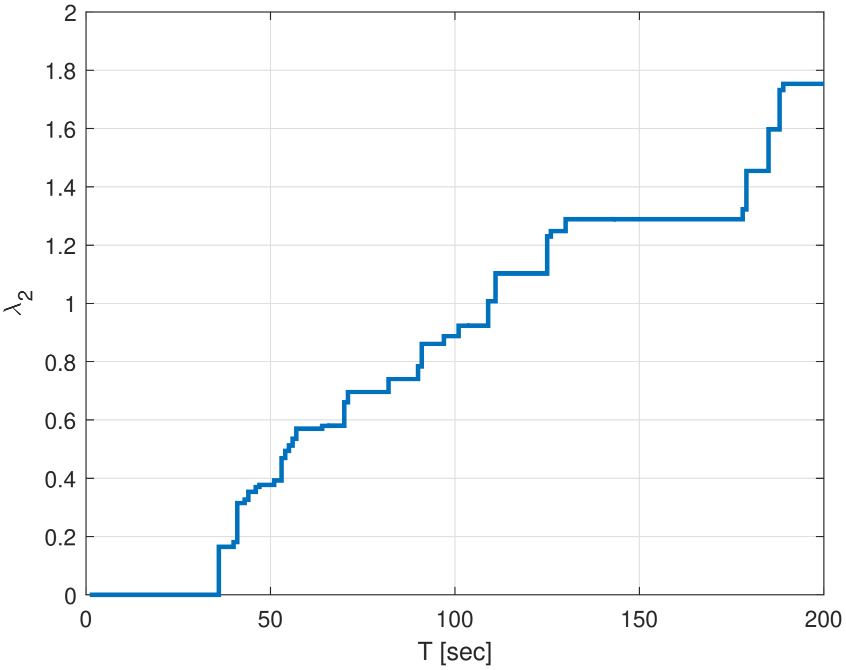

Figure 5.9: Network connectivity λ_2 in the level surface monitoring mission

5.4 Flocking under Gradient and Hessian Estimation

Unlike in the previous section, the field gradient and Hessian are now estimated at each agent's position based on the relative information from its neighborhood. Each agent i in the swarm can measure the field concentration $\Psi(q_i)$ at its position q_i. Because $\Psi(q_i)$ is twice continuously differentiable, it can be approximated by Taylor series:

$$\Psi(q_j){=}\Psi(q_i){+}(q_j{-}q_i)^T\nabla\Psi(q_i){+}\frac{1}{2}(q_j{-}q_i)^T\nabla^2\Psi(q_i)(q_j{-}q_i)+\dots \tag{5.29}$$

where $j \in \mathcal{N}_i$ denotes the neighborhood for agent i. For each agent i, it is possible to compute the relative position between agent i and its neighbors j and by neglecting the higher order terms in (5.29); this can be written in a compact form as

$$\Delta_i = \Phi_i \Theta_i \tag{5.30}$$

where the relative concentration measurement vector $\Delta_i \in \mathbb{R}^{|\mathcal{N}_i|}$ is

$$\Delta_i = \begin{bmatrix} \Psi(q_1) - \Psi(q_i) \\ \vdots \\ \Psi(q_{|\mathcal{N}_i|}) - \Psi(q_i) \end{bmatrix},$$

and if the agents are moving in a plane ($m = 2$), then the data matrix $\Phi_i \in \mathbb{R}^{|\mathcal{N}_i| \times 5}$ is

$$\Phi_i = \begin{bmatrix} q_{x_1} - q_{x_i} & q_{y_1} - q_{y_i} & \frac{1}{2}(q_{x_1} - q_{x_i})^2 & (q_{x_1} - q_{x_i})(q_{y_1} - q_{y_i}) & \frac{1}{2}(q_{y_1} - q_{y_i})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{x_{|\mathcal{N}_i|}} - q_{x_i} & q_{y_{|\mathcal{N}_i|}} - q_{y_i} & \frac{1}{2}(q_{x_{|\mathcal{N}_i|}} - q_{x_i})^2 & (q_{x_{|\mathcal{N}_i|}} - q_{x_i})(q_{y_{|\mathcal{N}_i|}} - q_{y_i}) & \frac{1}{2}(q_{y_{|\mathcal{N}_i|}} - q_{y_i})^2 \end{bmatrix},$$

and the parameter vector $\Theta_i \in \mathbb{R}^5$

$$\Theta_i = \begin{bmatrix} \nabla_x \hat{\Psi}(q_i) & \nabla_y \hat{\Psi}(q_i) & \nabla_{xx} \hat{\Psi}(q_i) & \nabla_{xy} \hat{\Psi}(q_i) & \nabla_{yy} \hat{\Psi}(q_i) \end{bmatrix}^T$$

The estimation problem can be solved by minimizing $\|\Delta_i - \Phi_i \Theta_i\|^2$; then, the vector of the estimated parameters Θ_i can be obtained by

$$\Theta_i = \Phi_i^{\dagger} \Delta_i \tag{5.31}$$

where $\Phi_i^{\dagger} = (\Phi_i^T \Phi_i)^{-1} \Phi_i^T$. To obtain a solution, the matrix Φ_i must be the full column rank (rank(Φ_i)= 5 in $\mathbb{R}^2$ and rank(Φ_i)= 9 in $\mathbb{R}^3$). The data matrix Φ_i is full rank if and only if agent i and its neighbors j are non-collinear; i.e., they do not collapse in a plane ($m = 3$), in a line ($m = 2$), or in a point ($m = 1$). Because $\Phi_i \in \mathbb{R}^{|\mathcal{N}_i| \times 5}$ when moving in a plane, this requires that $|\mathcal{N}_i| \geq 5$ and $\Phi_i \in \mathbb{R}^{|\mathcal{N}_i| \times 9}$ when agents moving in three dimensional space, then $|\mathcal{N}_i| \geq 9$. The minimum number of agents to estimate the field gradient and Hessian is 3 agents ($m = 1$), 6 agents ($m = 2$), and 10 agents ($m = 3$). Another efficient approach to obtain a solution for the over-determined system (5.30), based on the previous discussion, by the singular value decomposition of data matrix $\Phi_i = U_i \Sigma_i V_i^T$ and computing the parameter vector is as follows [Strang, 2006]:

$$\Theta_i = V_i \Sigma_i^{\dagger} U_i^T \Delta_i, \quad \Sigma_i^{\dagger} = (\Sigma_i^T \Sigma_i)^{-1} \Sigma_i^T \tag{5.32}$$

The following algorithm summarizes the flocking process:

```
Algorithm 2: Flocking under gradient and Hessian estimation in $\mathbb{R}^m$
  Step 1: For each agent-$i$, $i = 1, \dots, N$
            measure and transmit $x_i,\ v_i,\ \Psi(x_i)$
            collect neighbors $j \in \mathcal{N}_i$ data $x_j,\ v_j,\ \Psi(x_j)$

  Step 2: if $|\mathcal{N}_i| \geq N_{min}$ calculate $\Theta_i$ based on (5.31) or (5.32), else set $\Theta_i = 0$

  Step 3: Given $\nabla \hat{\Psi}(x_i),\ \nabla^2 \hat{\Psi}(x_i)$, calculate $u_i^{\gamma}$ based on required mission
```

When the number of agents is not sufficient to estimate both the gradient and Hessian, the problem is simplified to estimate only the field gradient, and the Hessian matrix in the control law can be replaced by identity. The system of equations that collect all data described in (5.29), after assuming nonlinear terms are bounded and can be neglected, can be written in compact form as

$$\bar{\Delta}_i = \bar{\Phi}_i \bar{\Theta}_i \tag{5.33}$$

where $\bar{\Delta}_i \in \mathbb{R}^{|\mathcal{N}_i|}$, $\bar{\Phi}_i \in \mathbb{R}^{|\mathcal{N}_i|\times m}$ and $\bar{\Theta}_i \in \mathbb{R}^m$. The solution $\|\bar{\Delta}_i - \bar{\Phi}_i\bar{\Theta}_i\|^2$ is given by

$$\bar{\Theta}_i = \bar{\Phi}_i^{\dagger}\bar{\Delta}_i \tag{5.34}$$

where $\bar{\Phi}_i^{\dagger} = (\bar{\Phi}_i^T\bar{\Phi}_i)^{-1}\bar{\Phi}_i^T$. The inverse of $(\bar{\Phi}_i^T\bar{\Phi}_i)$ exists if the column rank of $\bar{\Phi}_i = m$ which requires $|\mathcal{N}_i| \geq m$ [Rosero García, 2017]. In addition, the over-determined system (5.33) can be solved using the singular value decomposition approach:

$$\bar{\Theta}_i = \bar{V}_i\bar{\Sigma}_i^{\dagger}\bar{U}_i^T\bar{\Delta}_i, \quad \bar{\Sigma}_i^{\dagger} = (\bar{\Sigma}_i^T\bar{\Sigma}_i)^{-1}\bar{\Sigma}_i^T \tag{5.35}$$

```
Algorithm 3: Flocking under gradient estimation in $\mathbb{R}^m$
  Step 1: For each agent-$i$, $i = 1, \ldots, N$
            measure and transmit $x_i,\ v_i,\ \Psi(x_i)$
            collect neighbors $j \in \mathcal{N}_i$ data $x_j,\ v_j,\ \Psi(x_j)$
  Step 2: if $|\mathcal{N}_i| \geq m$, calculate $\bar{\Theta}_i$ based on (5.34) or (5.35), else set $\bar{\Theta}_i = 0$
  Step 3: Given $\nabla\hat{\Psi}(x_i)$, calculate $u_i^{\gamma}$ based on required mission
```

5.4.1 Simulation Scenarios

The potential of cooperative control is illustrated in the following missions, as with an abundance of available information and sensors for each agent, agents can exchange their information to achieve a target in a certain mission. In particular, in source seeking and level curve monitoring, agents exchange the concentration measurements and their states to estimate the gradient and the Hessian of the scalar field to achieve the corresponding target. In contrast, in case of no exchange of data, each agent has to collect lots of measurements to calculate the required data, and then, go which is time consuming and less efficient, and leads to the requirement of more power and sensors. In each of the following missions, a group of $N = 10$ LPV dynamic unicycles described in (3.10) is cooperating in a polluted area while applying flocking rules.

Mission IV: source seeking

In such missions, agents are required to find the origin (maximum) of the unknown field $\Psi(q)$ and each agent is equipped with a single sensor to measure the field concentration at its center of gravity; its position and velocity are available. The pollution is descried using a scalar field $\Psi(q) : \mathbb{R}^2 \to \mathbb{R}$,

$$\Psi(q) = 10e^{-0.1\left((\frac{q_x-50}{20})^2+(\frac{q_y-50}{20})^2\right)} \tag{5.36}$$

with the source located at $q_s = [50\ 50]^T$. First, the flocking control loop under gradient estimation only (Algorithm 3) and the scalar field, which is noise-free, is applied. A simulation of the source seeking scenario is shown in Figure 5.10.

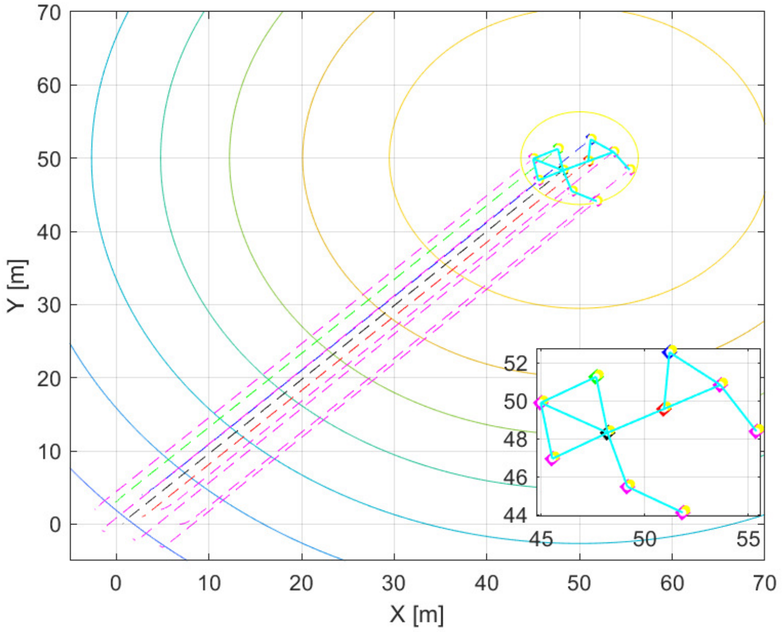

Figure 5.10: Source seeking flocking with LPV agents.

Figure 5.11 shows the velocity of each agent starting from zero initial velocities, and the agents move with the same velocity during the journey. When they reach the source, the velocities become zero.

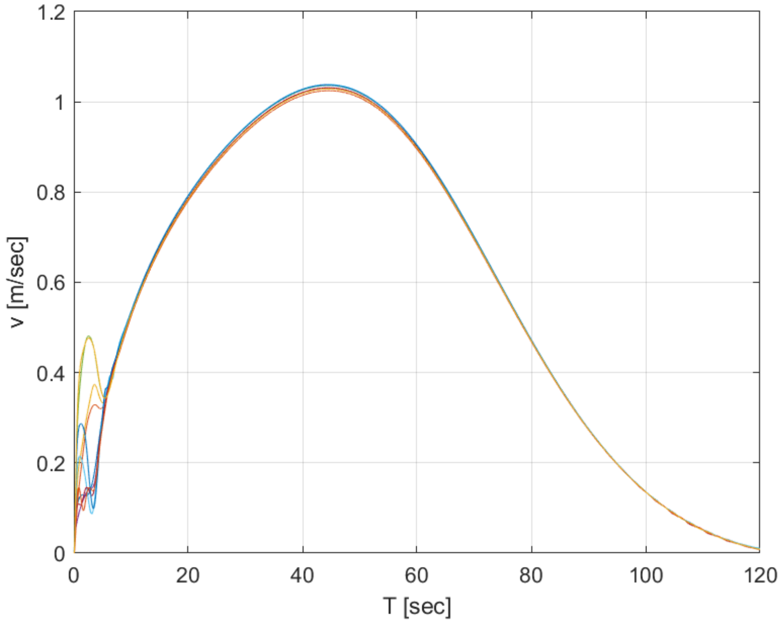

Figure 5.11: Agents' velocities during the mission.

Figure 5.12 shows a comparison between the estimated gradient for an arbitrary agent and true gradient, which indicates that the estimated gradient approximation leads to a very similar values to the true gradient.

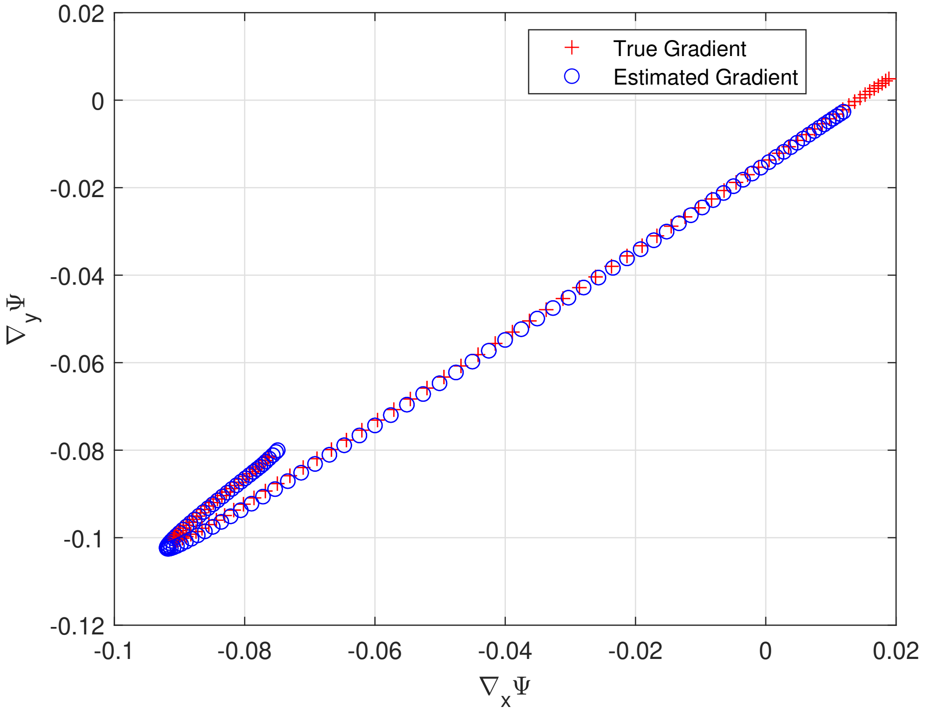

Figure 5.12: $\nabla_x \Psi$ vs. $\nabla_y \Psi$ components' estimated gradient for an arbitrary agent

Another scenario, where when the polluted area has multiple sources and is represented by a non-convex scalar field

$$\Psi(q) = 9e^{-0.01(q_x+3)^2-0.01(q_y-12)^2} + 9e^{-0.02(q_x-17)^2-0.02(q_y-5)^2} \tag{5.37}$$

with sources located at $q_{s_1} = [-3\ 12]^T$ and $q_{s_2} = [17\ 5]^T$. Agents are located in small groups in random places. They are able to locate the sources according to their initial positions, Figure 5.13. The velocity plot is shown in Figure 5.14. The spikes in velocities reflect the combination of subgroups and changing directions while starting the mission, but all velocities converge to zero once the agents find the sources.

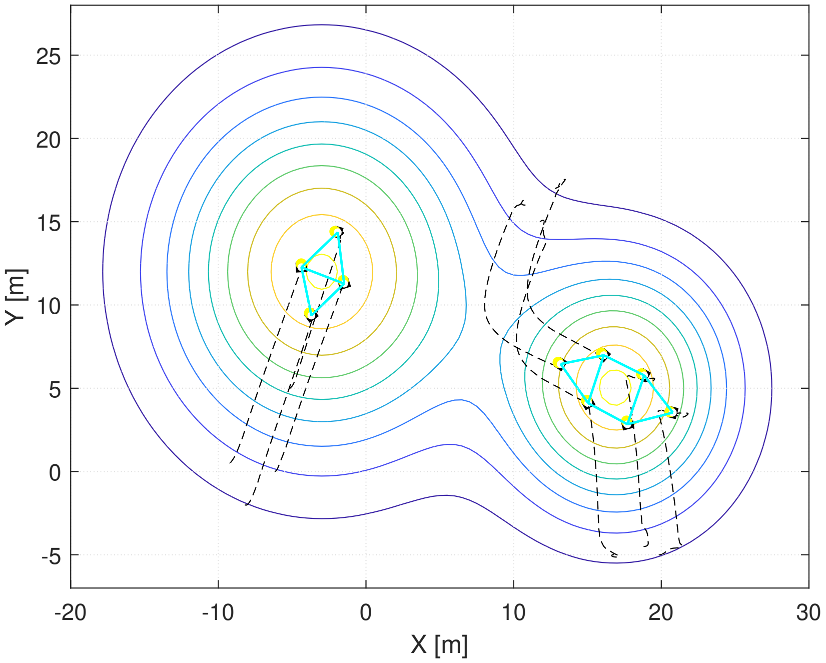

Figure 5.13: Multiple source seeking flocking with LPV agents.

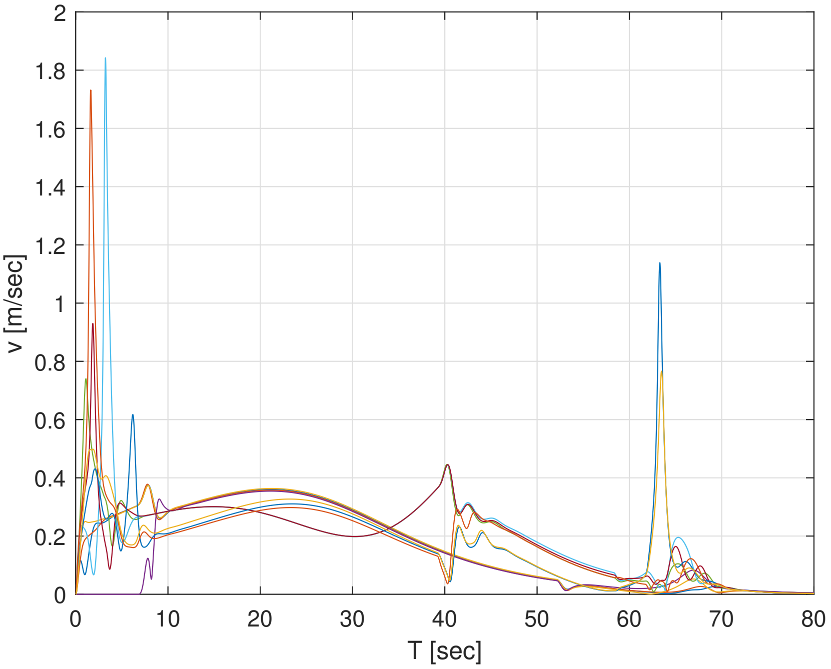

Figure 5.14: Velocities during the journey.

To test the performance of the estimation algorithm, the scalar field is corrupted by spatial white Gaussian noise with zero mean and variance 1. The signal to noise ratio is $20dB$ in the case of a single source and $5dB$ for multiple sources. Agents start from the same initial conditions and can find the source(s), Figure 5.15, Figure 5.17. The velocity curves are shown in Figure 5.16 and Figure 5.18

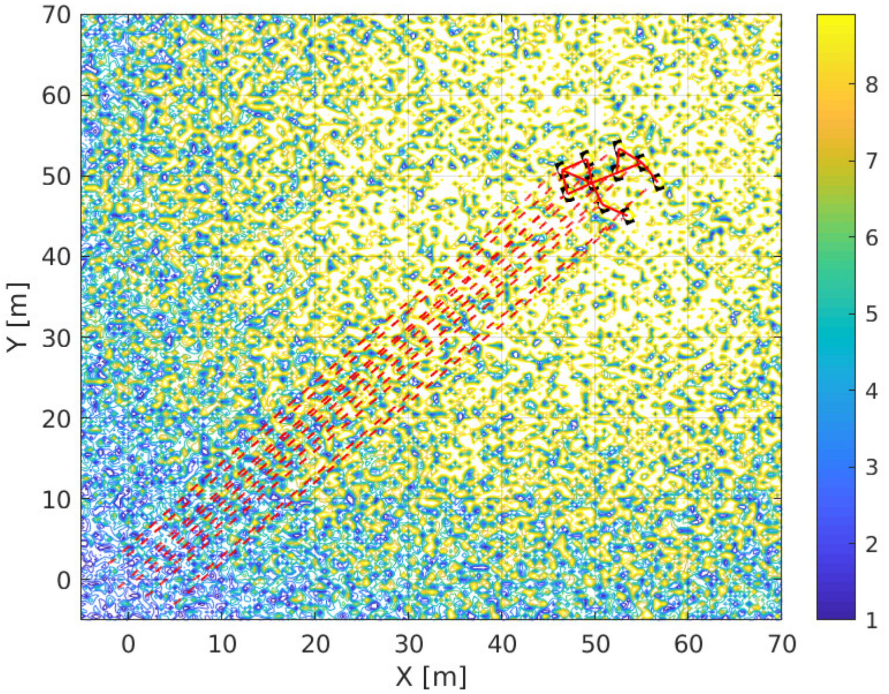

Figure 5.15: Source seeking flocking with LPV agents.

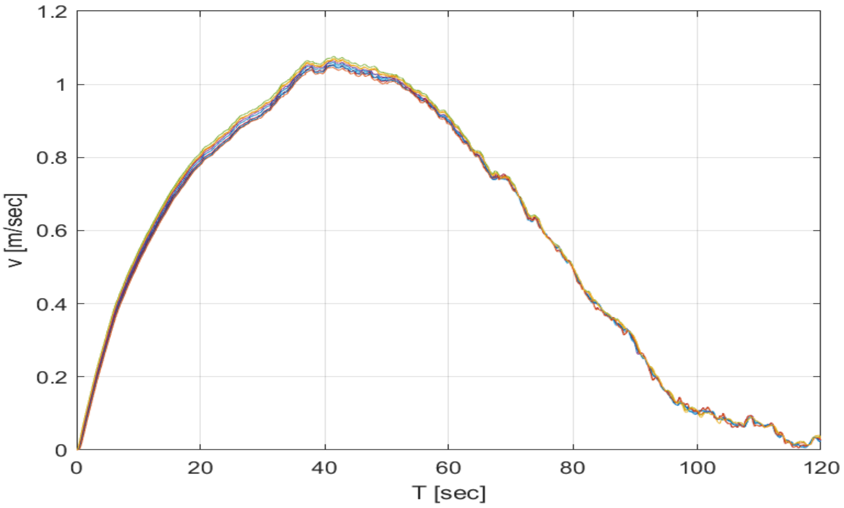

Figure 5.16: Velocities during the journey.

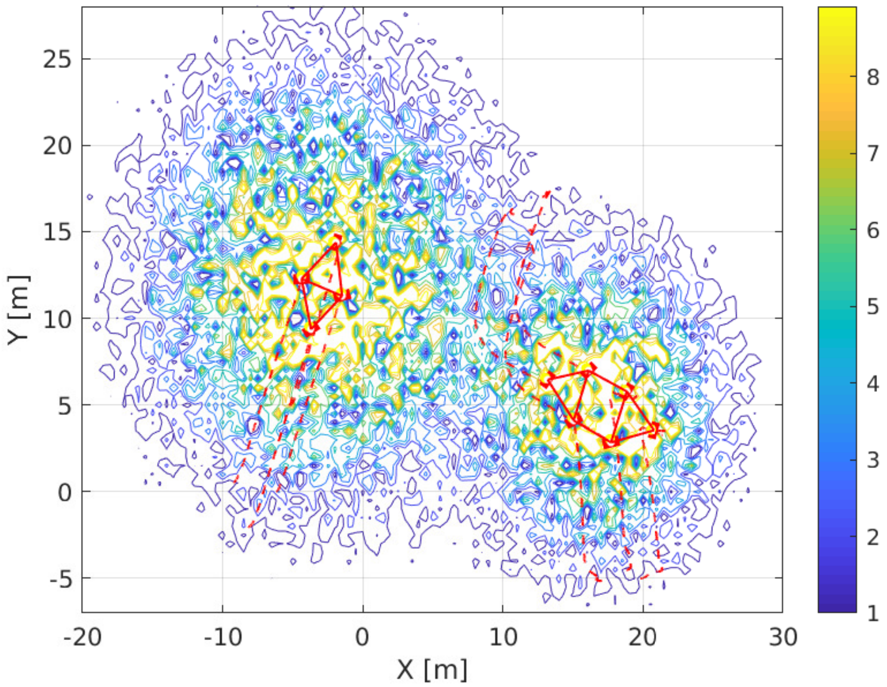

Figure 5.17: Multiple source seeking flocking with LPV agents.

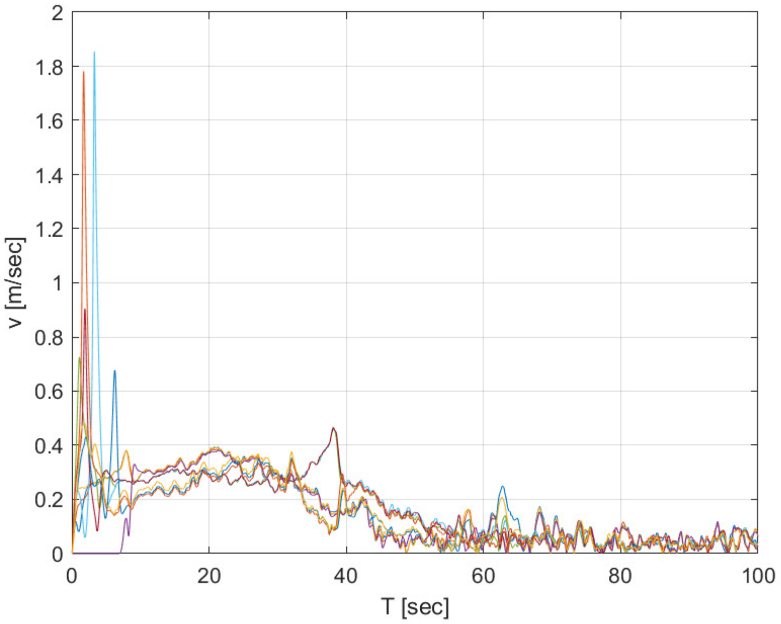

Figure 5.18: Velocities during the journey.

Mission V: level curve monitoring

In this scenario, agents are not required to access the source location. However, they should move along a certain concentration value which is represented by the level curve $\Psi(q) = \Psi_{ref}$ and can move with speed α along the curve. In this case, agents apply (Algorithm 2) to estimate both the field gradient and Hessian. To get a better estimation scheme, each agent must receive at least five different measurements to estimate both the gradient and Hessian. Instead of having at least five neighbors for agent-i in such a case, an alternative is that each agent has two sensors, one on the tip and one at the tail. Thus, this reduces the five neighbors to only two and the available measurements are six measurements to estimate the agent-i's center of the gradient and Hessian of the gravity field. A simulation of the flock of 10 LPV dynamic unicycles, moving along level curve $\Psi_{ref} = 9$ with required speed $\alpha = 0.7\ m/sec$, is shown in Figure 5.19, while the agents' velocities during the journey are shown in Figure 5.20

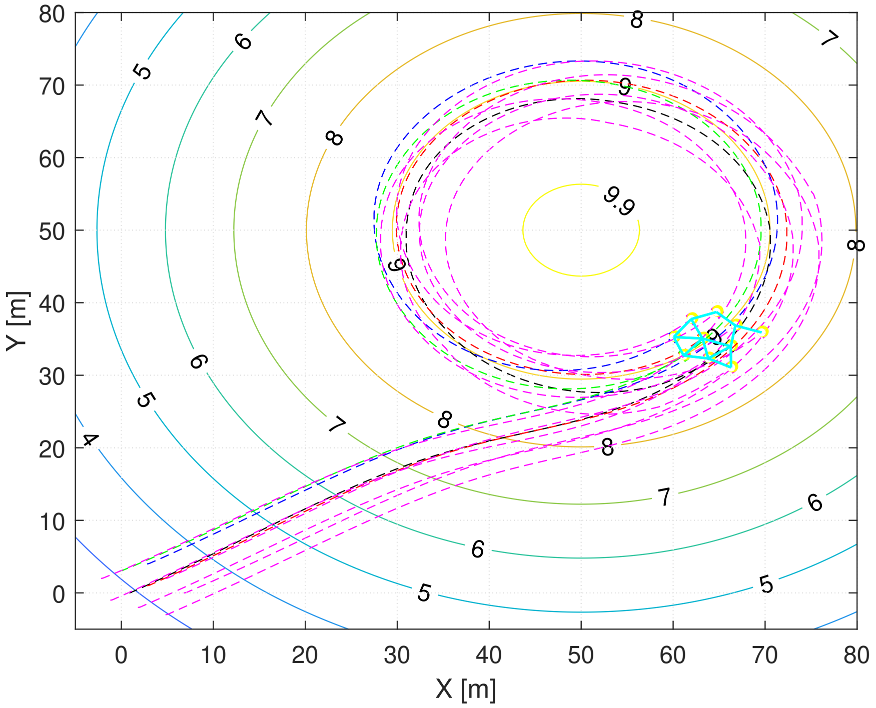

Figure 5.19: Level curve tracking with a flock of LPV agents.

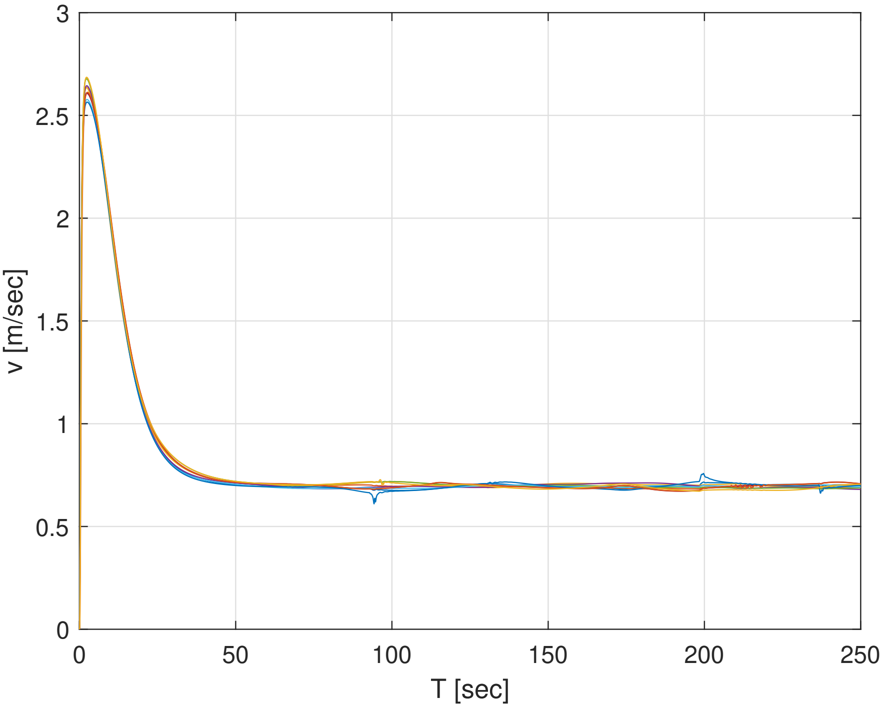

Figure 5.20: Velocities during level curve tracking mission.

5.5 Flocking Loop Test Experiment

This section shows experimental results obtained in the scope of the flocking loop algorithm. The goal of this work is to validate the proposed coupled flocking loop with application of source seeking and level curve monitoring scenarios experimentally. Agents have their stabilizing controllers, and the flocking loop itself is tested with application scenarios.

5.5.1 Experiment Platform

The platform considered in this work is based on multiple quadcopters as mobile agents, which collaborate to achieve a certain mission. The *crazyflie 2.1* is an open-source experimental platform of a nano-quadcopter from *Bitcraze*, allowing modifications to the firmware of both microcontrollers of its double-MCU architecture, an nRF51 radio chip, and an STM32 which runs the main firmware, Figure 5.21[1].

The platform uses an indoor positioning system based on the Decawave DWM1000 chip called Loco-Positioning System (LPS), developed by *Bitcraze*. LPS is a radio-based system consisting of multiple stationary radio modules with known positions (anchors), and a mobile radio module (tag) attached to each mobile robot. At each corner of a $3m \times$

---

[1] More information, see `https://wiki.bitcraze.io` and `https://www.bitcraze.io`

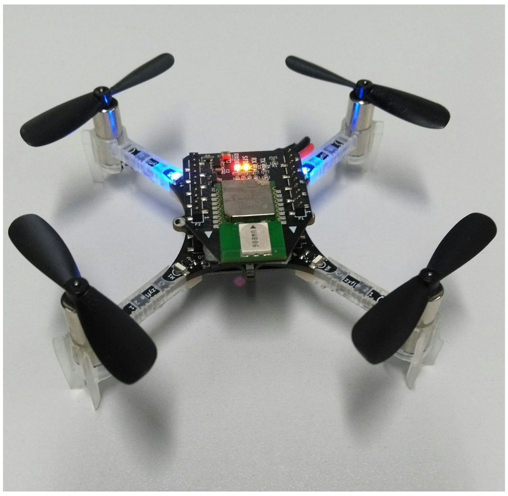

Figure 5.21: Crazyflie 2.1.

$4m$ area, two Loco positioning anchors are placed at different heights. The anchors are configured with their positions. To determine the positions, each crazyflie uses the method of Time Difference Of Arrival (TDOA), according to which, the tag uses the time difference between the arrival of packet from a master anchor and those of other anchors. As a result, for each anchor, the difference of the distances between this anchor and the master anchor can be computed, which results in a hyperbola of possible tag locations. The tag location is determined as the intersection of the hyperbolas determined for different anchors. Because the anchors are configured to send synchronous messages, and are based on the different times at which these messages are received, the position can be locally estimated on the crayzflie.

The firmware was modified to allow peer-to-peer communication, and each crazyflie broadcasts its message along with an ID that identifies the sender uniquely and a time stamp. A heuristic approach of adjusting to the slowest clock is used to address the problem of clock drift [Paulsen, 2018]. Data for logging purposes are broadcast, which are logged by a computer. The computer serves only to send a start command and then passively log the data for later analysis [Paulsen, 2019, Datar et al., 2020].

5.5.2 Flocking Control Experiments

For mission scenarios, we used 3 quad-copters; the scalar field $\Psi(q)$ has the source location at $q_s = [2\ 1.5]^T$ and is represented by

$$\Psi(q) = (q_x - 2)^2 + 2(q_y - 1.5)^2$$

It is stored on-board but only queried to get the field strength at the current location. The necessary information of position and velocity is communicated to establish a flock and the field strength at the current position to compute the navigational feedback term u_i^γ. In each scenario, the solid dot ($\bullet$) represents the initial position, while the symbol ($\times$) shows the position end.

Source Seeking Experiment

The quad-copters are located initially at $[3.1\ 2]^T$, $[1.05\ 2.4]^T$ and $[2.65\ 1.22]^T$. For the first 20 sec, they are turned on and achieve an altitude of 0.5 m; then, they are turned into the coupled flocking loop, where the planar source seeking mission is achieved. Figure 5.22 shows the path of quad-copters during the mission. At the source, agents have an offset distance of approximately 0.35 m due to the flocking rules. The velocity components of each quad-copter are shown in Figure 5.23.

Level Curve Monitoring Experiment

Another scenario of level curve monitoring is tested in a second experiment. Quad-rotors are placed at $[2.6\ 2.4]^T$, $[3.6\ 0.15]^T$ and $[0.25\ 0.75]^T$. They are required to find the level curve of $\Psi_{ref} = 1$, and they are required to move along it with speed of $\alpha = 0.4\ m/sec$. Figure 5.24 shows the path of the quad-copters, and the velocity components of each quad-copter are shown in Figure 5.25.

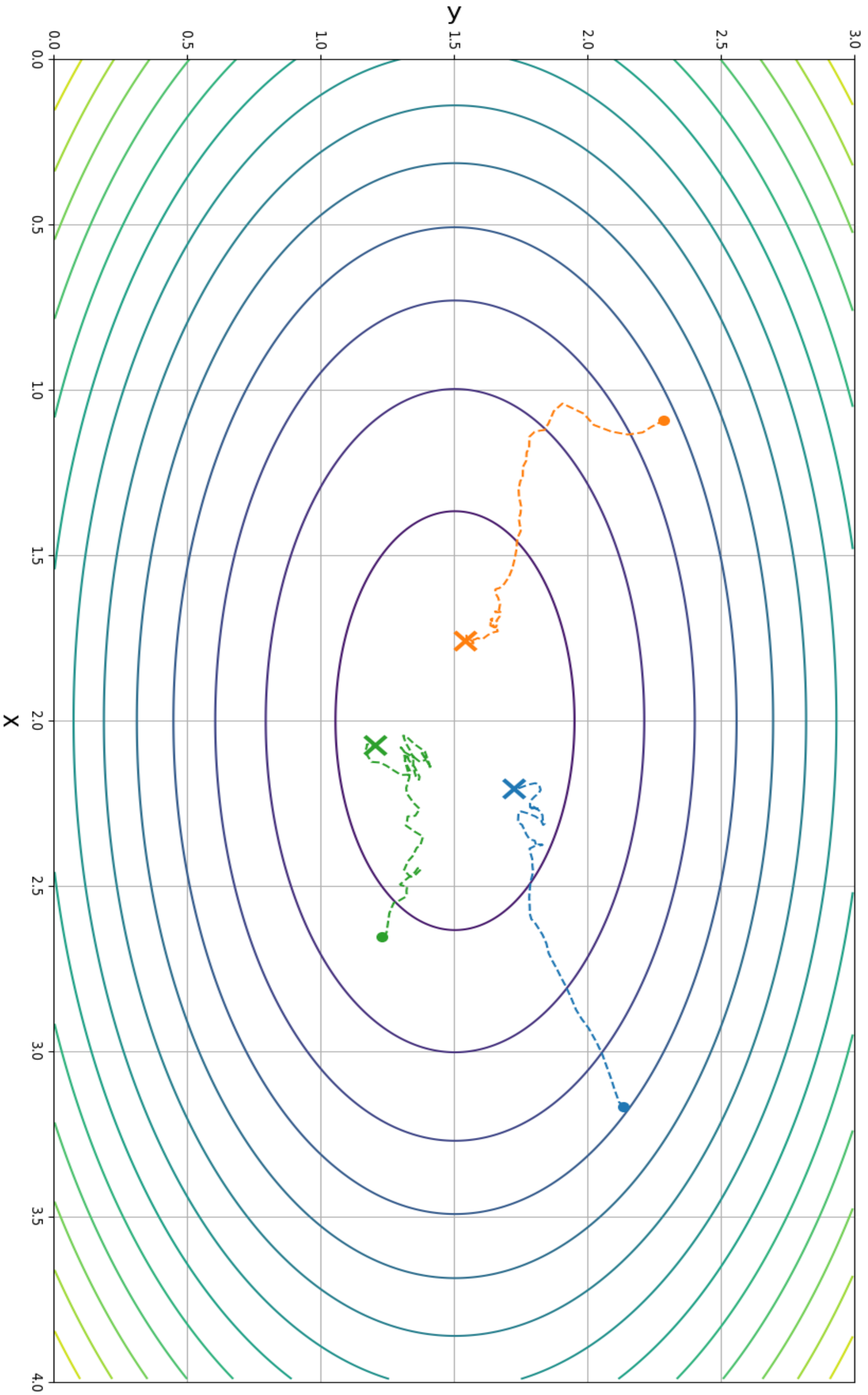

Figure 5.22: Source seeking with 3 Quad-copters.

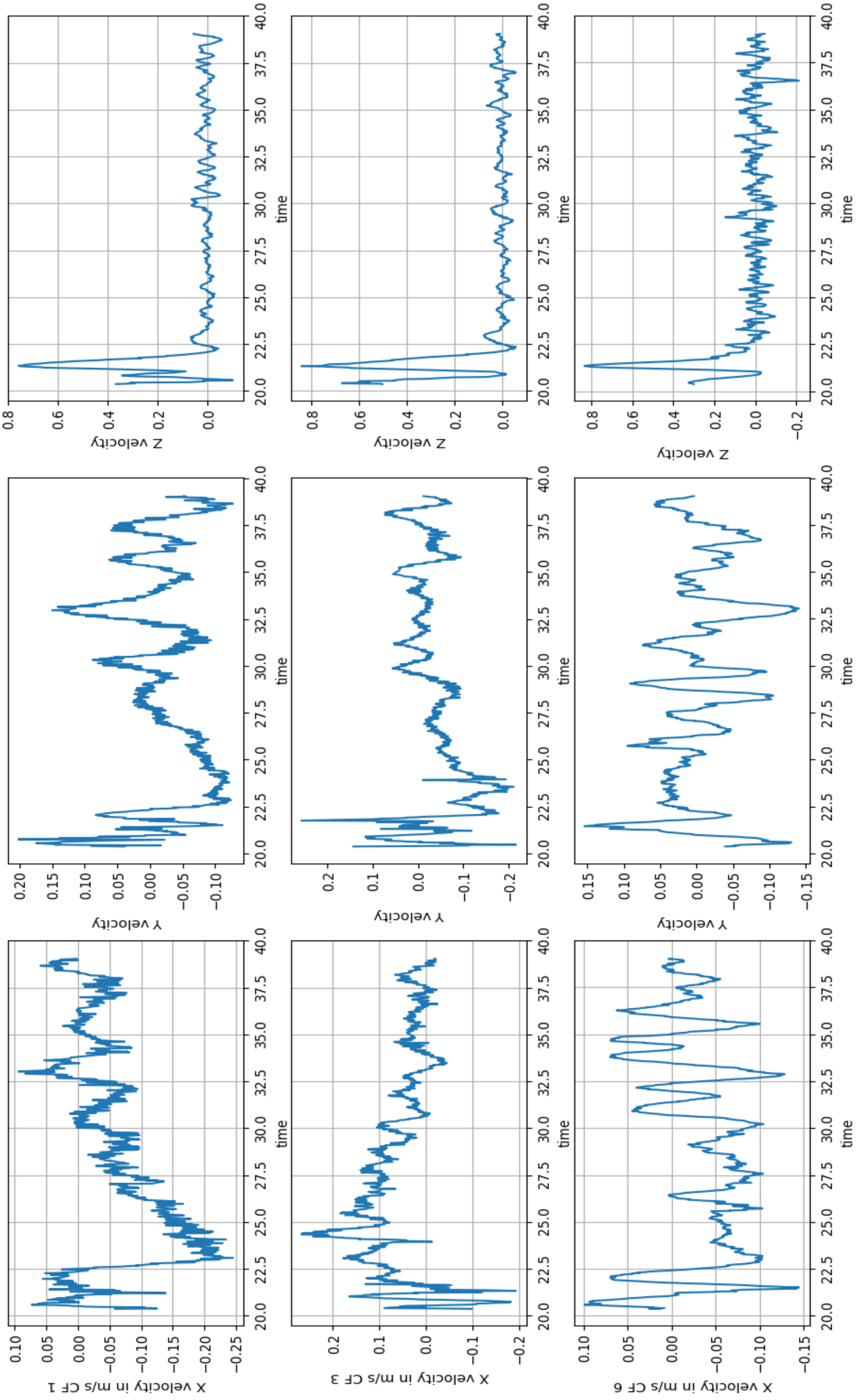

Figure 5.23: Velocity plots.

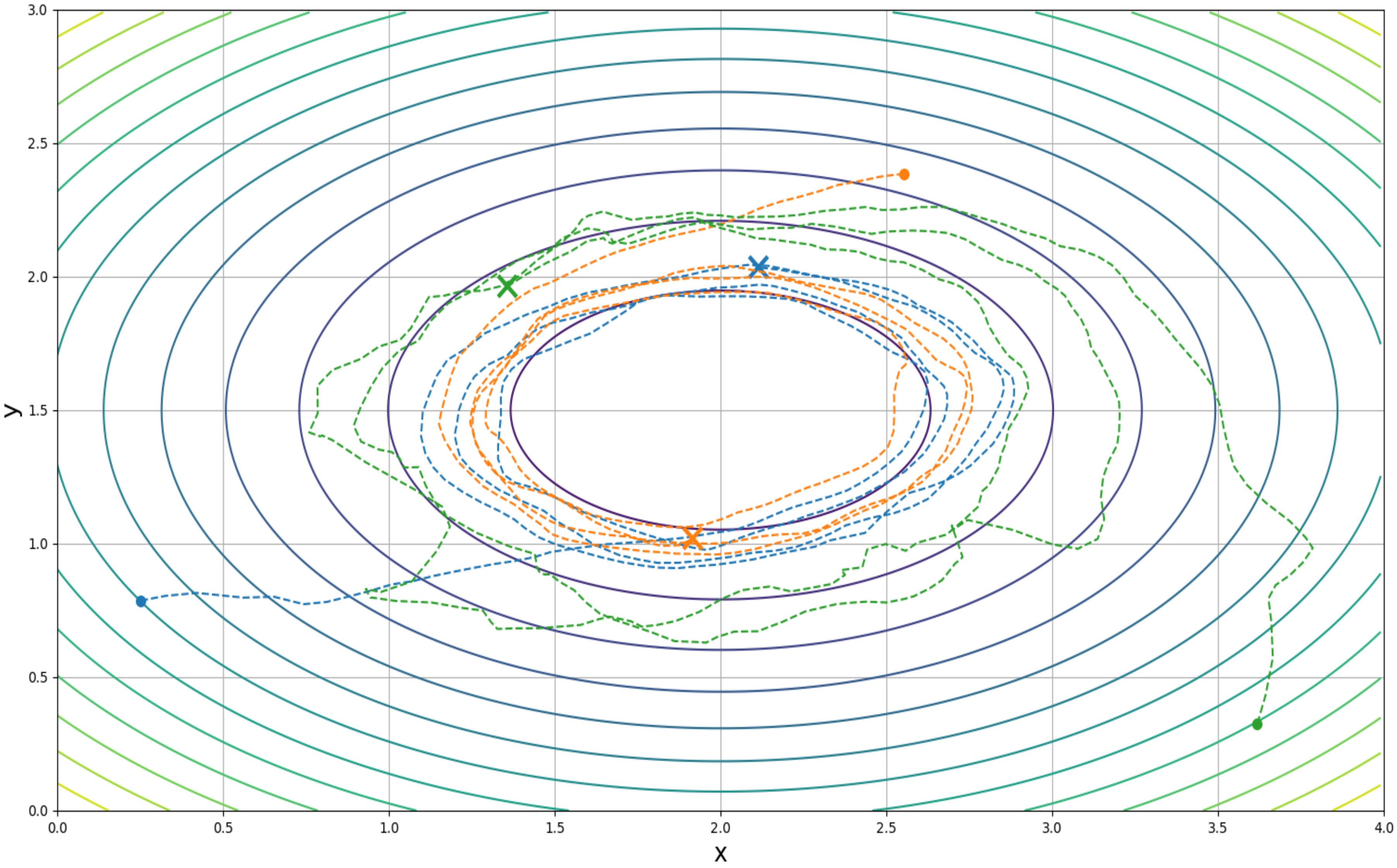

Figure 5.24: Level curve tracking with 3 Quad-copters.

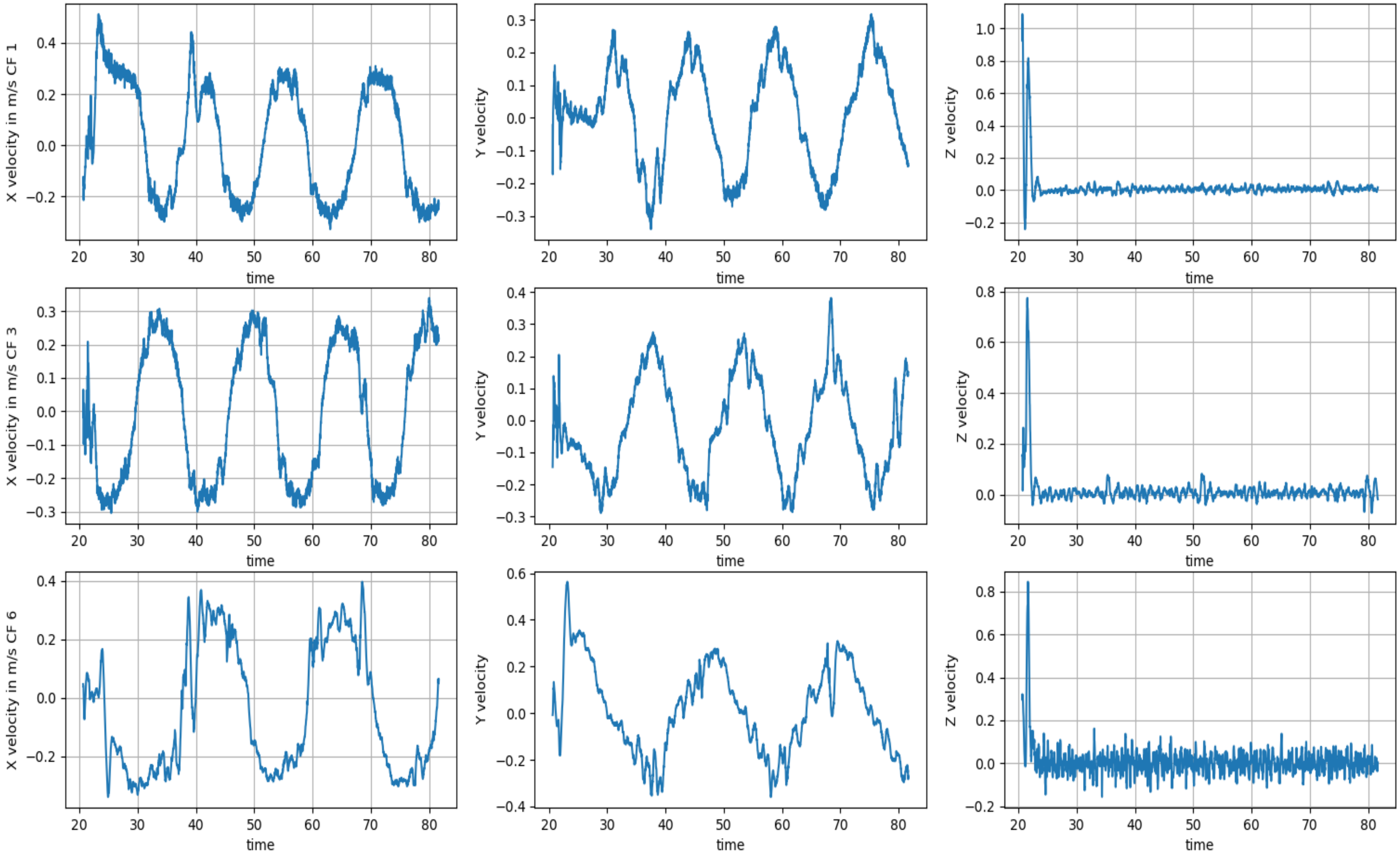

Figure 5.25: Velocity plots.

Chapter 6

Conclusions and Outlook

6.1 Conclusions

This thesis studies the interaction between the information flow dynamics that represent the virtual network and the physical dynamics that represent the agent dynamics which is given in LPV form. In this context, different control approaches have been investigated and analyzed to provide robust stability of the multi-agent system. One of the main aspects of this thesis is to apply the formation control and flocking control to nonholonomic agents represented in LPV form. Modeling nonholonomic agents as LPV models leads to efficient gain scheduled controllers. To apply different cooperative control schemes, a generic model of a dynamic unicycle is used; in addition, modeling of a realistic autonomous underwater vehicle as an LPV model is presented to provide realistic simulation for three-dimensional complex missions.

In the formation control approach, the information flow architecture is investigated and enhanced. The extension from LTI to LPV-modeled agents shows the power and applicability of the information flow scheme. A coupled architecture is proposed in which a distributed controller processes the data flow among the agents. A comparative simulation study between the separated architecture and the proposed coupled architecture shows the advantages of the cooperative behavior among agents in the swarm, and combining the network filter and tracking controller in a one-step design over the separated architecture especially when agents are subject to obstacles or external disturbances.

In case of flocking control, a two-layer coupled architecture is presented. The first layer is the virtual layer which represents the flocking dynamics, and the second layer represents the real agents. By this architecture, the agents' dynamics can be represented as closed loop LPV models. The flocking layer receives data and provides references based on flocking rules and required mission to be tracked by the agents.

Because the flocking layer plays the main role in mission design, it is considered as the main processor of the control loop. Different control laws are presented for source seeking and level surface/curve monitoring. Application of three-dimensional underwater pollution and swarms of autonomous underwater vehicles to locate the source of such pollution is presented. Another application for level surface monitoring is shown to demonstrate the efficacy of this architecture to achieve complex missions. In both case, scalar field gradient and Hessian are assumed to be available to local agents at their current positions.

In contrast to this latter assumption, an estimation scheme is presented where each agent can exchange field concentrations with its neighbors and is equipped with suitable processors to estimate both the scalar field gradient and Hessian. Simulation results show that LPV agents can successfully achieve the required missions.

6.2 Outlook

A number of relevant problems exist that are not covered in this thesis and may be attractive for further research. These topics can be classified in two main categories:

1. **Control system approaches:** like model predictive control, game theory in MAS would require much higher local computational power in agents, and event-triggered control which is motivated by limited communication bandwidth in wireless networks.
2. **Communication in control system approaches:** like communication delays, packet dropouts, switching topology and weighting strategies.

In addition to the aforementioned directions, other approaches include dealing with heterogeneous systems and combinations of two or more approaches like the Model-Predictive-Event-Trigger Control (MPETC) scheme with application to LTI and LPV agents, communication delays or packet dropouts with event-triggered schemes.

Appendix A

Additional Material

Kronecker Product

The operator $\otimes$ denotes the Kronecker product i.e., with $X = [x_{ij}] \in \mathbb{C}^{n_X \times m_X}, x_{ij} \in \mathbb{C}$ and $Y = [y_{ij}] \in \mathbb{C}^{n_Y \times m_Y}, y_{ij} \in \mathbb{C}$, then

$$X \otimes Y = \begin{bmatrix} x_{11}Y & \cdots & x_{1m_X}Y \\ \vdots & \ddots & \vdots \\ x_{n_X 1}Y & \cdots & x_{n_X m_X}Y \end{bmatrix} \tag{A.1}$$

System Decomposition

Given a pattern matrix $P \in \mathbb{R}^{N \times N}$ and a matrix $M \in \mathbb{R}^{Np \times Nq}$, the matrix M is called decomposable (with respect to P) if there exist matrices $M^d, M^c \in \mathbb{R}^{p \times q}$ such that

$$M = I_N \otimes M^d + P \otimes M^c$$

Note that this definition implies that a matrix $M \in \mathbb{R}^{k \times l}$ can only be decomposable w.r.t. $P \in \mathbb{R}^{N \times N}$ if k and l are integer multiples of N.

Lemma A.1. *Given a diagonalisable pattern matrix $P \in \mathbb{R}^{N \times N}$ and a matrix $Z \in \mathbb{R}^{N \times N}$ such that $\det Z \neq 0$ and $Z^{-1}PZ = \Lambda$, where Λ is diagonal. Then for any matrix $M \in \mathbb{R}^{Np \times Nq}$ that is decomposable w.r.t. P, i.e. for which there exist M^d and M^c such that*

$$M = I_N \otimes M^d + P \otimes M^c$$

then,

$$Z_{(p)}^{-1} M Z_{(q)} = I_N \otimes M^d + \Lambda \otimes M^c$$

Definition A.1. *[Hoffmann et al., 2013] Given $L \in \mathbb{R}^{N \times N}$, $M(\theta) : \mathbb{R}^{Nn_\theta} \to \mathbb{R}^{Np \times Nq}$ is called decomposable, if $\exists M^d(\theta_i), M^c(\rho) \in \mathbb{R}^{p \times q}$ with appropriate dimensions $p, q, \forall \theta \in \mathbb{R}^{n_\theta}$, such that*

$$M(\theta) = \bar{M}^d(\theta_i) + L \otimes M^c(\rho)$$

Definition A.2. *[Hoffmann et al., 2013] Matrices $M(\theta)$, with $\theta_i \neq \theta_j$ are called heterogeneous decomposable matrices, whereas matrices $M(\rho)$, with $\theta_i = \rho$ for all i are called homogeneous decomposable matrices.*

Lyapunov Stability

Lyapunov stability theory plays a central role in system theory, as it provides sufficient conditions for the stability of arbitrary system, including non-linear systems. For this reason, here a brief summary is given. Consider an unforced nonlinear system

$$\dot{x}(t) = f(x(t)) \tag{A.2}$$

with an equilibrium state $\bar{x}$ fulfilling $f(\bar{x}) = 0$.

Definition A.3. *An equilibrium $\bar{x}$ is*

- *stable if for every $\varepsilon > 0$ there exists a $\delta(\varepsilon) > 0$ such that*

$$|x(0) - \bar{x}| < \delta \quad \Rightarrow \quad |x(t) - \bar{x}| < \varepsilon \quad \forall t \geq 0$$

- *locally attractive if there exists an $\eta > 0$ such that*

$$|x(0) - \bar{x}| < \eta \quad \Rightarrow \quad \lim_{t \to \infty} x(t) = \bar{x}$$

- *globally attractive if $\lim_{t\to\infty} x(t) = \bar{x}$ is provided for all $x(0)$, i.e. $\eta = \infty$*
- *locally/globally asymptotically stable if it is stable and locally/globally attractive:*

$$\forall \varepsilon > 0 \ \exists \delta(\varepsilon) > 0 \text{ such that}$$
$$|x(0) - \bar{x}| < \delta \Rightarrow \begin{array}{l} |x(t) - \bar{x}| < \varepsilon \quad \forall t \geq 0 \\ \lim_{t\to\infty} x(t) = \bar{x} \end{array}$$

Based on these definitions, we have the following theorem:

Theorem A.1. *Given the system $\dot{x}(t) = f(x(t))$ and a continuously differentiable function $V : \mathbb{R}^m \to \mathbb{R}$ with $V(0) = 0$, the equilibrium $f(\bar{x} = 0) = 0$ is*

- *stable, if*
  - $V(x) > 0 \quad \forall x \in \mathbb{R}^m \backslash \{0\}$
  - $\dot{V}(x(t)) \leq 0 \ \forall x(t) \in \mathbb{R}^m$
- *locally asymptotically stable, if*
  - $V(x) > 0 \quad \forall x \in \mathbb{R}^m \backslash \{0\}$
  - $\dot{V}(x(t)) < 0 \ \forall x(t) \in \mathbb{R}^m \backslash \{0\}$

- *globally asymptotically stable, if*
  - $V(x) > 0 \ \forall x \in \mathbb{R}^m \backslash \{0\}$
  - $\dot{V}(x(t)) < 0 \ \forall x(t) \in \mathbb{R}^m \backslash \{0\}$
  - $V(x) \to \infty$ *as* $\|x\| \to \infty$

A function $V(x)$ that fulfills the conditions given in theorem A.1 is called a Lyapunov function. The forgoing argument shows that if in a domain about the origin, a Lyapunov function whose derivative along the system trajectories is negative semi-definite, no trajectory can stay identically at points where $\dot{V}(x) = 0$, except at the origin, then the origin is asymptotically stable. This argument follows from the LaSalle's Invariance Principle which is applicable to autonomous systems of the form

$$\dot{x} = f(x), \quad f(0) = 0$$

Theorem A.2. *Let* $V(x) : \mathbb{R}^m \to \mathbb{R}$ *and its time derivative* $\dot{V}(x)$ *be negative semidefinite. Let* Ω *be the union of all trajectories where*

$$\Omega = \{x \in \mathbb{R}^m | \dot{V}(x) = 0\}$$

if there exist $\mathbf{M} \in \Omega$ *such that* $\mathbf{M}$ *contains no other trajectory other than* $x = 0$*, then the origin* 0 *is asymptotically stable.*

Proof of Theorem 5.1

Consider a group of N LPV agents operating in an area on which a scalar field $\Psi(q)$ has been defined. The flocking dynamics described in (5.27) can be written as

$$\begin{aligned} \dot{q} &= p \\ \dot{p} &= -\nabla W(q - e_q) - H(x)(p - e_p) \end{aligned} \tag{A.3}$$

where $e_q = x - q, \ ep = v - p$, this can be written as

$$\begin{aligned} \dot{q} &= p \\ \dot{p} &= -\nabla W(q) - H(x)(p) + d \end{aligned} \tag{A.4}$$

where

$$d = \nabla W(q) - \nabla W(q - e_q) + H(x)e_p. \tag{A.5}$$

Consider the following bounds:

$$\begin{aligned} \|\nabla W(q) - \nabla W(q - e_q)\| &= \|\nabla V(q) - \nabla V(q - e_q) + \nabla\Psi(q) - \nabla\Psi(q - e_q)\| \\ &\leq \|\nabla V(q) - \nabla V(q - e_q)\| + \|\nabla\Psi(q) - \nabla\Psi(q - e_q)\| \\ &\leq c_V\|e_q\| + c_\Psi\|e_q\| \leq \alpha_1\|e_q\| \end{aligned} \tag{A.6}$$

where $\alpha_1 = c_V + c_\Psi$, and

$$\begin{aligned}\|He_p\| &= \|\mathcal{L}(m)e_p + \nabla^2\Psi e_p + ce_p\| \\ &\leq \|\mathcal{L}(m)e_p\| + \|\nabla^2\Psi e_p\| + c\|e_p\| \\ &\leq (\|\mathcal{L}(m)\| + c_\Psi + c)\|e_p\| \leq \alpha_2\|e_p\|\end{aligned} \tag{A.7}$$

where $\alpha_2 = (\|\mathcal{L}(m)\| + c_\Psi + c)$. This gives us the following point-wise in time bound on

$$\begin{aligned}\|d(t)\| &\leq \alpha_1\|e_q(t)\| + \alpha_2\|e_p(t)\| \quad \forall t \\ \|d(t)\|^2 &\leq \alpha_1^2\|e_q(t)\|^2 + \alpha_2^2\|e_p(t)\|^2 + 2\alpha_1\alpha_2\|e_q(t)\|\|e_p(t)\| \quad \forall t.\end{aligned}$$

Integrating over time, we obtain $\forall T$

$$\begin{aligned}\|d_T\|_{\mathcal{L}_2}^2 &\leq \alpha_1^2\|(e_q)_T\|_{\mathcal{L}_2}^2 + \alpha_2^2\|(e_p)_T\|_{\mathcal{L}_2}^2 + 2\alpha_1\alpha_2\|(e_q)_T\|_{\mathcal{L}_2}\|(e_p)_T\|_{\mathcal{L}_2} \\ &\leq (\alpha_1^2\gamma^2 + \alpha_2^2\gamma^2 + 2\alpha_1\alpha_2\gamma^2)\|p_t\|_{\mathcal{L}_2}^2 \\ &\leq (\alpha_1 + \alpha_2)^2\gamma^2\|p_t\|_{\mathcal{L}_2}^2.\end{aligned}$$

Consider the energy function $E : \mathbb{R}_+ \to \mathbb{R}_+$ defined as

$$E(t) := W(q(t)) + \frac{1}{2}p^T p. \tag{A.8}$$

Differentiating with respect to time, we obtain

$$\begin{aligned}\dot{E}(t) &= \nabla W(q)^T p + \dot{p}^T p \\ &= \nabla W(q)^T p - \nabla W(q)^T p - p^T H(x)p + d^T p \\ &= -p^T H(x)p + d^T p.\end{aligned} \tag{A.9}$$

Integrating on both sides, we have $\forall T > 0$

$$\begin{aligned}E(T) &= E(0) - \int_0^T p(\tau)^T H(x(\tau))p(\tau)d\tau + \int_0^T d(\tau)^T p(\tau)d\tau \\ &= E(0) - \int_0^T p^T\mathcal{L}(m)pd\tau - \int_0^T p^T\nabla^2\Psi pd\tau - c\int_0^T p^T pd\tau + \int_0^T d^T pd\tau \\ &\leq E(0) - c\int_0^T p^T pd\tau + \int_0^T d^T pd\tau \\ &\leq E(0) - c\|p_T\|_{\mathcal{L}_2}^2 + \langle d_T, p_T\rangle \\ &\leq E(0) - c\|p_T\|_{\mathcal{L}_2}^2 + \|d_T\|\|p_T\| \\ &\leq E(0) - c\|p_T\|_{\mathcal{L}_2}^2 + (\alpha_1 + \alpha_2)\gamma\|p_T\|_{\mathcal{L}_2}^2 \\ &\leq E(0) - (c - (\alpha_1 + \alpha_2)\gamma)\|p_T\|_{\mathcal{L}_2}^2.\end{aligned} \tag{A.10}$$

From the conditions of the Theorem 5.1, we have $c > (\alpha_1 + \alpha_2)\gamma$, which implies $E(T) \leq E(0)$. Therefore,

$$0 \leq \Psi(q(T)) \leq E(0) \quad \forall T > 0 \tag{A.11}$$

which proves the boundedness of trajectories because the level sets of $\Psi(q)$ are bounded by assumption.

Even though the Laplacian $\mathcal{L}_{(m)}$ is state dependent (and therefore time varying), an upper bound on $\|\mathcal{L}_{(m)}\|$ can be obtained using the maximum degree of an agent which can be calculated for planar graphs($\mathbb{R}^2$) to be 6 and can be estimated for agents living in $\mathbb{R}^3$ [Olfati-Saber, 2006]. This can be used for verifying the condition on c as per Theorem 5.1.

Appendix B

3D Dynamic Unicycle

A fixed-wing unmanned aerial vehicle model, Figure B.1, can be modeled as a three-dimensional dynamic unicycle with corresponding model,

$$\begin{aligned}
\dot{q}_x &= v\cos\theta\cos\psi \\
\dot{q}_y &= v\cos\theta\sin\psi \\
\dot{q}_z &= v\sin\theta \\
\dot{\theta} &= \omega_\theta \\
\dot{\psi} &= \omega_\psi \\
\dot{v} &= \frac{1}{m}f \\
\dot{\omega}_\psi &= \frac{1}{I_z}\tau_\psi \\
\dot{\omega}_\theta &= \frac{1}{I_y}\tau_\theta
\end{aligned} \tag{B.1}$$

where (q_x, q_y, q_z) is the UAV c.g. position in the three dimensional space. The angles θ and ψ are the pitch and yaw angles, v is the UAV velocity, ω_ψ and ω_θ are the yaw rate and pitch rate. The constants m, I_z, I_y are mass and moments of inertia around vertical and lateral axes. The UAV has control inputs of thrusting force f, yaw torque τ_ψ and pitch torque τ_θ.

To obtain the LPV model of such UAV, the same procedure in section 3.1 is used by introducing a new point at an appropriate small distance d ahead of UAV c.g.

The coordinates of this point are given by

$$\begin{bmatrix} q_{x_d} \\ q_{y_d} \\ q_{z_d} \end{bmatrix} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} + d\begin{bmatrix} \cos\theta\cos\psi \\ \cos\theta\sin\psi \\ \sin\theta \end{bmatrix} \tag{B.2}$$

A coordinate transformation on the body frame is represented by the orthonormal matrix

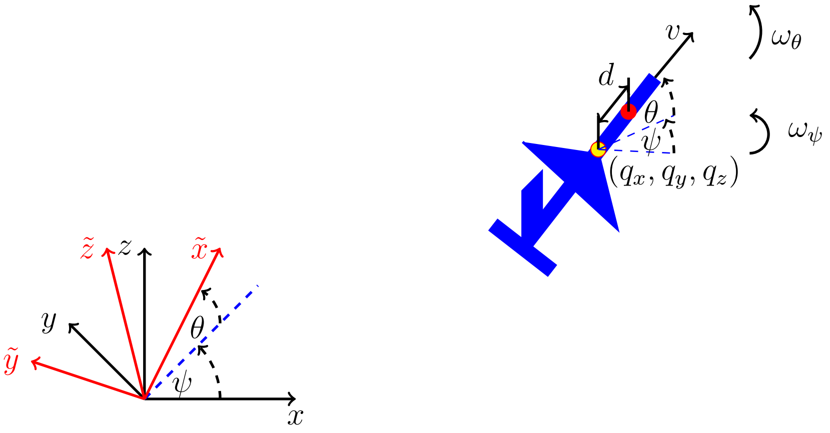

Figure B.1: Fixed-wing UAV

$R_{\theta\psi}$,

$$R_{\theta\psi} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & \sin\theta \\ -\sin\psi & \cos\psi & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix}$$

and,

$$\begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = R_{\theta\psi} \begin{bmatrix} q_{x_d} \\ q_{y_d} \\ q_{z_d} \end{bmatrix}$$

where $(\tilde{x}_d, \tilde{y}_d, \tilde{z}_d)$ is the position at d in the $\tilde{x}\tilde{y}\tilde{z}$-frame. Applying this transformation to (B.2), we have

$$\begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \begin{bmatrix} q_{x_d}\cos\theta\cos\psi + q_{y_d}\cos\theta\sin\psi + q_{z_d}\sin\theta + d \\ -q_{x_d}\sin\psi + q_{y_d}\cos\psi \\ -q_{x_d}\sin\theta\cos\psi - q_{y_d}\sin\theta\sin\psi + q_{z_d}\cos\theta \end{bmatrix}, \tag{B.3}$$

and taking the time derivative yields

$$\begin{bmatrix} \dot{\tilde{x}}_d \\ \dot{\tilde{y}}_d \\ \dot{\tilde{z}}_d \end{bmatrix} = \begin{bmatrix} v + \omega_\theta \tilde{z}_d + \omega_\psi \tilde{y}_d \cos\theta \\ -\omega_\psi \tilde{x}_d + d\omega_\psi \cos\theta \\ v\sin\theta \end{bmatrix}. \tag{B.4}$$

Note the following,

- A nonholonomic vehicle moves only in the direction of its orientation, and it is assumed to be a rigid body, so $v_{\tilde{x}} = v$ and $v_{\tilde{z}} = v\sin\theta$.

- $\omega_\psi = \frac{v_{\tilde{y}}}{d}$ and $\omega_\theta = \frac{v_{\tilde{z}}}{d}$

Based on previous considerations, the UAV model is

$$\begin{bmatrix} \dot{\tilde{x}}_d \\ \dot{\tilde{y}}_d \\ \dot{\tilde{z}}_d \\ \dot{v}_{\tilde{x}} \\ \dot{v}_{\tilde{y}} \\ \dot{v}_{\tilde{z}} \end{bmatrix} = \begin{bmatrix} 0 & \frac{v_{\tilde{y}}}{d}\cos\theta & \frac{v_{\tilde{z}}}{d} & 1 & 0 & 0 \\ -\frac{v_{\tilde{y}}}{d} & 0 & 0 & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \\ v_{\tilde{x}} \\ v_{\tilde{y}} \\ v_{\tilde{z}} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & d \end{bmatrix} \begin{bmatrix} \frac{f}{m} \\ \frac{\tau_\psi}{I_z} \\ \frac{\tau_\theta}{I_y} \end{bmatrix} \tag{B.5}$$

and one choice of a varying parameter vector is $\rho = \begin{bmatrix} v_{\tilde{y}} & v_{\tilde{z}} & \theta \end{bmatrix}^T$.

Appendix C

Nomenclature

Abbreviations

AUVs	Autonomous Underwater Vehicles
c.g.	center of gravity
CCW	Counter Clockwise
EoMs	Equations of Motion
IFF	Information Flow Filter
LFT	Linear Fraction Transformation
LPS	Loco-Positioning System
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
MAS	Multi-Agent Systems
MPETC	Model-Predictive-Event-Trigger Control
SISO	Single-Input-Single-Output
TDOA	Time Difference Of Arrival
UAVs	Unmanned Aerial Vehicles

Symbols

$\mathcal{A}^0$	unscaled adjacency matrix
$\mathcal{A}$	normalized adjacency matrix
A	state matrix
α	speed, flocking agent type, arbitrary constant
B	input matrix
$\mathbf{B}$	Body frame
β	scalar constant
$blkdiag$	block diagonal matrix of input matrices
C	output matrix
c	scalar constant
$\mathcal{D}^0$	unscaled degree matrix

$diag$	diagonal matrix of input elements
D	feed-through matrix
d	distance
$D(\nu)$	AUV damping matrix
$\mathcal{E}$	set of graph edges
e	error
ε	scalar constant
η	AUV position and orientation state
$\mathcal{F}_\mathcal{P}$	scheduling trajectories set
f	thrust force or function
$\mathcal{F}_\mathcal{P}^\mathcal{X}$	scheduling trajectories set
F	information flow filter
$\mathcal{G}$	graph
g	gravitational acceleration or function
γ	induced ℓ_2- or $\mathcal{L}_2$-gain, flocking agent type
$G(\rho)$	LPV plant model
$G(\eta)$	hydrostatic load
$\mathcal{G}_{in}$	scheduled AUV attitude loop dynamics
$\mathcal{G}_{out}$	scheduled AUV position loop dynamics
h	scalar constant
I_z	moment of inertia
$\mathbf{I}$	inertial frame
I_x	moment of inertia
I_y	moment of inertia
J	filter
$K(\rho)$	LPV controller
κ	scalar constant
$\mathcal{L}^0$	unscaled Laplacian matrix
$\mathcal{L}$	normalized Laplacian matrix
λ	eigenvalue of matrix
Λ	diagonal matrix
m	agent mass or space dimension
M_{RB}	inertia matrix
M_A	added mass
μ	scalar constant
N	number of agents in a swarm
$\mathcal{N}_i$	set of neighbours of agent-i
ν	AUV linear and angular velocities state vector
ω	angular velocity or complex number imaginary part
ω_{X_B}	roll rate
ω_{Y_B}	pitch rate
ω_{Z_B}	yaw rate
$\mathbf{P}$	closed Perron disk

ψ	yaw angle
ϕ	roll angle
p	velocity vector or virtual reference
$\Psi(q)$	scalar field
Ψ_{max}	source concentration value
q_x	x position in space
q_y	y position in space
q_z	z position in space
q_s	source position
r	reference signal, communication range
ρ	scheduling vector (continuous-time)
$R_{\phi\theta\psi}$	rotational tensor
ρ_h	bump function
σ	complex number real part or defined function
θ	discrete-time scheduling vector or pitch angle
τ_ψ	yaw torque
T_s	sampling time
τ_ϕ	roll torque
τ_θ	pitch torque
$T_{\phi\theta\psi}$	velocity transformation tensor
$\mathbf{U}$	closed unit disk
u	control input
$\mathcal{V}$	set of graph vertices
v_n	(longitudinal) linear velocity
v_t	lateral velocity
v_{X_B}	longitudinal velocity
v_{Y_B}	lateral velocity
v_{Z_B}	vertical velocity
W_s	shaping filter
W_{ks}	shaping filter
W_f	shaping filter
W_{ut}	shaping filter
W_t	shaping filter
χ	scheduling parameter rate vector
x	open-loop system state vector
x_C	controller state vector
x_{cl}	closed-loop system state vector
$\bar{\chi}$	scheduling parameter rate upper bound
$\mathcal{X}$	scheduling trajectories rate set
z	performance output, z-domain

Bibliography

H Abbas, SM Hashemi, and H Werner. Decentralized LPV gain-scheduled PD control of a robotic manipulator. In *ASME 2009 Dynamic Systems and Control Conference*, pages 801–808. American Society of Mechanical Engineers Digital Collection, 2009.

Hossam Seddik Abbas. *LPV Modeling, Identification and Low-Complexity Controller Synthesis*. Ph.d. thesis, Hamburg University of Technology, Hamburg, Germany, 2010.

Mukhtar Ali and Herbert Werner. Discrete-time LPV controller synthesis using dilated LMIs with application to an arm-driven inverted pendulum. *IFAC Proceedings Volumes*, 44(1):7708–7712, 2011.

Carl Anderson, Nigel R Franks, and Daniel W McShea. The complexity and hierarchical structure of tasks in insect societies. *Animal Behaviour*, 62(4):643–651, 2001.

Pierre Apkarian and Pascal Gahinet. A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864, 1995.

Pierre Apkarian, Pascal Gahinet, and Greg Becker. Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261, 1995.

Aly Attallah and Herbert Werner. Information flow in formation control for nonholonomic agents modeled as LPV systems. In *2020 European Control Conference (ECC)*, pages 459–464, 2020.

Aly Attallah, Adwait Datar, and Herbert Werner. Flocking of linear parameter varying agents: Source seeking application with underwater vehicles. *IFAC-PapersOnLine*, 53 (2):7305–7311, 2020.

Tucker Balch and Ronald C Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

Siavash Ahmadi Barogh, Esteban Rosero, and Herbert Werner. Formation control of non-holonomic agents with collision avoidance. In *2015 American Control Conference (ACC)*, pages 757–762, 2015.

Marcus Bartels. *Design of Distributed Formation Controllers for Multi-Agent Systems*. PhD thesis, Hamburg University of Technology, 2019.

Marcus Bartels and Herbert Werner. Cooperative and consensus-based approaches to formation control of autonomous vehicles. *IFAC Proceedings Volumes*, 47(3):8079–8084, 2014.

Randal W Beard, Jonathan Lawton, and Fred Y Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9 (6):777–790, 2001.

Jinde Cao, Guanrong Chen, and Ping Li. Global synchronization in an array of delayed neural networks with hybrid coupling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2):488–498, 2008.

Fei Chen and Wei Ren. On the control of multi-agent systems: A survey. *Foundations and Trends in Systems and Control*, 6:339–499, 2019.

H Chen, J Wang, XQ Gao, and ZY Liu. A moving horizon H_∞ tracking scheme for wheeled mobile robots with actuator constraints. *IFAC Proceedings Volumes*, 37(13): 627–632, 2004.

P. S. G. Cisneros, C. Hoffmann, M. Bartels, and H. Werner. Linear parameter-varying controller design for a nonlinear quad-rotor helicopter model for high speed trajectory tracking. In *2016 American Control Conference (ACC)*, pages 486–491, 2016.

Pablo S. G. Cisneros and Herbert Werner. A velocity algorithm for nonlinear model predictive control. *IEEE Transactions on Control Systems Technology*, 29(3):1310–1315, 2021.

Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. Leader–follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44(5):1343–1349, 2008.

Jorge Cortés and Magnus Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6):495–503, 2017.

Wenxia Cui, Yang Tang, Jian-An Fang, and Jürgen Kurths. Consensus analysis of second-order multi-agent networks with sampled data and packet losses. *IEEE Access*, 4: 8127–8137, 2016.

Adwait Datar. Private Communication, 2019.

Adwait Datar, Daniel Schneider, Furugh Mirali, Herbert Werner, and Hannes Frey. A memory weighted protocol for sampled-data systems subjected to packet dropouts. In *2018 American Control Conference (ACC)*, pages 2485–2490, 2018.

Adwait Datar, Peter Paulsen, and Herbert Werner. Flocking towards the source: Indoor experiments with quadrotors. In *2020 European Control Conference (ECC)*, 2020.

Jan De Caigny, Juan F Camino, Ricardo CLF Oliveira, Pedro LD Peres, and Jan Swevers. Gain-scheduled dynamic output feedback control for discrete-time LPV systems. *International Journal of Robust and Nonlinear Control*, 22(5):535–558, 2012.

C. E. de Souza, K. A. Barbosa, and A. T. Neto. Robust H_∞ filtering for discrete-time linear systems with uncertain time-varying parameters. *IEEE Transactions on Signal Processing*, 54(6):2110–2118, 2006.

Jonathan A DeCastro. Rate-based model predictive control of turbofan engine clearance. *Journal of Propulsion and Power*, 23(4):804–813, 2007.

Daniel Andre Duecker, Edwin Kreuzer, Gunnar Maerker, and Eugen Solowjow. Parameter identification for micro underwater vehicles. *PAMM*, 18(1):e201800350, 2018.

Annika Eichler and Herbert Werner. Performance bounds on decomposable systems. In *2013 American Control Conference*, pages 3224–3229, 2013.

Tolga Eren, Peter N Belhumeur, Brian DO Anderson, and A Stephen Morse. A framework for maintaining formations based on rigidity. *IFAC Proceedings Volumes*, 35(1):499–504, 2002.

J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.

Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

Pascal Gahinet and Pierre Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4(4):421–448, 1994.

Simon Garnier, Jacques Gautrais, and Guy Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31, 2007.

A. M. Gonzalez, C. Hoffmann, and H. Werner. LPV formation control for a class of nonholonomic agents with directed and switching communication topologies. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2792–2797, 2015.

Miroslav Halás, Mikuláš Huba, and Katarína Žáková. The exact velocity linearization method. *IFAC Proceedings Volumes*, 36(18):259–264, 2003.

He Hao and Prabir Barooah. Improving convergence rate of distributed consensus through asymmetric weights. In *2012 American Control Conference (ACC)*, pages 787–792, 2012.

Arnar Hjartarson, Peter Seiler, and Andrew Packard. Lpvtools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems. *IFAC-PapersOnLine*, 48 (26):139–145, 2015.

C. Hoffmann, A. Eichler, and H. Werner. Distributed control of linear parameter-varying decomposable systems. In *2013 American Control Conference*, pages 2380–2385, 2013.

Christian Hoffmann. *Linear parameter-varying control of systems of high complexity*. PhD thesis, Hamburg University of Technology, 2016.

Christian Hoffmann and Herbert Werner. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23(2):416–433, 2014.

Yiguang Hong, Jiangping Hu, and Linxin Gao. Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42(7):1177–1182, 2006.

Yiguang Hong, Guanrong Chen, and Linda Bushnell. Distributed observers design for leader-following control of multi-agent networks. *Automatica*, 44(3):846–850, 2008.

Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

Yongnan Jia and Long Wang. Experimental implementation of distributed flocking algorithm for multiple robotic fish. *Control Engineering Practice*, 30:1–11, 2014.

Isaac Kaminer, Antonio M Pascoal, Pramod P Khargonekar, and Edward E Coleman. A velocity algorithm for the implementation of gain-scheduled controllers. *Automatica*, 31(8):1185–1191, 1995.

Hassan K Khalil. *Nonlinear Systems*, volume 3. Prentice Hall Upper Saddle River, NJ, 2002.

Steffi Knorn, Zhiyong Chen, and Richard H Middleton. Overview: Collective control of multiagent systems. *IEEE Transactions on Control of Network Systems*, 3(4):334–347, 2015.

Laura Krick, Mireille E Broucke, and Bruce A Francis. Stabilization of infinitesimally rigid formations of multi-robot networks. In *2008 47th IEEE Conference on Decision and Control*, pages 477–482, 2008.

KN Krishnanand and Debasish Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intelligence*, 3(2): 87–124, 2009.

Andreas Kwiatkowski. *LPV Modeling and Application of LPV Controllers to SI Engines*. PhD thesis, Hamburg University of Technology, 2008.

Andreas Kwiatkowski, Marie-Theres Boll, and Herbert Werner. Automated generation and assessment of affine LPV models. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6690–6695, 2006.

Jonathan RT Lawton, Randal W Beard, and Brett J Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, 2003.

DJ Leith and WE Leithead. Gain-scheduled control of a skid-to-turn missile: relaxing slow variation requirements by velocity-based design. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 1, pages 500–505, 2001.

Douglas J Leith and WE Leithead. Comments on 'gain scheduling dynamic linear controllers for a nonlinear plant'. *Automatica*, 34:1041–1043, 1998a.

Douglas J Leith and WE Leithead. Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearization families. *International Journal of Control*, 70 (2):289–317, 1998b.

Douglas J Leith and WE Leithead. Gain-scheduled controller design: an analytic framework directly incorporating non-equilibrium plant dynamics. *International Journal of Control*, 70(2):249–269, 1998c.

Naomi Ehrich Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 3, pages 2968–2973, 2001.

Yanjiang Li and Chong Tan. A survey of the consensus for multi-agent systems. *Systems Science & Control Engineering*, 7(1):468–482, 2019.

Zhiyun Lin, Bruce Francis, and Manfredi Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on Automatic Control*, 50(1):121–127, 2005.

Marco Lovera, Marco Bergamasco, and Francesco Casella. LPV modelling and identification: An overview. In *Robust Control and Linear Parameter Varying Approaches*, pages 3–24. Springer, 2013.

Chang Boon Low. A flexible virtual structure formation keeping control design for nonholonomic mobile robots with low-level control systems, with experiments. In *2014 IEEE International Symposium on Intelligent Control (ISIC)*, pages 1576–1582, 2014.

I Lugo-Cárdenas, S Salazar, and R Lozano. Lyapunov based 3D path following kinematic controller for a fixed wing UAV. *IFAC-PapersOnLine*, 50(1):15946–15951, 2017.

Xiaoyuan Luo, Shaobao Li, and Xinping Guan. Flocking algorithm with multi-target tracking for multi-agent systems. *Pattern Recognition Letters*, 31(9):800–805, 2010.

Joshua A Marshall, Mireille E Broucke, and Bruce A Francis. Pursuit formations of unicycles. *Automatica*, 42(1):3–12, 2006.

Paolo Massioni and Michel Verhaegen. Distributed control for identical dynamically coupled systems: A decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, 2009.

Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

Furugh Mirali and Herbert Werner. Distributed weighting strategies for improved convergence speed of first-order consensus. In *2017 American Control Conference (ACC)*, pages 942–947, 2017.

Furugh Mirali, Antonio Mendez Gonzalez, and Herbert Werner. First-order average consensus for cooperative control problems using novel weighting strategies. *IFAC-PapersOnLine*, 50(1):14302–14307, 2017.

Alexander Mogilner and Leah Edelstein-Keshet. A non-local model for a swarm. *Journal of Mathematical Biology*, 38(6):534–570, 1999.

Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

R Olfati-Saber and RM Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 3, pages 2965–2971, 2002.

Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9): 1520–1533, 2004.

Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

Peter Paulsen. Development of a software framework for implementation of cooperative control algorithms on a swarm of small quad-rotors. Project Report, Hamburg University of Technology, 2018.

Peter Paulsen. Experimental investigations into the eect of communication constraints on performance of cooperative control algorithms: Case study with quadrotors. Master thesis, Hamburg University of Technology, 2019.

T Péni, G Szederkényi, J Bokor, and KM Hangos. Dynamic inversion based velocity tracking control of road vehicles. *IFAC Proceedings Volumes*, 37(13):919–924, 2004.

Ulf Pilz, Andrey P Popov, and Herbert Werner. An information flow filter approach to cooperative vehicle control. *IFAC Proceedings Volumes*, 44(1):7432–7437, 2011.

Andrey Popov. *Design of distributed and fixed-structure controllers for cooperative vehicle control*. PhD thesis, Hamburg University of Technology, 2012.

Andrey Popov and Herbert Werner. Robust stability of a multi-agent system under arbitrary and time-varying communication topologies and communication delays. *IEEE Transactions on Automatic Control*, 57(9):2343–2347, 2012.

Luc Reberga, Didier Henrion, Jacques Bernussou, and Florian Vary. LPV modeling of a turbofan engine. *IFAC Proceedings Volumes*, 38(1):526–531, 2005.

Wei Ren and Randal W Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5): 655–661, 2005.

Wei Ren and Randal W Beard. Consensus algorithms for double-integrator dynamics. *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications*, pages 77–104, 2008.

Wei Ren and Yongcan Cao. *Distributed coordination of multi-agent networks: emergent problems, models, and issues.* Springer Science & Business Media, 2010.

Craig W Reynolds. *Flocks, herds and schools: A distributed behavioral model*, volume 21. ACM, 1987.

Esteban Emilio Rosero García. *Cooperative source seeking and level curve tracking for multi-agent systems.* PhD thesis, Hamburg University of Technology, 2017.

Daniel P Scharf, Fred Y Hadaegh, and Scott R Ploen. A survey of spacecraft formation flying guidance and control. part ii: control. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 2976–2985, 2004.

Rodolphe Sepulchre, Derek A Paley, and Naomi Ehrich Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52(5):811–824, 2007.

Rodolphe Sepulchre, Derek A Paley, and Naomi Ehrich Leonard. Stabilization of planar collective motion with limited communication. *IEEE Transactions on Automatic Control*, 53(3):706–719, 2008.

J. S. Shamma and M. Athans. Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907, 1990.

Naohiko Shimoyama, Ken Sugawara, Tsuyoshi Mizuguchi, Yoshinori Hayakawa, and Masaki Sano. Collective motion in a system of motile elements. *Physical Review Letters*, 76(20):3870, 1996.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied Nonlinear Control*, volume 199. Prentice Hall Englewood Cliffs, NJ, 1991.

Gilbert Strang. *Linear Algebra and its Applications.* Thomson, Brooks/Cole, 2006.

Housheng Su, Xiaofan Wang, and Guanrong Chen. A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements. *International Journal of Control*, 82(7):1334–1343, 2009a.

Housheng Su, Xiaofan Wang, and Zongli Lin. Flocking of multi-agents with a virtual leader. *IEEE Transactions on Automatic Control*, 54(2):293–307, 2009b.

Tyler H Summers, Changbin Yu, Soura Dasgupta, and Brian DO Anderson. Control of minimally persistent leader-remote-follower and coleader formations in the plane. *IEEE Transactions on Automatic Control*, 56(12):2778–2792, 2011.

Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 2, pages 2010–2015, 2003a.

Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Stable flocking of mobile agents part ii: dynamic topology. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 2, pages 2016–2021, 2003b.

John Toner and Yuhai Tu. Flocks, herds, and schools: A quantitative theory of flocking. *Physical Review E*, 58(4):4828, 1998.

Roland Tóth. *Modeling and Identification of Linear Parameter-Varying Systems*, volume 403. Springer, 2010.

Avi Turgeman and Herbert Werner. Multiple source seeking using glowworm swarm optimization and distributed gradient estimation. In *2018 Annual American Control Conference (ACC)*, pages 3558–3563, 2018.

N. H. Vaidya, C. N. Hadjicostis, and A. D. DomÃnguez-GarcÃa. Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2761–2766, 2012.

Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 75 (6):1226, 1995.

Shen Wang, Cuijuan An, Xiao-Xiang Sun, and Xin Du. Average consensus over communication channels with uniform packet losses. In *2010 Chinese Control and Decision Conference*, pages 114–119, 2010.

Xiao Fan Wang and Guanrong Chen. Synchronization in small-world dynamical networks. *International Journal of Bifurcation and Chaos*, 12(01):187–192, 2002.

Guanghui Wen, Zhisheng Duan, Zhongkui Li, and Guanrong Chen. Flocking of multi-agent dynamical systems with intermittent nonlinear velocity measurements. *International Journal of Robust and Nonlinear Control*, 22(16):1790–1805, 2012.

Fen Wu. *Control of Linear Parameter Varying Systems.* PhD thesis, University of California, Berkeley, 1995.

Fen Wu, Xin Hua Yang, Andy Packard, and Greg Becker. Induced $\mathcal{L}_2$-norm control for LPV systems with bounded parameter variation rates. *International Journal of Robust and Nonlinear Control*, 6(9-10):983–998, 1996.

Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

Pian Yu, Li Ding, Zhi-Wei Liu, and Zhi-Hong Guan. Leader–follower flocking based on distributed event-triggered hybrid control. *International Journal of Robust and Nonlinear Control*, 26(1):143–153, 2016.

Wenwu Yu, Guanrong Chen, Ming Cao, and Jürgen Kurths. Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(3):881–891, 2009a.

Wenwu Yu, Guanrong Chen, and Jinhu Lü. On pinning synchronization of complex dynamical networks. *Automatica*, 45(2):429–435, 2009b.

Wenwu Yu, Guanrong Chen, and Ming Cao. Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica*, 46(6):1089–1095, 2010.

Michael M Zavlanos, Ali Jadbabaie, and George J Pappas. Flocking while preserving network connectivity. In *2007 46th IEEE Conference on Decision and Control*, pages 2919–2924, 2007.

Xiao-Wen Zhao, Zhi-Hong Guan, Juan Li, Xian-He Zhang, and Chao-Yang Chen. Flocking of multi-agent nonholonomic systems with unknown leader dynamics and relative measurements. *International Journal of Robust and Nonlinear Control*, 27(17):3685–3702, 2017.

Publications

Published

[1] Aly Attallah and Herbert Werner, "Information flow in formation control for nonholonomic agents modeled as LPV systems," In *2020 European Control Conference (ECC)*, pages 459-464, 2020.

[2] Aly Attallah, Adwait Datar, and Herbert Werner, "Flocking of linear parameter varying agents: Source seeking application with underwater vehicles," in 21^{st} *IFAC World Congress.* Elsevier, 2020.

Submitted

[1] Aly Attallah and Herbert Werner, "Distributed Control for Complex Missions: Quasi-LPV Approach," *International Journal of Robust and Nonlinear Control.*

Curriculum Vitae

Personal Information

Name	Aly Saeed Aly Aly Attallah
Date of birth	August 14, 1987
Place of birth	Menoufiya, Egypt
Citizenship	Egypt

School Education

09/1993-06/1998	Primary School: Al-Sadat Primary School, Menoufiya, Egypt.
09/1998-06/2001	Preparatory School: Al-Nasr Preparatory School, Menoufiya, Egypt.
09/2001-06/2004	Secondary School: Quwesna Secondary School, Menoufiya, Egypt.

Academic Education

09/2005-07/2010	B.Sc. in Electrical Engineering; Military Technical College (MTC), Cairo, Egypt. • Thesis: Unmanned Aerial Vehicles: Modeling and Control.
09/2013-08/2016	M.Sc. in Electrical Engineering; MTC, Cairo, Egypt. • Thesis: Range Extension of Launched Aerial Vehicles.
11/2017-10/2020	PhD studies; ICS, TUHH, Hamburg, Germany.

Work Experience

01/2011-07/2012	Research Assistant, EAF Research Center, Cairo, Egypt.
07/2012-08/2016	Research Assistant in Electrical Engineering Department; Military Technical College, Cairo, Egypt.
08/2016-10/2017	Associated Lecturer, MTC, Cairo, Egypt.
11/2017-10/2020	Research Assistant; ICS, Hamburg University of Technology • Research topic: Distributed Control of Multi-Agent Systems • Teaching responsibilities and supervision of thesis work • Supervision of thesis work

Awards

07/2010	Medal of Duty.
08/2017	Egyptian Government Scholarship for pursuing PhD at TU Hamburg.